
Volume III

Version 5.0

Document Number 007-0602-050
This document contains proprietary information of Silicon Graphics, Inc. The contents of this document may not be disclosed to third parties, copied or duplicated in any form, in whole or in part, without the prior written permission of Silicon Graphics, Inc.

Restricted Rights Legend

Use, duplication or disclosure of the technical data contained in this document by the Government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013, and/or in similar or successor clauses in the FAR, or the DOD or NASA FAR Supplement. Unpublished rights reserved under the Copyright Laws of the United States. Contractor/manufacturer is Silicon Graphics, Inc., 2011 N. Shoreline Blvd., Mountain View, CA 94039-7311.

Version 5.0
Document Number 007-0602-050

Silicon Graphics, Inc.
Mountain View, California

IRIX is a trademark of Silicon Graphics, Inc.
UNIX is a trademark of AT&T, Inc.
TABLE OF CONTENTS

4. File Formats

intro ........................................ introduction to file formats
a.out ........................................ assembler and link editor output
acct ........................................... per-process accounting file format
aliases ........................................ aliases file for sendmail
ar ............................................... archive (library) file format
autologin .................................... set autologin user identity
cftime ........................................ language specific strings
core ........................................... format of core image file
cpio ........................................... format of cpio archive
cshrc .......................................... system-wide csh initialization command file
dbg ........................................... the debug file system
dir ............................................. format of EFS directories
dirent .......................................... file system independent directory entry
filehdr ........................................ file header for MIPS object files
fs ............................................... layout of the Extent file system/fspec .......................................... format specification in text files
fstab .......................................... static information about filesystems
gettydefs .................................... speed and terminal settings used by getty
group .......................................... group membership file
hosts .......................................... host name data base
hosts.equiv .................................. list of trusted hosts
inittab ....................................... script for the init process
inode .......................................... format of an Extent File System inode
ldfcn .......................................... common object file access routines
limits ......................................... file header for implementation-specific constants
linenum ....................................... line number entries in a MIPS object file
login .......................................... login configuration file
lvtab .......................................... information about logical volumes
master ......................................... master configuration database
motd .......................................... message of the day
mtab .......................................... mounted file system table
networks ..................................... network name data base
passwd ........................................ password file
pnc ........................................... file format for card images
printcap ..................................... printer capability data base
profile ...................................... setting up an environment at login time
proto .......................................... prototype job file for at
protocols .................................... protocol name data base
queuedefs ................................... at/batch/cron queue description file
rcsfile ....................................... format of RCS file
reloc .......................................... relocation information for a common object file
resolver ...................................... host-address resolver configuration file
Table of Contents

rhosts ................. list of trusted hosts and users
rpc .................... RPC program number data base
scsfile ................. format of SCCS file
scnhdr ................. section header for a MIPS object file
scr_dump ............... format of curses screen image file
services ................ service name data base
syms ................... MIPS symbol table
sys_id .................. system identification file
system ................. system configuration information table
term .................... format of compiled term file.
terminfo ............... terminal capability data base
timezone .............. set default system time zone
transferdevice ......... a spec for extending WorkSpace menu
ttytype ................ data base of terminal types by port
unistd ................ file header for symbolic constants
utmp ................... utmp and wtmp entry formats
uuencode ............... format of an encoded uuencode file
visuallogin ........... select and control console login program

5. Miscellaneous Facilities

intro ................... introduction to miscellany
ascii .................. map of ASCII character set
charset ................. description of the standard supported character set
environ ................ user environment
fcntl .................. file control options
hostname ............... host name resolution description
math ................... math functions and constants
regexp ................... regular expression compile and match routines
stat .................... data returned by stat system call
stdarg ................ variable argument list
term ................... conventional names for terminals
types ................... primitive system data types
values ................ machine-dependent values
varargs ................ variable argument list
winicons ............... stowed window image mechanism
NAME
intro – introduction to file formats

DESCRIPTION
This section outlines the formats of various files. The C structure declarations for the file formats are given where applicable. Usually, the header files containing these structure declarations can be found in the directories /usr/include or /usr/include/sys. For inclusion in C language programs, however, the syntax #include <filename.h> or #include <sys/filename.h> should be used.
NAME
a.out – assembler and link editor output

SYNOPSIS
#include <a.out.h>

DESCRIPTION
\( a.out \) is the output file format of the assembler \textit{as}(1) and the link editor \textit{ld}(1). Both programs make \textit{a.out} executable if there were no errors and no unresolved external references. The debugger uses the \textit{a.out} file to provide symbolic information to the user.

IRIX and the MIPS compilers use a file format that is similar to standard AT\&T System V COFF (common object file format). For more information, see the \textit{Assembly Language Programmer’s Guide}.

The MIPS File Header definition is based on the System V header file \textit{filehdr.h} with the following changes (also see \textit{filehdr}(4)):

- The symbol table file pointer, \textit{f_symptr}, and the number of symbol table entries \textit{f_nsym}, now specify the file pointer and the size of the Symbolic Header respectively.
- All tables that specify symbolic information have their file pointers and number of entries in the Symbolic Header.

The Optional Header definition has the same format as the System V header file \textit{aouthdr.h} (the “standard” (pre-COFF) UNIX system \textit{a.out} header) except the following fields have been added: \textit{bss_start}, \textit{gprmask}, \textit{cprmask}, and \textit{gp_value}.

The Section Header definition has the same format as the System V header file \textit{scnhdr.h}, except the line number fields (\textit{s_innopr} and \textit{s_nlinno}) are used for gp tables (see \textit{scnhdr}(4)).

The MIPS relocation information definition is similar to that in Berkeley 4.3 UNIX, which has “local” relocation types (see \textit{reloc}(4)). Also see the section entitled \textbf{Section Relocation Information} in the \textit{Assembly Language Programmer’s Guide} for more detailed information.

For more information about System V COFF, refer to the AT\&T UNIX System V Support Tools Guide.

The MIPS file format follows this scheme:
- File Header
- Optional Headers
- Section Headers
- Section Data—includes text, read-only data, large data, 8 and 4-byte literal pools, small data, small bss (0 size), large bss (0 size), and shared library information.
- Section Relocation Information—includes text, read-only data, large data, 8 and 4-byte literal pools, and small data.
- Global Pointer (GP) Tables—missing if relocation information is not saved.
- Symbolic Header—missing if fully stripped.
- Line Numbers—created only if debugging is on, and missing if stripped of non-globals or fully stripped.
- Procedure Descriptor Table—missing if fully stripped.
- Local Symbols—missing if stripped of non-globals or if fully stripped.
- Optimization Symbols—created only if debugging is on, and missing if stripped of non-globals or fully stripped.
- Auxiliary Symbols—created only if debugging is on, and missing if stripped of non-globals or fully stripped.
- Local Strings—missing if stripped of non-globals or if fully stripped.
- External Strings—missing if fully stripped.
- Relative File Descriptor—missing if stripped of non-globals or if fully stripped.
- File Descriptors—missing if stripped of non-globals or if fully stripped.
- External Symbols—missing is fully stripped.

The Section Data

MIPS files are represented in several sections: .text, .rdata (read-only data), .data (data), .sdata (small data), .lit8 (8-byte literal pool), .lit4 (4-byte literal pool), .sbss (small block started by storage), .bss (block started by storage), .init (initialization), and .lib (shared library references).
Generally only sections actually needed in an *a.out* file are present in the file.

The `.text` section contains the machine instructions that are to be executed; the `.rdata`, `.data`, `.sdata`, `.lit8`, and `.lit4` sections contain initialized data; the `.sbss` and `.bss` sections reserve space for ininitialized data that is created by the kernel loader for the program before execution and filled with zeros.

The `.init` section contains shared library interface initialization information. The `.lib` section contains references to the shared libraries this *a.out* file uses.

**SEE ALSO**

*Assembly Language Programmer's Guide*

`as(1), ld(1), nm(1), dbx(1), strip(1), filehdr(4), scnhdr(4), reloc(4), sym(4), linenum(4)`
NAME
acct – per-process accounting file format

SYNOPSIS
#include <sys/acct.h>

description
Files produced as a result of calling acct(2) have records in the form defined by <sys/acct.h>, whose contents are:
typedef ushort comp_t;    /* "floating point" */
                        /* 13-bit fraction, 3-bit exponent */

struct acct
{
    char ac_flag;       /* Accounting flag */
    char ac_stat;       /* Exit status */
    ushort ac_uid;      /* Accounting user ID */
    ushort ac_gid;      /* Accounting group ID */
    dev_t ac_ity;       /* control typewriter */
    time_t ac_btime;    /* Beginning time */
    comp_t ac_utime;    /* acctng user time in clock ticks */
    comp_t ac_stime;    /* acctng system time in clock ticks */
    comp_t ac_etime;    /* acctng elapsed time in clock ticks */
    comp_t ac_mem;      /* memory usage in clicks */
    comp_t ac_io;       /* chars trnsfrd by read/write */
    comp_t ac_rw;       /* number of block reads/writes */
    char ac_comm[8];    /* command name */
};

extern struct acct acctbuf;
extern struct inode *acctp; /* inode of accounting file */

#define AFORK 01    /* has executed fork, but no exec */
#define ASU 02      /* used super-user privileges */
#define ACCTF 0300  /* record type: 00 = acct */

In ac_flag, the AFORK flag is turned on by each fork(2) and turned off by an exec(2). The ac_comm field is inherited from the parent process and is reset by any exec. Each time the system charges the process with a clock tick, it also adds to ac_mem the current process size, computed as follows:

(data size) + (text size) / (number of in-core processes using text)
The value of $ac\_mem/(ac\_stime + ac\_ut ime)$ can be viewed as an approximation to the mean process size, as modified by text sharing.

The structure acct, which resides with the source files of the accounting commands, represents the total accounting format used by the various accounting commands:

 /*
 * total accounting (for acct period), also for day
 */

 struct tacct {
   uid_t    ta_uid;  /* userid */
   char     ta_name[8];  /* login name */
   float    ta_cpu[2];  /* cum. cpu time, p/np (mins) */
   float    ta_kcore[2];  /* cum kcore-minutes, p/np */
   float    ta_con[2];  /* cum. connect time, p/np, mins */
   float    ta_du;  /* cum. disk usage */
   long     ta_pc;  /* count of processes */
   unsigned short    ta_sc;  /* count of login sessions */
   unsigned short    ta_dc;  /* count of disk samples */
   unsigned short    ta_fee;  /* fee for special services */
};


BUGS

The $ac\_mem$ value for a short-lived command gives little information about the actual size of the command, because $ac\_mem$ may be incremented while a different command (e.g., the shell) is being executed by the process.
NAME
aliases – aliases file for sendmail

SYNOPSIS
/usr/lib/aliases

DESCRIPTION
This file describes user id aliases used by /usr/lib/sendmail. It is formatted as a series of lines of the form
name: name_1, name2, name_3, ...
The name is the name to alias, and the name_n are the aliases for that name. Lines beginning with white space are continuation lines. Lines beginning with ‘#’ are comments.

Aliasing occurs only on local names. Loops can not occur, since no message will be sent to any person more than once.

After aliasing has been done, local and valid recipients who have a “.forward” file in their home directory have messages forwarded to the list of users defined in that file.

This is only the raw data file; the actual aliasing information is placed into a binary format in the files /usr/lib/aliases.dir and /usr/lib/aliases.pag using the program newaliases(1). A newaliases command should be executed each time the aliases file is changed for the change to take effect.

SEE ALSO
newaliases(1), sendmail(1M)
SENDMAIL Installation and Operation Guide.
SENDMAIL: An Internetwork Mail Router.

BUGS
Because of restrictions in dbm(3B) a single alias cannot contain more than about 1000 bytes of information. You can get longer aliases by “chaining”; that is, make the last name in the alias be a dummy name which is a continuation alias.
NAME
ar — archive (library) file format

SYNOPSIS
#include <ar.h>

DESCRIPTION
The archive command ar combines several files into one. Archives are used mainly as libraries to be searched by the link-editor ld.

A file produced by ar has a magic string at the start, followed by the constituent files, each preceded by a file header. The magic number and header layout as described in the include file are:

#define ARMAG "!<arch>
#define SARMAG 8
#define ARFMAG "\n"

struct ar_hdr
{
  char       ar_name[16];
  char       ar_date[12];
  char       ar_uid[6];
  char       ar_gid[6];
  char       ar_mode[8];
  char       ar_size[10];
  char       ar_fmag[2];
};
typedef struct ar_hdr ARHDR;

The name is a blank-padded string. The ar_fmag field contains ARFMAG to help verify the presence of a header. The other fields are left-adjusted, blank-padded numbers. They are decimal except for ar_mode, which is octal. The date is the modification date of the file at the time of its insertion into the archive.

Each file begins on a even (0 mod 2) boundary; a new-line is inserted between files if necessary. Nevertheless the size given reflects the actual size of the file exclusive of padding.

There is no provision for empty areas in an archive file.

The encoding of the header is portable across machines. If an archive contains printable files, the archive itself is printable.

SEE ALSO
ar(1), ld(1), nm(1)
BUGS

File names lose trailing blanks. Most software dealing with archives takes even an included blank as a name terminator.
NAME
autologin – set autologin user identity

SYNOPSIS
/etc/autologin

DESCRIPTION
/etc/autologin is an ASCII file containing the login user name to be used when autologin is enabled. The file is used by getty(1M) and login(1) when automatically initiating a terminal session on the graphics console.

FILES
/etc/autologin

SEE ALSO
NAME
cftime – language specific strings

DESCRIPTION
The programmer can create one printable file per language. These files must be kept in a special directory /lib/cftime. If this directory does not exist, the programmer should create it. The contents of these files are:

• abbreviated month names (in order)
• month names (in order)
• abbreviated weekday names (in order)
• weekday names (in order)
• default strings that specify formats for local time (%x) and local date (%X).
• default format for cftime, if the argument for cftime is zero or null.
• AM (ante meridian) string
• PM (post meridian) string

Each string is on a line by itself. All white space is significant. The order of the strings in the above list is the same order in which the strings appear in the file shown below.

EXAMPLE
/lib/cftime/usaenglish
Jan
Feb
...
January
February
...
Sun
Mon
...
Sunday
Monday
...
%H:%M:%S
%m/%d/%y
%a %b %d %T %Z %Y
AM
PM

FILES

/lib/cftime – directory that contains the language specific printable files
(create it if it does not exist)

SEE ALSO

NAME

core – format of core image file

SYNOPSIS

#include <core.out.h>

DESCRIPTION

The IRIX system writes out a core image of a terminated process when any of various errors occur. See signal(2) for the list of reasons; the most common are memory violations, illegal instructions, bus errors, and user-generated quit signals. The core image is called core and is written in the process’s working directory (provided it can be; normal access controls apply). A process with an effective user ID different from the real user ID will not produce a core image.

The format of the core image is defined by <core.out.h>. It consists of a header, maps, descriptors, and section-data.

The header data includes the process name (as in ps(1)), the signal that caused the core-dump, the descriptor array, and the corefile location of the map array.

Each descriptor defines the length of useful process data. One descriptor defines the general-purpose registers at the time of the core-dump for example. The data is present in the core image at the file-location given in the descriptor only if the IVALID flag is set in the descriptor.

Each map defines the virtual address and length of a section-of-the-process at the time of the core-dump. The data is present in the core image at the file-location given in the descriptor only if the VDUMPED flag is set in the map. The process’ stack, and data sections are normally written in the core image. The process’ text is not normally written in the core image.

NOTE

Core image format designed by Silicon Graphics, Inc.

SEE ALSO

dedge(1), dbx(1), ps(1), setuid(2), signal(2).
NAME

cpio – format of cpio archive

DESCRIPTION

The header structure, when the -e option of cpio(1) is not used, is:

```c
struct {
    short    h_magic,
              h_dev;
    ushort   h_ino,
              h_mode,
              h_uid,
              h_gid;
    short    h_nlink,
              h_rdev,
              h_mtime[2],
              h_namesize,
              h_filesize[2];
    char     h_name[h_namesize rounded to word];
} Hdr;
```

When the -e option is used, the header information is described by:

```c
  sscanf(Chdr,"%60%60%60%60%60%60%60%60%11lo%60%11lo%s",
       &Hdr.h_magic, &Hdr.h_dev, &Hdr.h_ino, &Hdr.h_mode,
       &Hdr.h_uid, &Hdr.h_gid, &Hdr.h_nlink, &Hdr.h_rdev,
       &Longtime, &Hdr.h_namesize,&Longfile,Hdr.h_name);
```

Longtime and Longfile are equivalent to Hdr.h_mtime and Hdr.h_filesize, respectively. The contents of each file are recorded in an element of the array of varying length structures, archive, together with other items describing the file. Every instance of h_magic contains the constant 070707 (octal). The items h_dev through h_mtime have meanings explained in stat(2). The length of the null-terminated path name h_name, including the null byte, is given by h_namesize.

The last record of the archive always contains the name TRAILER!!!. Special files, directories, and the trailer are recorded with h_filesize equal to zero.

SEE ALSO

stat(2),

NAME
cshrc – system-wide csh initialization command file

DESCRIPTION
The file /etc/cshrc contains a list of commands to be invoked whenever a user logs into the system with csh(1) as their login shell. These commands are executed before those in the .cshrc and .login files in the home directory of the user.

FILES
/etc/cshrc

SEE ALSO
csh(1).
NAME
dbg, debug – the debug file system

SYNOPSIS
#include <sys/types.h>
#include <sys/fs/dbfcntl.h>

DESCRIPTION
The debug file system, normally mounted under /debug, provides an interface to running processes that may be used by debuggers such as dbx. The "files" of this file system are of the form /debug/<pid>, where <pid> is the process id of a running process. These files actually consume no disk space, and are only convenient handles by which a debugger can attach to a process. The debugger does so by opening the desired /debug file with the open(2) system call. The debugger may perform various fcntl(2) commands to the process; for example, to suspend and restart the process. When the process is suspended, ordinary seek(2), read(2), and write(2) system calls will access the process' address space.

The statfs(2) system call will return valid information concerning the dbg file system. The total and free blocks as reported by df(1) respectively represent the total virtual memory (real memory plus swap space) available and currently free.

The following fcntl(2) codes are recognized by dbg files. An optional argument may be supplied to the call; the form of the argument varies with the request and will be described where appropriate.

DFCSTOP
Suspend the process as soon as possible. This usually leaves the process in the stopped state; this can be verified with ps(1). However, sleeping processes remain on the sleep queue and are marked as suspended, and are therefore can be considered stopped.

DFCWSTOP
Wait for the process to stop. This is useful after setting breakpoints, system call trace masks, etc.

DFCRUN
Resume process execution after process is suspended. The argument specifies how pending signals are to be treated. If this is CLEARNO-SIG, no signals are cleared; if this is CLEARCURSIG, only the current signal is cleared; or if this is CLEARALLSIG, all pending signals are cleared. If the value is between 1 and NSIG inclusive, clear all pending signals and continue with the given signal.
DFCSSTEP
Resume process execution, but stop after executing a single machine instruction. The argument is the same as that of the DFCRUN request.

DFCCSIG
Clear the highest signal pending. This request takes no arguments. (This request is really not too useful, since the argument to DFCRUN or DFCSSTEP provides for more precise signal control.)

DFCKILL
Send a signal to the process. The argument is a an unsigned int representing the signal number.

DFCSMASK
Sets the signal trace mask. The argument is a pointer to a long indicating whether the process should stop upon receiving specified signals. To trace signal s, set (1<<(s-1)) in the signal mask.

DFCGMASK
Retrieve the signal trace mask. The argument is a pointer to a long in which to store the current signal mask.

DFCSENTRYMASK
Sets the system call entry mask. The argument is a pointer to an array of SYSMASKLEN long values indicating whether the process should stop prior to executing certain system calls. For example, to trace system call syscallno the code would resemble something like:

```c
#include <sys/user.h> /* defines SYSMASKLEN */
#include <sys/fs/dbfcntl.h>
#define BITSPERLONG 32
long entrymask[SYSMASKLEN]

entrymask[syscallno/BITSPERLONG] |=
        (1 << (syscallno%BITSPERLONG));
fcntl(fd, DFCSENTRYMASK, &entrymask);
```

Note that the system call numbers as defined in /usr/include/sys.s are relative to a large offset, SYSVoffset. This offset should be subtracted from the actual system call number before setting the bit in the mask.
DFCGENTRYMASK
Sets the system call exit mask. As above, the argument is a pointer to an array of SYSMASKLEN long values; however, this indicates that the process should stop immediately after executing specified system calls.

DFCABORT
Abort the system call in progress. If the process is stopped prior to executing a system call (see DFCGENTRYMASK above), this request will cause the signal call to return prematurely with an EINTR error code when the process is resumed.

DFCSEXEC
Have the process stop after an exec(2) system call. This is useful for stopping the process to set initial breakpoints.

DFCSEXEC
No longer stop the process after an exec(2) system call.

DFCGGETREGS
Retrieve the processor registers and signal handlers. The argument should be a pointer to an array of NPTRC_REGS unsigned integers. The general purpose processor registers, floating point registers, signal handlers, and special purpose control registers will be deposited in this array. NPTRC_REGS, as well as a layout of the registers in the array, is defined in the header file /usr/include/sys/ptrace.h.

DFCPUTREGS
Write the processor registers. The argument should be a pointer to an array of NPTRC_REGS unsigned integers. The contents of the array are copied to the appropriate register locations. Note that the signal handlers and read-only control registers are not modified.

DFCGGETPRINFO
Retrieve some useful information about the process. The argument to this request should be a pointer to a buffer of sizeof(struct procinfo) bytes. This structure, defined in /usr/include/sys/fs/dbfcntl.h, contains useful information concerning the process status.

DFCGGTRAN
Get the entire contents of the process table entry corresponding to the process. The argument supplied should be a pointer to a buffer of at least sizeof(struct proc) bytes. (Due to the lack of portability of this call, the DFCGGETPRINFO request is preferred.)
DFCOPENT
If the given argument is zero, return the file descriptor corresponding to the text region of the process. If the given argument is non-zero, interpret this value as a virtual address of the process. Return the file descriptor corresponding to the region containing this address. This call may be used to locate symbol tables of the process.

DFCEXCLU
If the given argument is zero, make the text region "private" and writable. Otherwise, make the region corresponding to the given address private. (Note: the first write to a read-only region will have the same affect and thus this call is usually not necessary.)

DIAGNOSTICS
These return codes are relevant to most system calls applied to dbg files.

[ENOENT] The process no longer exists or is in the zombie state.

[EACCES] The caller does not have permission to trace the process.

[EIO] The process should be stopped before attempting this call.

[EINVAL] An argument to this call was invalid.

[EFAULT] An address supplied by the user was invalid.

[EPERM] This operation is not allowed on a dbg file.

[EROFS] The dbg filesystem was mounted read only.

[ENOMEM] The kernel could not allocate enough memory to perform this request.

SEE ALSO
dbx(1), fcntl(2)
NAME

dir – format of EFS directories

SYNOPSIS

#include <sys/fs/efs_dir.h>

DESCRIPTION

A directory behaves exactly like an ordinary file, save that no user may
write into a directory. The fact that a file is a directory is indicated by a bit
in the flag word of its i-node entry [see inode(4)]. The EFS directory for-
mat supports variable length names of up to 255 characters.

DIRECTORY BLOCKS

Each EFS directory is segmented into directory blocks defined by the fol-
lowing data structure:

#define EFS_DIRBLK_HEADERSIZE 4
struct efs_dirblk {
    /* begin header */
    ushort magic;
    uchar firstused;
    uchar slots;
    /* end header */

    /* rest is space for efs_dent’s */
    uchar space[EFS_DIRB_SIZE - EFS_DIRBLK_HEADERSIZE];
};

Each directory block is subdivided into three separate areas: a header, an
array of entry offsets and an array of directory entries. The system restricts
directory entries to short boundaries and stores offsets in the directory block
compacted by shifting them right by one.

The header area contains a magic number to identify the block as being a
directory block. If the magic number is incorrect, the operating system will
refuse to manipulate the directory, thus avoiding further corruption.

The array of entry offsets immediately follows the header and is sized
according to the directories contents and contains compacted offsets which
point to each directory entry. The number of entry offsets available is kept
in slots. The firstused field contains a compacted offset which positions the
first byte of the directory entries.

The space between the end of the entry array and the beginning of the
directory entries (firstused) is free space which the system uses for allocat-
ing new directory entries and entry offsets. The system keeps the free space
in a directory block compacted by coalescing holes created by entry remo-
val. When a directory entry is removed, the system adjusts the entry offsets
for all entries that move. Also, the entry offset for the removed entry is zeroed. If the removed entry was the last in the entry offset array, the number of slots is reduced. Directory entries never change which entry offset they use.

DIRECTORY ENTRIES
Directory entries have the following structure:

```c
struct efs_dent {
    union {
        ulong l;
        ushort s[2];
    } ud_inum;
    unchar d_nameLEN;
    char d_name[3];
};
```

The `d_name` field is actually of variable size, depending upon the value contained in `d_nameLEN`. The system pads out the directory entry to insure that it begins on a short boundary in the directory block. The `ud_inum` field contains the entries inode number.

SEE ALSO
`fs(4), inode(4)`.
NAME
dirent — file system independent directory entry

SYNOPSIS
#include <sys/dirent.h>

DESCRIPTION
Different file system types may have different directory entries. The dirent
structure defines a file system independent directory entry, which contains
information common to directory entries in different file system types. A
set of these structures is returned by the getdents(2) system call.

The dirent structure is defined below.

    struct dirent {
        long       d_ino;
        off_t      d_off;
        unsigned short d_reclen;
        char       d_name[1];
    };

The d_ino is a number which is unique for each file in the file system. The
field d_off is an opaque offset (i.e., not necessarily in bytes) of the next
directory entry in the actual file system directory. The field d_name is the
beginning of the character array giving the name of the directory entry.
This name is null terminated and may have at most MAXNAMLEN charac-
ters. This results in file system independent directory entries being variable
length entities. The value of d_reclen is the record length of this entry. This
length is defined to be the number of bytes between the current entry and
the next one, so that it will always result in the next entry being on a long
boundary.

SEE ALSO
directory(3X), getdents(2).
NAME

filehdr – file header for MIPS object files

SYNOPSIS

#include <filehdr.h>

DESCRIPTION

Every MIPS object file begins with a 20-byte header. The following C
struct declaration is used:

struct filehdr
{
    unsigned short   f_magic;    /* magic number */
    unsigned short   f_nscns;    /* number of sections */
    long             f_timdat;   /* time & date stamp */
    long             f_symptr;   /* file pointer to symbolic header */
    long             f_nsyms;    /* sizeof(symbolic header) */
    unsigned short   f_opthdr;   /* sizeof(optional header) */
    unsigned short   f_flags;    /* flags */
};

F_symptr is the byte offset into the file at which the symbolic header can be
found. Its value can be used as the offset in fseek(3S) to position an I/O
stream to the symbolic header. The UMIPS system optional header is 56-
bytes. The valid magic numbers are given below:

#define MIPSEBMAGIC  0x0160 /* objects for MIPS big-endian machines */
#define MIPSELF magic 0x0162 /* objects for MIPS little-endian machines */
#define MIPSEBUMAGIC 0x0180 /* unicode objects for MIPS*/

/* big-endian machines */
#define MIPSELFUMAGIC 0x0182 /* unicode objects for MIPS*/
/* little-endian machines */

MIPS object files can be loaded and examined on machines differing from
the object’s target byte sex. Therefore, for object file magic numbers, the
byte swapped values have define constants associated with them:

#define SMIPSEBMAGIC  0x6001
#define SMIPSELF MAGIC 0x6201

The value in f_timdat is obtained from the time(2) system call. Flag bits
used in MIPS objects are:

#define F_RELFLG  0000001 /* relocation entries stripped */
#define F_EXEC    0000002 /* file is executable */
#define F_LNNO    0000004 /* line numbers stripped */
#define F_LSYMS   0000010 /* local symbols stripped */
SEE ALSO

time(2), fseek(3), a.out(4).
NAME
efs — layout of the Extent file system

SYNOPSIS
#include <sys/param.h>
#include <sys/fs/efs.h>

DESCRIPTION
Every Extent file system storage volume has a common format for certain
vital information. Every such volume is divided into a certain number of
512 byte long sectors, also called basic blocks. Basic block 0 is unused and
is available to contain a bootstrap program or other information.

Basic block 1 is the super-block. The format of an Extent file system
super-block is:

/*
 * Structure of the super-block for the Extent file system
 */
struct efs {

  /*
   * This portion is read off the volume
   */
  long fs_size;          /* size of file system, in sectors */
  long fs_firstcg;       /* bb offset to first cg */
  long fs_cgfsize;       /* size of cylinder group in bb’s */
  short fs_cgisize;      /* bb’s in inodes per cylinder group */
  short fs_sectors;      /* sectors per track */
  short fs_heads;        /* heads per cylinder */
  short fs_ncg;          /* # of groups in file system */
  short fs_dirty;        /* fs needs to be fsck’d */
  time_t fs_time;        /* last super-block update */
  long fs_magic;          /* magic number */
  char fs_fname[6];      /* file system name */
  char fs_fpack[6];      /* file system pack name */
  long fs_bmsize;        /* size of bitmap in bytes */
  long fs_tfree;         /* total free data blocks */
  long fs_tinode;        /* total free inodes */
  long fs_bmblock;       /* bitmap location */
  long fs_replsb;        /* location of replicated superblock. */
  char fs_spare[24];     /* space for expansion */
  long fs_checksum;      /* checksum of volume portion of fs */

  /*
   * The remainder of this structure, defined fully in <sys/fs/efs_sb.h>
   * is used by the operating system only.
   */


Note that the struct efs that is defined in `<sys/fs/efs_sb.h>` contains more fields. The extra fields are used internally by the operating system, and are not discussed here. `fs_size` holds the size in basic blocks of the file system. This variable is filled in when the file system is first created with `mkfs(1M).

`fs_firstcrg` contains the basic block offset to the first cylinder group. There are `fs_ncg` cylinder groups contained in the file system. Each cylinder group is composed of `fs_cgfsiz` basic blocks, of which `fs_cgsiz` basic blocks are used for inodes.

`fs_sectors`, and `fs_heads` are used to specify the geometry of the underlying disk containing the file system. Note that `fs_heads` is in fact currently unused, and should not be relied upon.

`fs_dirty` is a flag which indicates if the file system needs to be checked by the `fsck(1M)` program. The `fs_time` field contains the time stamp of when the file system was last modified. `fs_name` holds the name of the file system (where it is mounted, more or less) while `fs_fpack` contains which volume this file system is. The `fs_fpass` field is singularly useless, but is provided for utility compatibility. `fs_magic` is used to tag the superblock of the file system as an Extent file system.

The `fs_bmsiz` field contains, in bytes, the size of the data block bitmap. The data block bitmap is used for data block allocation. Each one in the bitmap indicates a free block. The `fs_bmblock` field contains the location of the bitmap if it has been moved from its default location (basic block 2) because the file system has been constructed on a logical volume which has been extended (see `growfs(1M)`).

`fs_ifree` and `fs_tinode` contain the total free blocks and inodes, respectively. The `fs_repsb` field contains the location of a replicated superblock, if one exists.

The `fs_spare` field is reserved for future use.

Lastly, the `fs_checksum` variable holds a checksum of the above fields (not including itself).

During the `mount(1M)` of the file system, the `fs_dirty` and `fs_checksum` fields are examined. If `fs_dirty` is non-zero, or the `fs_checksum` variable does not match the systems computed checksum, then the file system must be cleaned with `fsck` before it can be mounted. If the file system is the root partition, then this check is ignored, as it is necessary to be able to run `fsck` on a dirty root from a dirty root. For the format of an inode and its flags, see `inode(4)`.
FILES
/usr/include/sys/fs/efs*.h
/usr/include/sys/stat.h

SEE ALSO
fsck(1M), mkfs(1M), inode(4).
NAME

fspec — format specification in text files

DESCRIPTION

It is sometimes convenient to maintain text files on the UNIX system with
non-standard tabs, (i.e., tabs which are not set at every eighth column).
Such files must generally be converted to a standard format, frequently by
replacing all tabs with the appropriate number of spaces, before they can be
processed by UNIX system commands. A format specification occurring in
the first line of a text file specifies how tabs are to be expanded in the
remainder of the file.

A format specification consists of a sequence of parameters separated by
blanks and surrounded by the brackets <: and :>. Each parameter consists
of a keyletter, possibly followed immediately by a value. The following
parameters are recognized:

\texttt{tabs} The t parameter specifies the tab settings for the file. The
value of \texttt{tabs} must be one of the following:

1. a list of column numbers separated by commas, indicating tabs set at the specified columns;
2. a – followed immediately by an integer \textit{n}, indicating tabs at intervals of \textit{n} columns;
3. a – followed by the name of a “canned” tab specification.

Standard tabs are specified by \texttt{t-8}, or equivalently, \texttt{t\textbf{1,9,17,25}}, etc. The canned tabs which are recognized are defined by the \texttt{tabs(1)} command.

\texttt{ssize} The s parameter specifies a maximum line size. The value of
\texttt{size} must be an integer. Size checking is performed after tabs have been expanded, but before the margin is prepended.

\texttt{mmargin} The m parameter specifies a number of spaces to be
prepended to each line. The value of \texttt{margin} must be an
integer.

\texttt{d} The d parameter takes no value. Its presence indicates that
the line containing the format specification is to be deleted
from the converted file.

\texttt{e} The e parameter takes no value. Its presence indicates that the
current format is to prevail only until another format
specification is encountered in the file.
Default values, which are assumed for parameters not supplied, are t–8 and m0. If the s parameter is not specified, no size checking is performed. If the first line of a file does not contain a format specification, the above defaults are assumed for the entire file. The following is an example of a line containing a format specification:

* ::=t5,10,15 s72::*

If a format specification can be disguised as a comment, it is not necessary to code the d parameter.

SEE ALSO
NAME
fstab – static information about filesystems

DESCRIPTION
The file `/etc/fstab` describes the filesystems used by the local machine. The
system administrator can modify it with a text editor. It is read by com-
mands that mount, unmount and check the consistency of filesystems. The
file consists of a number of lines of the form:

    filesystem directory type options frequency pass

For example:

    /dev/root  /  efs  rw  0  0

Fields are separated by white space; a `#` as the first non-white character
indicates a comment.

The entries from this file are accessed using the routines in `getmntent(3)`,
which returns a structure of the following form:

```c
struct mntent {
    char  *mnt_fsname;  /* filesystem name */
    char  *mnt_dir;     /* filesystem path prefix */
    char  *mnt_type;    /* efs, nfs, dbg, or ignore */
    char  *mnt_opts;    /* rw, ro, hard, soft */
    int   mnt_freq;    /* dump frequency, in days */
    int   mnt_passno;  /* parallel fsck pass number */
};
```

This structure is defined in the `<mntent.h>` include file. To compile and
link a program that calls `getmntent(3)`, follow the procedures for section
(3Y) routines as described in `intro(3)`.

The `mnt_dir` field is the full path name of the directory to be mounted on.
The `mnt_type` field determines how the `mnt_fsname` and `mnt_opts` fields will
be interpreted. Here is a list of the filesystem types currently supported, and
the way each of them interprets these fields:

- **efs**  `mnt_fsname` must be a block special device (e.g., `/dev/root`).
- **dbg**  `mnt_fsname` should be the `/debug` directory. See `dbg(4)`.
- **nfs**  `mnt_fsname` is the path on the server of the directory to be
  served. (NFS option only).

If the `mnt_type` is specified as `ignore`, then the entry is ignored. This is use-
ful to show disk partitions not currently used. `mnt_freq` and `mnt_passno`
are not supported.
The `mnt_opts` field contains a list of comma-separated option words. Some `mnt_opts` are valid for all filesystem types, while others apply to a specific type only.

Options valid on `efs` and `nfs` filesystems (the default is `rw`):

- **rw** read/write.
- **ro** read-only.
- **noauto** ignore this entry during a `mount` -a command, to allow the definition of `fstab` entries for commonly-used filesystems that should not be automatically mounted.
- **grpid** causes a file created within the filesystem to have the group-ID of its parent directory, not the creating process's group-ID.

Options specific to `efs` filesystems (the default is `fsck`):

- **raw=path** the filesystem's raw device pathname (e.g. `/dev/rroot`).
- **fsck** `fsck(1M)` invoked with no filesystem arguments should check this filesystem.
- **nofsck** `fsck(1M)` should not check this filesystem by default.
- **lsize=n** the number of bytes transferred in each read or synchronous write operation.

The value assigned to the `lsize` option must be a power of two at least as large as `NPBC` (as defined in `/usr/include/sys/param.h`), and no larger than 64K. The current default for `lsize` is the largest power of two less than or equal to the size of one disk track. An invalid size will cause the mount to fail with the error `EINVAL`. Note that less than `lsize` bytes will be transferred if there are not `lsize` contiguous bytes of the addressed portion of the file on disk.

If the NFS option is installed, the following options are valid for `nfs` filesystems:

- **bg** if the first attempt fails, retry in the background.
- **fg** retry in foreground. (Default)
- **retry=n** set number of mount failure retries to `n`. (Default = 10000)
- **rsz=n** set read buffer size to `n` bytes. (Default = 8K)
wsizen  set write buffer size to \( n \) bytes. (Default = 8K)
timeon  set NFS timeout to \( n \) tenths of a second. (Default = 7)
retransn  set number of NFS retransmissions to \( n \). (Default = 4)
portn  set server UDP port number to \( n \). (Default = 2049)
hard  retry request until server responds. (Default)
soft  return error if server doesn’t respond.
intr  allow hard mounts to be interrupted by uncaught fatal signals. (Default)
nointr  don’t allow hard mounts to be interrupted.
acregminn  t  set the regular file minimum attribute cache timeout to \( t \) seconds. (Default = 3)
acregmaxn  t  set the regular file maximum attribute cache timeout to \( t \) seconds. (Default = 60)
acdirminn  t  set the directory minimum attribute cache timeout to \( t \) seconds. (Default = 30)
acdirmaxn  t  set the directory maximum attribute cache timeout to \( t \) seconds. (Default = 60)
actimeon  t  set regular and directory minimum and maximum attribute cache timeouts to \( t \) seconds.
noac  no attribute caching.
private  do not flush delayed writes on last close of an open file, and use local file and record locking instead of a remote lock manager.

The \texttt{bg} option causes \texttt{mount} to run in the background if the server’s \texttt{mountd}(1M) does not respond. \texttt{Mount} attempts each request \texttt{retry=n} times before giving up.

Once the filesystem is mounted, each \texttt{nfs} request made waits \texttt{timeo=n} tenths of a second for a response. If no response arrives, the time-out is multiplied by 2 and the request is retransmitted. When \texttt{retrans=n} retransmissions have been sent with no reply a \texttt{soft} mounted filesystem returns an error on the request and a \texttt{hard} mounted filesystem retries the request. Filesystems that are mounted \texttt{rw} (read-write) should use the \texttt{hard} option. The number of bytes in a read or write request can be set with the \texttt{rsize} and \texttt{wsize} options.
In the absence of client activity that would invalidate recently acquired file attributes, NFS holds attributes cached for an interval between acregmin and acregmax for regular files, and between acdirmin and acdirmax for directories. The actimeo option sets all attribute timeout constraints to a given number of seconds. The noac option disables attribute caching altogether.

The private option greatly improves write performance by caching data and delaying writes on the assumption that only this client modifies files in the remote filesystem. It should be used only if the greater risk of lost delayed-write data in the event of a crash is acceptable given better performance. Note that EFS uses caching strategies similar to private NFS, and that the system reduces the risk of data loss for all filesystems by automatically executing a partial sync(2) at regular intervals.

NOTES

The default fstab supplied with SGI systems contains the following entry for the /usr filesystem:

```
/dev/usr /usr efs rw,raw=/dev/rusr 0 0
```

The setup program MAKEDEV (see makedev(1M)) creates /dev/usr and /dev/rusr as links to partition 6 on the root disk. This is the normal disk usage; however, if you wish to set up a machine with the /usr filesystem residing elsewhere (for example, on a second disk or on a logical volume, described in lv(7M)), the mnt_fstname field must be changed to the full pathname of the device where the /usr filesystem actually resides. If present, the path specified by the raw option should also be changed to the corresponding full pathname. For example:

```
/dev/dsk/ips0d1s7 /usr efs rw,raw=/dev/rdsk/ips0d1s7 0 0
```

Note that if this is done, the /dev/usr and /dev/rusr devices created by MAKEDEV will not point to the device containing the /usr filesystem, and they should not be referenced.

Caution: do not attempt to reconfigure a system with /usr in a non-default volume by manually recreating these /dev/usr and /dev/rusr links and leaving the fstab entry unchanged. While this would work in normal operation, it will lead to incorrect behaviour when installing new software.

FILES

/etc/fstab

SEE ALSO

fsck(1M), mount(1M), mtab(4)
getmntent(3) if the NFS option is installed.
Extensions by Silicon Graphics, Inc.
NAME

gettydefs – speed and terminal settings used by getty

DESCRIPTION

The /etc/gettydefs file contains information used by getty(1M) to set up the speed and terminal settings for a line. It supplies information on what the login(1) prompt should look like. It also supplies the speed to try next if the user indicates the current speed is not correct by typing a <break> character.

NOTE: Customers who need to support terminals that pass 8 bits to the system (as is typical outside the U.S.A.) must modify the entries in /etc/gettydefs as described in the WARNINGS section.

Each entry in /etc/gettydefs has the following format:

```
label# initial-flags # final-flags # login-prompt #next-label
```

Each entry is followed by a blank line. The various fields can contain quoted characters of the form \b, \n, \e, etc., as well as \nnn, where nnn is the octal value of the desired character. The various fields are:

- **label**
  - This is the string against which getty(1M) tries to match its second argument. It is often the speed, such as 1200, at which the terminal is supposed to run, but it need not be (see below).

- **initial-flags**
  - These flags are the initial ioctl(2) settings to which the terminal is to be set if a terminal type is not specified to getty(1M). The flags that getty(1M) understands are the same as the ones listed in /usr/include/sys/termio.h [see termio(7)]. Normally only the speed flag is required in the initial-flags. getty(1M) automatically sets the terminal to raw input mode and takes care of most of the other flags. The initial-flag settings remain in effect until getty(1M) executes login(1).

- **final-flags**
  - These flags take the same values as the initial-flags and are set just before getty(1M) executes login(1). The speed flag is again required. The composite flag SANE takes care of most of the other flags that need to be set so that the processor and terminal are communicating in a rational fashion. The other two commonly specified final-flags are TAB3, so that tabs are sent to the terminal as spaces, and HUPCL, so that the line is hung up on the final close.
login-prompt  This entire field is printed as the login-prompt. Unlike the above fields where white space is ignored (a space, tab or new-line), they are included in the login-prompt field.

next-label  If this entry does not specify the desired speed, indicated by the user typing a <break> character, then getty(1M) will search for the entry with next-label as its label field and set up the terminal for those settings. Usually, a series of speeds are linked together in this fashion, into a closed set; for instance, 2400 linked to 1200, which in turn is linked to 300, which finally is linked to 2400.

If getty(1M) is called without a second argument, then the first entry of /etc/gettydefs is used, thus making the first entry of /etc/gettydefs the default entry. It is also used if getty(1M) can not find the specified label. If /etc/gettydefs itself is missing, there is one entry built into getty(1M) which will bring up a terminal at 300 baud.

It is strongly recommended that after making or modifying /etc/gettydefs, it be run through getty(1M) with the check option to be sure there are no errors.

FILES

/etc/gettydefs

SEE ALSO


WARNINGS

To support terminals that pass 8 bits to the system (also, see the BUGS section), modify the entries in the /etc/gettydefs file for those terminals as follows: add CS8 to initial-flags and replace all occurrences of SANE with the values: BRKINT IGNPAR ICRNL IXON OPOST ONCLR CS8 ISIG ICANON ECHO ECHOK

An example of changing an entry in /etc/gettydefs is illustrated below. All the information for an entry must be on one line in the file.

Original entry:

    CONSOLE # B9600 HUPCL OPOST ONLCR # B9600 SANE IXANY TAB3 HUPCL # Console Login: # console

Modified entry:

    CONSOLE # B9600 CS8 HUPCL OPOST ONLCR # B9600 BRKINT IGNPAR ICRNL IXON OPOST ONLCR CS8 ISIG ICANON ECHO ECHOK IXANY TAB3 HUPCL # Console Login: # console
This change will permit terminals to pass 8 bits to the system so long as the system is in MULTI-USER state. When the system changes to SINGLE-USER state, the getty(1M) is killed and the terminal attributes are lost. So to permit a terminal to pass 8 bits to the system in SINGLE-USER state, after you are in SINGLE-USER state, type (see stty(1)):

```
stty -istrip cs8
```

**BUGS**

8-bit with parity mode is not supported.
NAME

group – group membership file

DESCRIPTION

The /etc/group file contains for each group the following information:

- group name
- encrypted password
- numerical group ID
- a comma separated list of all users allowed in the group

For example, the entry for the sys group is:

```
sys::0:root,bin,sys,adm
```

This is an ASCII file. The fields are separated by colons; each group is separated from the next by a new-line. If the password field is null, no password is demanded. A "*" in the password field locks the entry.

This file resides in the /etc directory. Because of the encrypted passwords, it can and does have general read permission and can be used, for example, to map numerical group ID’s to names.

YELLOW PAGES

If the NFS option is installed, a group file can have a line beginning with a plus (+), which means to incorporate entries from the Yellow Pages. There are two styles of + entries: All by itself, + means to insert the entire contents of the Yellow Pages group file at that point; +name means to insert the entry (if any) for name from the Yellow Pages at that point. If a + entry has a non-null password or group member field, the contents of that field will override what is contained in the Yellow Pages. The numerical group ID field cannot be overridden.

A group file can also have a line beginning with a minus (−), these entries are used to disallow group entries. There is only one style of − entries: an entry that consists of −name means to disallow any subsequent entry (if any) for name. These entries will be disallowed regardless of whether the subsequent entry comes from the Yellow Pages or the local group file.

For example, if the following entries

```
−oldproj
+myproject::bill, steve
+
```

appear at the end of a group file, then the group oldproj will be ignored if it appears after the entry −oldproj. Also, the group myproject will have members bill and steve, and the password and group ID of the Yellow Pages entry for the group myproject. All the groups listed in the Yellow Pages will be pulled in and placed after the entry for myproject.
FILE

/etc/group

SEE ALSO

crypt(3), newgrp(1M), passwd(1), passwd(4)

BUGS

The passwd(1) command won't change group passwords.
Sun Microsystems (YP version)
NAME
hosts – host name data base

DESCRIPTION
The /etc/hosts file contains information regarding the known hosts on the
network. For each host a single line should be present with the following
information:

Internet address
official host name
aliases

Items are separated by any number of blanks and/or tab characters. A ‘‘#’’
indicates the beginning of a comment; characters up to the end of the line
are not interpreted by routines which search the file.

This file must include entries for the machine’s network interfaces, the
‘‘localhost’’ address and a few important machines on the local network.
ifconfig(1M) uses this file when assigning addresses to the network inter-
faces.

By default, this file is used by gethostbyname(3N) and gethostbyaddr(3N)
only when the Yellow Pages or the Berkeley name server (named(1M)) are
not enabled. The system can be configured to use YP, named, and/or this
file, as described in resolver(4) and the Network Communications Guide.

If the host is not connected to any network, the file should contain an entry
defining the hostname as an alias for the ‘‘localhost’’ entry. For example, if
the hostname is IRIS, the /etc/hosts file should contain this line:

    127.1 localhost IRIS

Sites connected to the Internet should configure the system to use the name
server. This file may be created from the official host data base maintained
at the Network Information Center (NIC), though local changes may be
required to bring it up to date regarding unofficial aliases and/or unknown
hosts. The host data base maintained at NIC is incomplete.

Network addresses are specified in the conventional ‘‘.’’ (dot) notation
using the inet_addr() routine from the Internet address manipulation library,
inet(3N). Host names may contain any printable character other than a field
delimiter, newline, or comment character.

FILES
/etc/hosts

SEE ALSO
gethostbyname(3N), ifconfig(1M), named(1M), resolver(4), hostname(5)
The network administration chapters in the Network Communications
Guide.
NAME
hosts.equiv – list of trusted hosts

DESCRIPTION
The /etc/hosts.equiv file contains a list of trusted hosts. When an rlogin(1C), rcp(1C), rdist(1C), or rsh(1C) request from such a host is made, and the initiator of the request is in /etc/passwd, then, no further validity checking is done. That is, rlogin does not prompt for a password, and rcp, rdist, and rsh complete successfully. So a remote user is “equivalenced” to a local user with the same user name when the remote user is in hosts.equiv.

The format of hosts.equiv is a list of host names, as in this example:

bonnie.sgi.com
clyde.sgi.com

A line consisting of a simple host name means that anyone logging in from that host is trusted. Programs scan hosts.equiv linearly, and stop at the first hit.

The .rhosts file has the same format as hosts.equiv. When user XXX executes rlogin or rsh, the .rhosts file from XXX’s home directory is conceptually concatenated onto the end of hosts.equiv for permission checking. In the special case when the user is root, then only the .rhosts file is checked.

If an entry in a local user’s .rhosts file contains a remote host name and a user name separated by white space, the remote user is allowed to log in as the local user. Thus the entry

bonnie.sgi.com  faye

in warren’s .rhosts allows faye to log in from bonnie as warren. If this entry was placed in clyde’s /etc/hosts.equiv, then faye could login as anyone who is in clyde’s /etc/passwd file.

FILES
/etc/hosts.equiv
/.rhosts

SEE ALSO
rcp(1C), rdist(1C), rlogin(1C), rsh(1C), rcmd(3N), rhosts(4)
NAME

inittab – script for the init process

DESCRIPTION

The `/etc/initab` file supplies the script to `init`'s role as a general process dispatcher. The process that constitutes the majority of `init`'s process dispatching activities is the line process `/etc/getty` that initiates individual terminal lines. Other processes typically dispatched by `init` are daemons and the shell.

The `initab` file is composed of entries that are position dependent and have the following format:

```
id:rstate:action:process
```

Each entry is delimited by a newline; however, a backslash (\) preceding a newline indicates a continuation of the entry. Up to 512 characters per entry are permitted. Comments may be inserted in the `process` field using the `sh(1)` convention for comments. Comments for lines that spawn `getty`s are displayed by the `who(1)` command. It is expected that they will contain some information about the line such as the location. There are no limits (other than maximum entry size) imposed on the number of entries within the `initab` file. The entry fields are:

- `id` This field, of up to four characters, is used to uniquely identify an entry.
- `rstate` This defines the `run-level` in which this entry is to be processed. `Run-levels` effectively correspond to a configuration of processes in the system. That is, each process spawned by `init` is assigned a `run-level` or `run-levels` in which it is allowed to exist. The `run-levels` are represented by the letter s (or S), or a number ranging from 0 through 6. As an example, if the system is in `run-level` 1, only those entries having a 1 in the `rstate` field will be processed. When `init` is requested to change `run-levels`, all processes which do not have an entry in the `rstate` field for the target `run-level` will be sent the warning signal (SIGTERM) and allowed a grace period (see `init(1M)` for the length of this grace period), before being forcibly terminated by a kill signal (SIGKILL). The `rstate` field can define multiple `run-levels` for a process by selecting more than one `run-level` in any combination from 0–6, s and S. If no `run-level` is specified, then the process is assumed to be valid at all `run-levels`. There are three other values, a, b and c, which can appear in the `rstate` field, even though they are not true `run-levels`. Entries which have these characters in the `rstate` field are processed only when the `telinit` [see `init(1M)`] process requests them to be run (regardless of the current `run-level` of the system). They differ
from *run-levels* in that *init* can never enter *run-level* a, b or c. Also, a request for the execution of any of these processes does not change the current *run-level*. Furthermore, a process started by an a, b or c command is not killed when *init* changes levels. They are only killed if their line in /etc/inittab is marked off in the *action* field, their line is deleted entirely from /etc/inittab, or *init* goes into the *SINGLE USER* state.

*action* Key words in this field tell *init* how to treat the process specified in the *process* field. The actions recognized by *init* are as follows:

**respawn** If the process does not exist then start the process, do not wait for its termination (continue scanning the *inittab* file), and when it dies restart the process. If the process currently exists then do nothing and continue scanning the *inittab* file.

**wait** Upon *init*'s entering the *run-level* that matches the entry's *rstate*, start the process and wait for its termination. All subsequent reads of the *inittab* file while *init* is in the same *run-level* will cause *init* to ignore this entry.

**once** Upon *init*'s entering a *run-level* that matches the entry's *rstate*, start the process, do not wait for its termination. When it dies, do not restart the process. If upon entering a new *run-level*, where the process is still running from a previous *run-level* change, the program will not be restarted.

**boot** The entry is to be processed only at *init*'s boot-time read of the *inittab* file. *Init* is to start the process, not wait for its termination; and when it dies, not restart the process. In order for this instruction to be meaningful, the *rstate* should be the default or it must match *init*'s *run-level* at boot time. This action is useful for an initialization function following a hardware reboot of the system.

**bootwait** The entry is to be processed the first time *init* goes from single-user to multi-user state after the system is booted. (If *initdefault* is set to 2, the process will run right after the boot.) *Init* starts the process, waits for its termination and, when it dies, does not restart the process.
powerfail

Execute the process associated with this entry only when init receives a power fail signal [SIGPWR, see signal(2)].

powerwait

Execute the process associated with this entry only when init receives a power fail signal (SIGPWR) and wait until it terminates before continuing any processing of initab.

off

If the process associated with this entry is currently running, send the warning signal (SIGTERM) and wait 20 seconds before forcibly terminating the process via the kill signal (SIGKILL). If the process is nonexistent, ignore the entry.

ondemand

This instruction is really a synonym for the respawn action. It is functionally identical to respawn but is given a different keyword in order to divorce its association with run-levels. This is used only with the a, b or c values described in the rstate field.

initdefault

An entry with this action is only scanned when init initially invoked. Init uses this entry, if it exists, to determine which run-level to enter initially. It does this by taking the highest run-level specified in the rstate field and using that as its initial state. If the rstate field is empty, this is interpreted as 0123456 and so init will enter run-level 6. Additionally, if init does not find an initdefault entry in /etc/inittab, then it will request an initial run-level from the user at reboot time.

sysinit

Entries of this type are executed before init tries to access the console (i.e., before the Console Login: prompt). It is expected that this entry will be only used to initialize devices on which init might try to ask the run-level question. These entries are executed and waited for before continuing.

process

This is a sh command to be executed. The entire process field is prefixed with exec and passed to a forked sh as sh -c 'exec command'. For this reason, any legal sh syntax can appear in the process field. Comments can be inserted with the ; #comment syntax.

NOTES

Strictly speaking, except for comments in the process field, there is no comment convention for inittab files. Note in particular that a leading # in itself does not cause a line to be treated as a comment. However, lines not in the
*id: rstate: action: process* format will be ignored by *init*.

**FILES**

/etc/init.tab

**SEE ALSO**

exec(2), open(2), signal(2).

getty(1M), init(1M) in the *System Administrator's Reference Manual*.

sh(1), who(1) in the *User's Reference Manual*.
NAME
inode – format of an Extent File System inode

SYNOPSIS
#include <sys/param.h>
#include <sys/inode.h>

DESCRIPTION
An inode is the volume data structure used by a file system to implement
the abstraction of a file. (This is not to be confused with the in-core inode
used by the operating system to manage files in use.)

An inode contains the type (e.g., plain file, directory, symbolic link, or dev-
vice file) of the file; its owner, group and public access permissions; the
owner and group id numbers; its size in bytes; the number of links (direct-
ory references) to the file; and the times of last access and last modification
to the file. In addition, there is a file system type-dependent representation
of the list of data blocks claimed by the file.

An inode under the Extent File System has the following structure.

#define EFS_DIRECTEXTENTS 12

/ *
* Extent based file system inode as it appears on disk.
* The efs inode is 128 bytes long.
*/
struct efs_dinode {
    ushort  di_mode;    /* type and access permissions */
    short   di_nlink;  /* number of links */
    ushort  di_uid;    /* owner’s user id number */
    ushort  di_gid;    /* group’s group id number */
    off_t   di_size;   /* number of bytes in file */
    time_t  di_atime;  /* time of last access (to contents) */
    time_t  di_mtime;  /* of last modification (of contents) */
    time_t  di_ctime;  /* of last modification to inode */
    long    di_gen;    /* generation number */
    short   di_numextents; /* # of extents */
    short   di-unused; /* UNUSED */
    union {
        extent  di_extents[EFS_DIRECTEXTENTS];
        dev_t   di_dev; /* device for IFCHR/IFBLK */
    } di_u;
};
The types *ushort*, *off_t*, *time_t*, and *dev_t* are defined in *types(5)*. The *extent* type is defined as follows:

```c
typedef struct extent {
    unsigned int
    ex_magic:8, /* magic #, must be 0 */
    ex_bn:24,  /* bb # on volume */
    ex_length:8, /* length of this extent in bb's */
    ex_offset:24; /* logical file offset in bb's */
} extent;
```

*di_mode* contains the type of the file (plain file, directory, etc), and its read, write, and execute permissions for the file’s owner, group, and public. *di_nlink* contains the number of links to the inode. Correctly formed directories have a minimum of two links: a link in the directory’s parent and the ‘.’ link in the directory itself. Additional links may be caused by ‘.’ links from subdirectories. *di_uid* and *di_gid* contain the user id and group id of the file (used to determine which set of access permissions apply: owner, group, or public). *di_size* contains the length of the file in bytes.

*di_atime* is the time of last access to the file’s contents. *di_mtime* is the time of last modification of the file’s contents. *di_ctime* is the time of last modification of the inode, as opposed to the contents of the file it represents. These times are given in seconds since the beginning of 1970 GMT.

*di_gen* is the inode generation number used to sequence instantiations of the inode.

An extent descriptor maps a logical segment of a file to to a physical segment (i.e., extent) on the volume. The physical segment is characterized by a starting address and a length, both in basic blocks (of 512 bytes) and a logical file offset, also in basic blocks.

*di_numextents* is the number of extents claimed by the file. If less than or equal to *EFS DIRECTEXTENTS* then the extent descriptors appear directly in the inode as *di_u.di_extents[0 .. di_numextents-1]*. When the number of extents exceeds this range, then *di_u.di_extents[0 .. di_u.di_extents[0].ex_offset-1]* are indirect extents that map blocks holding extent information. There are at most *EFS DIRECTEXTENTS* indirect extents.

If the inode is a block or character special inode, *di_u.di_numextents* is 0, and *di_u.di_dev* contains a number identifying the device.

**FILES**

/usr/include/sys/param.h
/usr/include/sys/types.h
/usr/include/sys/inode.h

April 1990 - 2 - Version 5.0
/usr/include/sys/stat.h

SEE ALSO
stat(2), fs(4), cfs(4), types(5).
NAME
ldfcn – common object file access routines

SYNOPSIS
#include <stdio.h>
#include <filehdr.h>
#include <syms.h>
#include <ldfcn.h>

DESCRIPTION
The common object file access routines are a collection of functions for reading an object file that is in common object file form. Although the calling program must know the detailed structure of the parts of the object file that it processes, the routines effectively insulate the calling program from knowledge of the overall structure of the object file.

The interface between the calling program and the object file access routines is based on the defined type LDFILE (defined as struct ldfile), which is declared in the header file <ldfcn.h>. Primarily, this structure provides uniform access to simple object files and object files that are members of an archive file.

The function ldopen(3X) allocates and initializes the LDFILE structure, reads in the symbol table header, if present, and returns a pointer to the structure to the calling program. The fields of the LDFILE structure can be accessed individually through macros defined in <ldfcn.h>. The fields contain the following information:

LDFILE *ldptr;
TYPE(ldptr) The file magic number, used to distinguish between archive members and simple object files.
IOPTR(ldptr) The file pointer returned by fopen(3S) and used by the standard input/output functions.
OFFSET(ldptr) The file address of the beginning of the object file; if the object file is a member of an archive file, the offset is non-zero.
HEADER(ldptr) The file header structure of the object file.
SYMHEADER(ldptr) The symbolic header structure for the symbol table associated with the object file.
PFD(ldptr) The file table associated with the symbol table.

SYMTAB(ldptr) A pointer to a copy of the symbol table in memory. It's accessed through the pCHDR structure (see <emplsrs/stsupport.h>). If no symbol table is present, this field is NULL. NOTE: This macro causes the whole symbol table to be read.

LDSWAP(ldptr) If the header and symbol table structures are swapped within the object file and all access requires using libsex, this field is set to true. NOTE: If you use libmld.a routines, all structures, except the optional header and auxiliaries, are swapped.

The object file access functions can be divided into five categories:

1. functions that open or close an object file

   *ldopen(3X)* and *ldaopen*

   open a common object file

   *ldclose(3X)* and *ldaclose*

   close a common object file

2. functions that return header or symbol table information

   *ldahread(3X)*

   read the archive header of a member of an archive file

   *ldfheader(3X)*

   read the file header of a common object file

   *ldshread(3X)* and *ldnshread*

   read a section header of a common object file

   *ldtbread(3X)*

   read a symbol table entry of a common object file

   *ldgetname(3X)*

   retrieve a symbol name from a symbol table entry or from the string table

   *ldgtaux(3X)*

   retrieve a pointer into the aux table for the specified ldptr

   *ldgetsymstr(3X)*

   create a type string (for example, C declarations) for the specified symbol

   *ldgetpd(3X)*

   retrieve a procedure descriptor

   *ldgetrfd(3X)*

   retrieve a relative file table entry
(3) functions that position an object file at (seek to) the start of the section, relocation, or line number information for a particular section

\textit{ldohseek}(3X)
seek to the optional file header of a common object file
\textit{ldssseek}(3X) and \textit{ldnsseek}
seek to a section of a common object file
\textit{ldrseek}(3X) and \textit{ldnrseek}
seek to the relocation information for a section of a common object file
\textit{ldlseek}(3X) and \textit{ldnlseek}
seek to the line number information for a section of a common object file
\textit{ldtseek}(3X)
seek to the symbol table of a common object file

(4) miscellaneous functions
\textit{ldtbindx}(3X)
return the index of a particular common object file symbol table entry
\textit{ranhashinit}(3X)
initialize the tables and constants so that the archive hash and lookup routines can work
\textit{ranhash}(3X)
give a string return the hash index for it
\textit{ranlookup}(3X)
return an archive hash bucket that is empty or matches the string argument
\textit{disassembler}(3X)
print MIPS assembly instructions
\textit{ldreadst}(3X)
cause section of the the symbol table to be read

These functions are described in detail in the manual pages identified for each function.

\textit{Ldopen} and \textit{ldaopen} both return pointers to a \texttt{LDFILE} structure.

MACROS

Additional access to an object file is provided through a set of macros defined in \texttt{<ldfcn.h>}. These macros parallel the standard input/output file reading and manipulating functions. They translate a reference of the \texttt{LDFILE} structure into a reference to its file descriptor field.
The following macros are provided:

GETC(ldptr)
FGETC(ldptr)
GETW(ldptr)
UNGETC(c, ldptr)
FGETS(s, n, ldptr)
FREAD((char *) ptr, sizeof (*ptr), nitems, ldptr)
FSEEK(ldptr, offset, ptrname)
FTELL(ldptr)
REWIND(ldptr)
FEOF(ldptr)
FERROR(ldptr)
FILENAME(ldptr)
SETBUF(ldptr, buf)
STROFFSET(ldptr)

The STROFFSET macro calculates the address of the local symbol’s string table in an object file. See the manual entries for the corresponding standard input/output library functions for details on the use of these macros. (The functions are identified as 3S in Section 3 of this manual.)

The program must be loaded with the object file access routine library libmld.a.

WARNINGS

The macro FSEEK defined in the header file <ldfcn.h> translates into a call to the standard input/output function fseek(3S). FSEEK should not be used to seek from the end of an archive file since the end of an archive file cannot be the same as the end of one of its object file members.

When applied to object files in an archive FSEEK (ldptr,offset,BEGINNING) uses (and FTELL returns) an offset relative to the beginning of an individual object file, not the absolute file locations used by fseek (stream,offset,0) (and returned by ftell).

SEE ALSO

Assembly Language Programmer’s Guide
ar(1), fopen(3S), fseek(3S), ldahread(3X), ldclose(3X), ldftime(3X), ldgetname(3X), ldldread(3X), ldldseek(3X), ldohseek(3X), ldopen(3X), ldresize(3X), ldldseek(3X), ldshread(3X), ldtbindex(3X), ldtbthread(3X), ldtsbseek(3X).
NAME
limits – file header for implementation-specific constants

SYNOPSIS
#include <limits.h>
#include <float.h>

DESCRIPTION
The header file <limits.h> is a list of minimum magnitude limitations imposed by the IRIX operating system. In some cases, the actual values may be greater, and can be obtained at runtime via sysconf() or pathconf() system calls (depending upon the variable desired). In this way a program can use, for example, dynamic memory allocation to utilize greater-than-default limits. See sysconf(2) and pathconf(2). <limits.h> also specifies the sizes of integral types as required by the proposed ANSI C standard.

The header file <float.h> specifies the characteristics of floating types as required by the proposed ANSI C standard. The constants that refer to long doubles (those prefixed by LDBL_) that appear in <float.h> are not specified because long doubles are not implemented.

All values in <limits.h> and <float.h> are specified in decimal.

The file <limits.h> contains:

#define ARG_MAX 10240 /* max length of arguments to exec */
define CHAR_BIT 8 /* # of bits in a "char" */
define CHAR_MAX 255 /* max integer value of a "char" */
define CHAR_MIN 0 /* min integer value of a "char" */
define CHILD_MAX 25 /* max # of processes per user id */
define CLK_TCK 100 /* # of clock ticks per second */
define DBL_DIG 15 /* digits of precision of a "double" */
define DBL_MAX 1.797693134862316e+308 /* max decimal value of a "double" */
define DBL_MIN 2.225073858507201e-308 /* min decimal value of a "double" */
define FCHR_MAX 2147483647 /* max size of a file in bytes */
define FLT_DIG 6 /* digits of precision of a "float" */
define FLT_MAX 3.40282347e+38 /* max decimal value of a "float" */
define FLT_MIN 1.17549435e-38 /* min decimal value of a "float" */
define HUGE_VAL 1.797693134862316e+308 /* error value returned by Math lib */
define INT_MAX 2147483647 /* max decimal value of an "int" */
define INT_MIN -2147483648 /* min decimal value of an "int" */
define LINK_MAX 1000 /* max # of links to a single file */
define LONG_MAX 2147483647 /* max decimal value of a "long" */
define LONG_MIN -2147483648 /* min decimal value of a "long" */
define MAX_CANON 255 /* max # of bytes in a terminal */
/* canonical input queue */
define MAX_INPUT 255 /* max # of bytes for which space will be */
POSIX additions to <limits.h>. The POSIX 1003.1 standard requires the following symbols to be defined in <limits.h>, with the values shown. These define minimum values for certain features of the system; hence no POSIX 1003.1 conforming system can provide a more restrictive value. For each of these symbols, there is an analogous symbol defined in <limits.h>, which reflects the actual implementation, and which are, in most cases, less restrictive.

#define POSIX_ARG_MAX 4096 /* Minimum value for ARG_MAX */
#define POSIX_CHILD_MAX 6 /* Minimum value for CHILD_MAX */
#define POSIX_LINK_MAX 8 /* Minimum value for LINK_MAX */
#define POSIX_MAX_CANON 255 /* Minimum value for MAX_CANON */
#define POSIX_MAX_INPUT 255 /* Minimum value for MAX_INPUT */
#define POSIX_NAME_MAX 14 /* Minimum value for NAME_MAX */
#define POSIX_NGROUPS_MAX 0 /* Minimum value for NGROUPS_MAX */
#define POSIX_OPEN_MAX 16 /* Minimum value for OPEN_MAX */
#define POSIX_PATH_MAX 255 /* Minimum value for PATH_MAX */
#define POSIX_PIPE_BUF 512 /* Minimum value for PIPE_BUF */
The file `<float.h>` contains:

```c
#define FLT_RADIX 2                  /* radix of exponent representation */
#define FLT_ROUNDS 1                  /* addition rounds (>0 implemented) */

/* number of base-FLT_RADIX digits in the floating point mantissa */
#define FLT_MANT_DIG 24
#define DBL_MANT_DIG 53

/* minimum positive floating-point number x such that 1.0 + x ≠ 1.0 */
#define FLT_EPSILON 1.17549435e-07
#define DBL_EPSILON 2.2204460492503131e-16

/* number of decimal digits of precision */
#define FLT_DIG 6
#define DBL_DIG 15

/* minimum negative integer such that FLT_RADIX raised to that */
/* power is a normalized floating point number */
#define FLT_MIN_EXP -125
#define DBL_MIN_EXP -1021

/* minimum normalized positive floating-point number */
#define FLT_MIN 1.17549435e-38
#define DBL_MIN 2.225073858507201e-308

/* minimum negative integer such that 10 raised to that power */
/* is in normalized floating-point numbers */
#define FLT_MIN_10_EXP -37
#define DBL_MIN_10_EXP -307

/* maximum integer such that FLT_RADIX raised to that power */
/* minus 1 is a representable finite floating-point number */
#define FLT_MAX_EXP 128
#define DBL_MAX_EXP 1024

/* maximum representable finite floating-point number */
#define FLT_MAX 3.40282347e+38
#define DBL_MAX 1.797693134862316e+308

/* maximum integer such that 10 raised to that power is in the */
/* range of finite floating-point numbers */
#define FLT_MAX_10_EXP 38
```
#define DBL_MAX_10_EXP 308

SEE ALSO
    sysconf(2), pathconf(2)
NAME
linenum – line number entries in a MIPS object file

SYNOPSIS
#include <sym.h>

DESCRIPTION
The `cc(1), f77(1), pc(1), and pl1(1)` commands generate an entry in the object file for each source line on which a breakpoint is possible [when any of the commands are invoked with the `-g` option]. Users can then reference line numbers when using the appropriate software test system (see `edge(1)` and `dbx(1)`). The structure of these line number entries is described in the *Assembly Language Programmer’s Guide*.

NOTE
Do not include `<linenum.h>`; the structures for dealing with line numbers in `<sym.h>` supersede those in `<linenum.h>`.

SEE ALSO
*Assembly Language Programmer’s Guide.*
`cc(1), edge(1), dbx(1), f77(1), pc(1), pl1(1), a.out(4), syms(4)`.
NAME
login – login configuration file

SYNOPSIS
/etc/config/login.options

DESCRIPTION
login.options is an ASCII file consisting of lines of the form keyword or keyword=value. Keywords can be separated by white-space or placed on separate lines. Keywords that take values must be one word with no white-space between the keyword, equals sign and value. A "#" indicates the beginning of a comment; characters up to the end of the line are ignored.

The following keywords are recognized:

maxtries=value
The number of unsuccessful attempts permitted before ending the session. 0 is "no limit". The default is 5 tries.

disabletime=value
The amount of time in seconds login waits before ending the session after maxtries unsuccessful attempts. The default is 20 seconds.

passwdreq
All accounts must have passwords. If the user does not have a password, the user will be forced to choose one before being allowed to login.

lastlog
Inform the user about the last successful login attempt. It shows the date, time and the name of the terminal or remote host from which the previous login attempt occurred.

syslog=value
Record successful and unsuccessful login attempts to syslog(3) if value is all or record unsuccessful attempts only if value is fail.

FILES
/etc/config/login.options

SEE ALSO
getty(1M),
NAME

lvtab – information about logical volumes

DESCRIPTION

The file /etc/lvtab describes the logical volumes used by the local machine. There is an entry in this file for every logical volume which will be used by the machine. It is read by commands that create, install and check the consistency of logical volumes. The system administrator can modify it with a text editor to add new logical volumes or to extend existing ones.

The file consists of entries which have the form:

    volume_device_name:[volume_name]:[options]:device_pathnames

For example:

    lv0:logical volume test:stripes=3:dev=/dev/dsk/ips0d1s7,\  
     /dev/dsk/ips0d2s7, /dev/dsk/ips0d3s7

Fields are separated by colons, and lines may be continued by the usual backslash convention as illustrated above. A '#' as the first non-white character indicates a comment; blank lines may be present in the file and will be ignored.

The fields in each entry have the following significance:

volume_device_name

This indicates the names of the special files through which the system will access the logical volume. In the above example, the entry lv0 implies that the logical volume will be accessed via the device special files /dev/dsk/lv0 and /dev/rdsk/lv0. Note that volume device names are expected to be of the form 'lv' followed by one or 2 digits; this is enforced by the logical volume utilities.

volume name

This is a human-readable identifying name for the logical volume. The logical volume labels on the disks constituting a volume also carry a copy of the volume name, so utilities are able to check that the logical volume on the disks physically present is actually the volume expected by /etc/lvtab.

This field may be null (indicated by a second colon immediately following the one terminating the volume_device_name field). This is legal but deprecated, since in this case, no identity check of the logical volume can be done by the utilities.
options Some numerical options concerning the volume may appear. These are specified in the format "option_name=number". There must be no space between the option_name, the '=' sign, and the numerical value given. Options are separated by colons, as with other fields in an entry.

Currently recognized options are:

stripes=
step=

The stripes option allows a striped logical volume to be created; the value of the parameter specifies the number of ways the volume storage is striped across its constituent devices. If this option is omitted, the logical volume is unstriped.

The step option is meaningful only for striped volumes (and is ignored otherwise); it specifies the granularity with which the storage is to be round-robin distributed over the constituent devices. If this option is omitted, the default is a step of the device tracksize; this is generally a good value so the step option is not normally needed.

device_pathnames
Following any numerical options, there must be a list of the block special file pathnames of the devices constituting the logical volume. This is introduced by the keyword
devs=

The pathnames must be comma-separated.
Each pathname should be the name of the special file for a disk device partition in the /dev/dsk directory. The partition must be one which is legal for use as normal data storage, ie. it must not be one of the dedicated partitions such as the disk volume label, track replacement area etc.

Note that if the volume is striped, some restrictions apply: the number of pathnames must be a multiple of stripes. Further, considering the pathnames as successive groups, each of stripes pathnames, the devices in each group must be all of the same size, and must have the same number of sectors per track.
To obtain best performance from striping, each disk (within every group of 'stripes' disks) should be on a separate controller.
The entries from this file are accessed using the routines in getlvent(3), which returns a structure of the following form:

```c
struct lvtabent {
    char *devname;    /* volume device name */
    char *volname;    /* volume name (human-readable) */
    unsigned stripe;  /* number of ways striped */
    unsigned gran;    /* granularity of striping */
    unsigned ndevs;   /* number of constituent devices */
    int mindex;       /* not currently used */
    char *pathnames[1]; /* pathnames of constituent devices */
};
```

This structure is defined in the <lvtab.h> include file.

**FILES**

/etc/lvtab

**SEE ALSO**

lvinit(1M), mklv(1M), lvck(1M), getlvent(3), lv(7M).
NAME
master – master configuration database

DESCRIPTION
The master configuration database is a collection of files. Each file contains configuration information for a device or module that may be included in the system. A file is named with the module name to which it applies. This collection of files is maintained in a directory called /usr/sysgen/master.d.

Each individual file has an identical format. For convenience, this collection of files will be referred to as the master file, as though it was a single file. This will allow a reference to the master file to be understood to mean the individual file in the master.d directory that corresponds to the name of a device or module. The file is used by the lboot(1M) program to obtain device information to generate the device driver and configurable module files. master consists of two parts; they are separated by a line with a dollar sign ($) in column 1. Part 1 contains device information for both hardware and software devices, and loadable modules. Part 2 contains parameter declarations. Any line with an asterisk (*) in column 1 is treated as a comment.

Part 1, Description
Hardware devices, software drivers and loadable modules are defined with a line containing the following information. Field 1 must begin in the left most position on the line. Fields are separated by white space (tab or blank).

Field 1: element characteristics:

- o specify only once
- r required device
- b block device
- c character device
- t initialize cdevsw[].d_ttys
- j file system
- s software driver
- f STREAMS driver
- m STREAMS module
- x not a driver; a loadable module
- k kernel module
- n driver is fully semaphored for multi-processor operation; the n, p and l directives are ignored on single-processor systems
p driver is not semaphored and should run on only one processor
q driver is not semaphored and should run on network processor

Field 2: handler prefix (14 chars. maximum)
Field 3: software driver external major number; "-" if not a software driver, or to be assigned during execution of lboot(1M)
Field 4: number of sub-devices per device; "-" if none
Field 5: dependency list (optional); this is a comma separated list of other drivers or modules that must be present in the configuration if this module is to be included

For each module, two classes of information are required by lboot(1M): external routine references and variable definitions. Routine lines begin with white space and immediately follow the initial module specification line. These lines are free form, thus they may be continued arbitrarily between non-blank tokens as long as the first character of a line is white space. Variable definition lines begin after a line that contains a 'S' in column one. Variable definitions follow C language conventions, with slight modifications.

Part 1, Routine Reference Lines
If the UNIX system kernel or other dependent module contains external references to a module, but the module is not configured, then these external references would be undefined. Therefore, the routine reference lines are used to provide the information necessary to generate appropriate dummy functions at boot time when the driver is not loaded. Routine references are defined as follows:

Field 1: routine name ()
Field 2: the routine type: one of
{} routine_name(){
{nulldev} routine_name(){nulldev();}
{nosh} routine_name(){return nosys();}
{nodev} routine_name(){return nodev();}
{false} routine_name(){return 0;}
{true} routine_name(){return 1;}

April 1990
{fsnull} routine_name(){return fsnull();}
{fsstray}
    routine_name(){return fsstray();}
{nopkg}
    routine_name(){nopkg();}
{noreach}
    routine_name(){noreach();}

Part 2, Variables
Variables may be declared and (optionally) statically initialized on lines after a line whose first character is a dollar sign ('$'). Variable definitions follow standard C syntax for global declarations, with the following in–line substitutions:

##M the internal major number assigned to the current module if it is a device driver; zero if this module is not a device driver
##E the external major number assigned to the current module; either explicitly defined by the current master file entry, or assigned by lboot(1M)
##C number of controllers present; this number is determined dynamically by lboot(1M) for hardware devices, or by the number provided in the system file for non-hardware drivers or modules
##D number of devices per controller taken directly from the current master file entry

EXAMPLES
A sample master file for a shared memory module would be named "shm". The module is an optional loadable software module that can only be specified once. The module prefix is shm, and it has no major number associated with it. In addition, another module named "ipc" is necessary for the correct operation of this module.

* FLAG PREFIX SOFT #DEV DEPENDENCIES
ox shm -- -- ipc
    shmsys(){nosys}
    shmexec(){}
    shmexit(){}
    shmfork(){}
    shmslp(){true}
    shmtext(){}

$
#define SHMMAX 131072
#define SHMMIN 1
#define SHMMNI 100
#define SHMSEG 6
#define SHMALL 512

struct shmid_ds shmem[SHMMNI];
struct shminfo shminfo = {
    SHMMAX,
    SHMMIN,
    SHMMNI,
    SHMSEG,
    SHMALL,
};

This *master* file will cause routines named *shmsys, shmexec*, etc., to be generated by the boot program if the *shm* driver is not loaded, and there is a reference to this routine from any other module loaded. When the driver is loaded, the structure array *shmem* will be allocated, and the structure *shminfo* will be allocated and initialized as specified.

A sample *master* file for a VME disk driver would be named "dkip". The driver is a block and a character device, the driver prefix is *dkip*, and the external major number is 4. The VME interrupt priority level and vector numbers are declared in the system file */usr/sysgen/system* (see lboot(1M)).

* FLAG PREFIX SOFT #DEV DEPENDENCIES
be dkip 4 – – io

$$$
/* disk driver variable tables */
#include "sys/dvh.h"
#include "sys/dkipreg.h"
#include "sys/elog.h"

struct iotime dkipiotime[##C][DKIPUPC];                           /* io statistics */
struct iobuf dkipetab[##C];                                      /* controller queues */
struct iobuf dkiputab[##C][DKIPUPC];                             /* drive queues */
int dkipmajor = ##E;                                             /* external major # */

This *master* file will cause entries in the block and character device switch tables to be generated, if this module is loaded. Since this is a hardware device (implied by the block and character flags), VME interrupt structures
will be generated, also, by the boot program. The declared arrays will all be sized to the number of controllers present, which is determined by the boot program, based on information in the system file /usr/sysgen/system.

FILES

/usr/sysgen/master.d/*
/usr/sysgen/system

SEE ALSO

system(4), lboot(1M)
NAME
mOTD – message of the day

DESCRIPTION
The file /etc/motd contains information intended to be displayed on the terminal at login time. For sh(1) users, this function is performed by the script /etc/profile. For csh(1) users, the /etc/cshrc displays the contents of the message of the day file.

FILES
/etc/motd

SEE ALSO
csh(1), login(1), sh(1), cshrc(4), profile(4).
NAME
mtab – mounted file system table

DESCRIPTION
Mtab resides in the /etc directory, and contains a table of filesystems currently mounted by the mount command. Umount removes entries from this file.

The file contains a line of information for each mounted filesystem, structurally identical to the contents of /etc/fstab, described in fstab(4). There are a number of lines of the form:

    fsname dir type opts freq passno

For example:

    /dev/root / efs rw 0 0

The file is accessed by programs using getmntent(3), and by the system administrator using a text editor.

FILES
/etc/mtab

SEE ALSO
mount(1M), fstab(4)
getmntent(3) if the NFS option is installed.
NAME

networks – network name database

DESCRIPTION

The /etc/networks file contains information regarding the known networks which comprise the DARPA Internet. For each network a single line should be present with the following information:

- official network name
- network number
- aliases

Items are separated by any number of blanks and/or tab characters. A "#" indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines which search the file. This file is normally created from the official network data base maintained at the Network Information Control Center (NIC), though local changes may be required to bring it up to date regarding unofficial aliases and/or unknown networks.

Network number may be specified in the conventional "." notation using the inet_network() routine from the Internet address manipulation library, inet(3N). Network names may contain any printable character other than a field delimiter, newline, or comment character.

YELLOW PAGES

If the NFS option is installed and Yellow Pages is running, the getnetent(3N) library routines do not access this file.

FILES

/etc/networks

SEE ALSO

gnetent(3N)

BUGS

A name server should be used instead of a static file. A binary indexed file format should be available for fast access.
NAME

passwd – password file

DESCRIPTION

The `/etc/passwd` file contains the following information for each user:

name User’s login name — contains no upper case characters and must not be greater than eight characters long.

password Encrypted password and optional password aging information

numerical user ID
This is the user’s ID in the system and it must be unique.

numerical group ID
This is the number of the group that the user belongs to.

user’s real name
In some versions of UNIX, this field also contains the user’s office, extension, home phone, and so on. For historical reasons this field is called the GECOS field.

initial working directory
The directory that the user is positioned in when they log in — this is known as the ‘home’ directory.

shell program to use as Shell when the user logs in.

An entry beginning with # is ignored. The user’s real name field may contain ‘&’, meaning insert the login name.

The password file is an ASCII file. Each field within each user’s entry is separated from the next by a colon. Each user is separated from the next by a new-line. If the password field is null, no password is demanded; if the Shell field is null, `/bin/sh` is used.

Password aging is effected for a particular user if his encrypted password is followed by a comma and a non-null string of characters from a 64-character alphabet (.,0-9, A-Z, a-z). The first character of the age, \( M \) say, denotes the maximum number of weeks for which a password is valid. A user who attempts to login after his password has expired will be forced to change his password. The next character, \( m \) say, denotes the minimum period in weeks which must expire before the password may be changed. If the second character is omitted, zero week is the default minimum. \( M \) and \( m \) have numerical values in the range 0-63 that correspond to the 64-character alphabet shown above (i.e., / = 1 week; \( z = 63 \) weeks). If \( m = M = 0 \) (derived from the string . or ..) the user will be forced to change his password the next time he logs in (and the "age" will disappear from his entry in the password file). If \( m > M \) (signified, e.g., by the string \( J \)) only the super-user will be able to change the password.
The password file resides in the /etc directory. Because of the encrypted passwords, it has general read permission and can be used, for example, to map numerical user ID's to names.

The passmgmt command can be used to update information in the /etc/passwd file. Appropriate precautions must be taken to lock the /etc/passwd file against simultaneous changes if it is to be edited with a text editor.

YELLOW PAGES

If the NFS option is installed, the passwd file can also have a line beginning with a plus (+), which means to incorporate entries from the Yellow Pages. There are three styles of + entries: all by itself, + means to insert the entire contents of the Yellow Pages password file at that point; +name means to insert the entry (if any) for name from the Yellow Pages at that point; +@name means to insert the entries for all members of the network group name at that point. If a + entry has a non-null password, directory, GECOS, or shell field, they will override what is contained in the Yellow Pages. The numerical user ID and group ID fields cannot be overridden.

Here is a sample /etc/passwd file;

```
root:q,mJzTnu8icF:0:10:superuser:;/bin/csh
bill:6k/7KCFRPNVXg,z:508:10:Bill The Cat:/usr2/bill:/bin/csh
+john:
+@documentation:no-login:
+:..:Guest
```

In this example, there are specific entries for users root and bill, in case the Yellow Pages are not functioning. The user bill will have 63 weeks of maximum password aging and 1 week of minimum password aging. The user john will have his password entry in the Yellow Pages incorporated without change; anyone in the netgroup documentation will have their password field disabled, and anyone else will be able to log in with their usual password, shell, and home directory, but with a GECOS field of Guest.

FILES

/etc/passwd

SEE ALSO

getpwnent(3), passmgmt(1M), login(1), crypt(3), a64l(3C), passwd(1), group(4), netgroup(4)
NAME
pnch – file format for card images

DESCRIPTION

The PNCH format is a convenient representation for files consisting of card images in an arbitrary code.

A PNCH file is a simple concatenation of card records. A card record consists of a single control byte followed by a variable number of data bytes. The control byte specifies the number (which must lie in the range 0-80) of data bytes that follow. The data bytes are 8-bit codes that constitute the card image. If there are fewer than 80 data bytes, it is understood that the remainder of the card image consists of trailing blanks.
NAME
printcap – printer capability data base

SYNOPSIS
/etc/printcap

DESCRIPTION
Printcap is a simplified version of the termcap(4) data base used to describe line printers. The spooling system accesses the printcap file every time it is used, allowing dynamic addition and deletion of printers. Each entry in the data base is used to describe one printer. This data base may not be substituted for, as is possible for termcap, because it may allow accounting to be bypassed.

The default printer is normally lp, though the environment variable PRINTER may be used to override this. Each spooling utility supports an option, –Pprinter, to allow explicit naming of a destination printer.

Refer to the 4.3BSD Line Printer Spooler Manual for a complete discussion on how setup the database for a given printer.

CAPABILITIES
Refer to termcap(4) for a description of the file layout.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>af</td>
<td>str</td>
<td>NULL</td>
<td>name of accounting file</td>
</tr>
<tr>
<td>br</td>
<td>num</td>
<td>none</td>
<td>if lp is a tty, set the baud rate (ioctl call)</td>
</tr>
<tr>
<td>cf</td>
<td>str</td>
<td>NULL</td>
<td>cifplot data filter</td>
</tr>
<tr>
<td>df</td>
<td>str</td>
<td>NULL</td>
<td>tex data filter (DVI format)</td>
</tr>
<tr>
<td>fc</td>
<td>num</td>
<td>0</td>
<td>if lp is a tty, clear flag bits (see compatibility notes)</td>
</tr>
<tr>
<td>ff</td>
<td>str</td>
<td>&quot;&quot;f&quot;</td>
<td>string to send for a form feed</td>
</tr>
<tr>
<td>fo</td>
<td>bool</td>
<td>false</td>
<td>print a form feed when device is opened</td>
</tr>
<tr>
<td>fs</td>
<td>num</td>
<td>0</td>
<td>like ‘fc’ but set bits (see compatibility notes)</td>
</tr>
<tr>
<td>gf</td>
<td>str</td>
<td>NULL</td>
<td>graph data filter (plot (3X) format)</td>
</tr>
<tr>
<td>hl</td>
<td>bool</td>
<td>false</td>
<td>print the burst header page last</td>
</tr>
<tr>
<td>ic</td>
<td>bool</td>
<td>false</td>
<td>driver supports (non standard) ioctl to indent printout</td>
</tr>
<tr>
<td>if</td>
<td>str</td>
<td>NULL</td>
<td>name of text filter which does accounting</td>
</tr>
<tr>
<td>lf</td>
<td>str</td>
<td>&quot;/dev/console&quot;</td>
<td>error logging file name</td>
</tr>
<tr>
<td>lo</td>
<td>str</td>
<td>&quot;lock&quot;</td>
<td>name of lock file</td>
</tr>
<tr>
<td>lp</td>
<td>str</td>
<td>&quot;/dev/lp&quot;</td>
<td>device name to open for output</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>mx</td>
<td>num</td>
<td>1000</td>
<td>maximum file size in BUFSIZ blocks, zero = unlimited</td>
</tr>
<tr>
<td>nd</td>
<td>str</td>
<td>NULL</td>
<td>next directory for list of queues (unimplemented)</td>
</tr>
<tr>
<td>nf</td>
<td>str</td>
<td>NULL</td>
<td>ditroff data filter (device independent troff)</td>
</tr>
<tr>
<td>of</td>
<td>str</td>
<td>NULL</td>
<td>name of output filtering program</td>
</tr>
<tr>
<td>pc</td>
<td>num</td>
<td>200</td>
<td>price per foot or page in hundredths of cents</td>
</tr>
<tr>
<td>pl</td>
<td>num</td>
<td>66</td>
<td>page length (in lines)</td>
</tr>
<tr>
<td>pw</td>
<td>num</td>
<td>132</td>
<td>page width (in characters)</td>
</tr>
<tr>
<td>px</td>
<td>num</td>
<td>0</td>
<td>page width in pixels (horizontal)</td>
</tr>
<tr>
<td>py</td>
<td>num</td>
<td>0</td>
<td>page length in pixels (vertical)</td>
</tr>
<tr>
<td>rf</td>
<td>str</td>
<td>NULL</td>
<td>filter for printing FORTRAN style text files</td>
</tr>
<tr>
<td>rg</td>
<td>str</td>
<td>NULL</td>
<td>restricted group. Only members of group allowed access</td>
</tr>
<tr>
<td>rm</td>
<td>str</td>
<td>NULL</td>
<td>machine name for remote printer</td>
</tr>
<tr>
<td>rp</td>
<td>str</td>
<td>&quot;lp&quot;</td>
<td>remote printer name argument</td>
</tr>
<tr>
<td>rs</td>
<td>bool</td>
<td>false</td>
<td>restrict remote users to those with local accounts</td>
</tr>
<tr>
<td>rw</td>
<td>bool</td>
<td>false</td>
<td>open the printer device for reading and writing</td>
</tr>
<tr>
<td>sb</td>
<td>bool</td>
<td>false</td>
<td>short banner (one line only)</td>
</tr>
<tr>
<td>sc</td>
<td>bool</td>
<td>false</td>
<td>suppress multiple copies</td>
</tr>
<tr>
<td>sd</td>
<td>str</td>
<td>&quot;/usr/spool/lpd&quot;</td>
<td>spool directory</td>
</tr>
<tr>
<td>sf</td>
<td>bool</td>
<td>false</td>
<td>suppress form feeds</td>
</tr>
<tr>
<td>sh</td>
<td>bool</td>
<td>false</td>
<td>suppress printing of burst page header</td>
</tr>
<tr>
<td>st</td>
<td>str</td>
<td>&quot;status&quot;</td>
<td>status file name</td>
</tr>
<tr>
<td>tf</td>
<td>str</td>
<td>NULL</td>
<td>troff data filter (cat phototypesetter)</td>
</tr>
<tr>
<td>tr</td>
<td>str</td>
<td>NULL</td>
<td>trailer string to print when queue empties</td>
</tr>
<tr>
<td>vf</td>
<td>str</td>
<td>NULL</td>
<td>raster image filter</td>
</tr>
<tr>
<td>xc</td>
<td>num</td>
<td>0</td>
<td>if lp is a tty, clear local mode bits (tty (4))</td>
</tr>
<tr>
<td>xs</td>
<td>num</td>
<td>0</td>
<td>like 'xc' but set bits</td>
</tr>
</tbody>
</table>

If the local line printer driver supports indentation, the daemon must understand how to invoke it.

**FILTERS**

The *lpd*(1M) daemon creates a pipeline of *filters* to process files for various printer types. The filters selected depend on the flags passed to *lpr*(1). The pipeline set up is:

```
-p  pr | if  regular text + pr(1)
none if  regular text
-c  cf  cfilter
-d  df  DVI (tex)
```

April 1990

Version 4.0
The **if** filter is invoked with arguments:

```
if [-c ] -wwidth -llength -indent -n login -h host acct-file
```

The `-c` flag is passed only if the `-l` flag (pass control characters literally) is specified to `lpr`. `Width` and `length` specify the page width and length (from `pw` and `pl` respectively) in characters. The `-n` and `-h` parameters specify the login name and host name of the owner of the job respectively. `Acct-file` is passed from the `af printcap` entry.

If no `if` is specified, `of` is used instead, with the distinction that `of` is opened only once, while `if` is opened for every individual job. Thus, `if` is better suited to performing accounting. The `of` is only given the `width` and `length` flags.

All other filters are called as:

```
filter -xwidth -ylength -n login -h host acct-file
```

where `width` and `length` are represented in pixels, specified by the `px` and `py` entries respectively.

All filters take `stdin` as the file, `stdout` as the printer, may log either to `stderr` or using `syslog(3)`, and must not ignore SIGINT.

**LOGGING**

Error messages generated by the line printer programs themselves (that is, the `lp*` programs) are logged by `syslog(3)` using the LPR facility. Messages printed on `stderr` of one of the filters are sent to the corresponding `If` file. The filters may, of course, use `syslog` themselves.

Error messages sent to the console have a carriage return and a line feed appended to them, rather than just a line feed.

**COMPATIBILITY NOTES**

In an attempt to provide compatibility with existing BSD printcap entries, the SGI version of the `lpd` spooler emulates the output bits in the BSD tty flag word (defined in the BSD include file `<sgtty.h>`) via IRIX termio.

**SEE ALSO**

`termcap(4), lpc(1M), lpd(1M), pac(1M), lpr(1), lpq(1), lprm(1)`

April 1990

- 3 -

Version 4.0
NAME
profile – setting up an environment at login time

SYNOPSIS
/etc/profile
$HOME/.profile

DESCRIPTION
All users who have the shell, sh(1), as their login command have the commands in these files executed as part of their login sequence.

/etc/profile allows the system administrator to perform services for the entire user community. Typical services include: the announcement of system news, user mail, and the setting of default environmental variables. It is not unusual for /etc/profile to execute special actions for the root login or the su(1) command.

The file $HOME/.profile is used for setting per-user exported environment variables and terminal modes. The following example is typical (except for the comments):

```bash
# Set the file creation mask to prohibit
# others from reading my files.
umask 027
# Add my own /bin directory to the shell search sequence.
PATH=$PATH:$HOME/bin
# Set terminal type
eval \'tset -S -Q\'
# Set the interrupt character to control-c.
stty intr "c"
# List directories in columns if standard out is a terminal.
ls() { if [-t ]; then /bin/ls -C $*; else /bin/ls $*; fi }
```

FILES
/etc/TIMEZONE timezone environment
$HOME/.profile user-specific environment
/etc/profile system-wide environment

SEE ALSO
terminfo(4), timezone(4), environ(5), term(5).

evn(1), login(1), mail(1), sh(1), stty(1), tset(1), tput(1) in the User’s Reference Manual.
User’s Guide.
Chapter 9 in the Programmer’s Guide.
NOTES

Care must be taken in providing system-wide services in /etc/profile. Personal .profile files are better for serving all but the most global needs.
NAME
  proto – prototype job file for at

SYNOPSIS
  /usr/lib/cron/.proto
  /usr/lib/cron/.proto.queue

DESCRIPTION
  When a job is submitted to at(1) or batch(1), the job is constructed as a
  shell script. First, a prologue is constructed, consisting of:
  • A header whether the job is an at job or a batch job (actually, at jobs
    submitted to all queues other than queue a, not just to the batch queue b,
    are listed as batch jobs); the header will be
      : at job
    for an at job, and
      : batch job
    for a batch job.
  • A set of Bourne shell commands to make the environment (see
    environ(5)) for the at job the same as the current environment;
  • A command to run the user’s shell (as specified by the SHELL
    environment variable) with the rest of the job file as input.

At then reads a "prototype file," and constructs the rest of the job file from
it.

Text from the prototype file is copied to the job file, except for special
"variables" that are replaced by other text:

$d  is replaced by the current working directory
$l  is replaced by the current file size limit (see ulimit(2))
$m  is replaced by the current umask (see umask(2))
$t  is replaced by the time at which the job should be run,
    expressed as seconds since January 1, 1970, 00:00
    Greenwich Mean Time, preceded by a colon
$<  is replaced by text read by at from the standard input (that
    is, the commands provided to at to be run in the job)

If the job is submitted in queue queue, at uses the file
/usr/lib/cron/.proto.queue as the prototype file if it exists, otherwise it will
use the file /usr/lib/cron/.proto.

EXAMPLES
  The standard .proto file supplied is:
#ident "@(#)adm:proto 1.2"
cd $d
ulimit $l
umask $m
$<

which causes commands to change the current directory in the job to the current directory at the time `at` was run, to change the file size limit in the job to the file size limit at the time `at` was run, and to change the umask in the job to the umask at the time `at` was run, to be inserted before the commands in the job.

FILES
/usr/lib/cron/.proto
/usr/lib/cron/.proto.queue

SEE ALSO
`at(1)`
NAME
protocols – protocol name data base

DESCRIPTION
The /etc/protocols file contains information regarding the known protocols used in the DARPA Internet. For each protocol a single line should be present with the following information:

  official protocol name
  protocol number
  aliases

Items are separated by any number of blanks and/or tab characters. A "#" indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines which search the file.

Protocol names may contain any printable character other than a field delimiter, newline, or comment character.

YELLOW PAGES
If the NFS option is installed and Yellow Pages is running, the getprotoent(3N) library routines do not access this file.

FILES
/etc/protocols

SEE ALSO
getprotoent(3N)

BUGS
A name server should be used instead of a static file.
NAME
queuedefs – at/batch/cron queue description file

SYNOPSIS
/usr/lib/cron/queuedefs

DESCRIPTION
The queuedefs file describes the characteristics of the queues managed by cron(1M). Each non-comment line in this file describes one queue. The format of the lines are as follows:

q.[njob][nicen][nwaitw]

The fields in this line are:

q    The name of the queue. a is the default queue for jobs started by at(1); b is the default queue for jobs started by batch(1); c is the default queue for jobs run from a cron-tab file.

njob  The maximum number of jobs that can be run simultaneously in that queue; if more than njob jobs are ready to run, only the first njob jobs will be run, and the others will be run as jobs that are currently running terminate. The default value is 100.

nice  The nice(1) value to give to all jobs in that queue that are not run with a user ID of super-user. The default value is 2.

nwait The number of seconds to wait before rescheduling a job that was deferred because more than njob jobs were running in that job’s queue, or because more than 25 jobs were running in all the queues. The default value is 60.

Lines beginning with # are comments, and are ignored.

EXAMPLE

a.4j1n
b.2j2n90w

This file specifies that the a queue, for at jobs, can have up to 4 jobs running simultaneously; those jobs will be run with a nice value of 1. As no nwait value was given, if a job cannot be run because too many other jobs are running cron will wait 60 seconds before trying again to run it. The b queue, for batch jobs, can have up to 2 jobs running simultaneously; those jobs will be run with a nice value of 2. If a job cannot be run because too many other jobs are running, cron will wait 90 seconds before trying again to run it. All other queues can have up to 100 jobs running simultaneously; they will be run with a nice value of 2, and if a job cannot be run because
too many other jobs are running cron will wait 60 seconds before trying again to run it.

FILES
/usr/lib/cron/queuedefs

SEE ALSO
cron(1M)
NAME
rcsfile – format of RCS file

DESCRIPTION
An RCS file is an ASCII file. Its contents are described by the grammar below. The text is free format, i.e., spaces, tabs and new lines have no significance except in strings. Strings are enclosed by ‘@’. If a string contains a ‘@’, it must be doubled.

The meta syntax uses the following conventions: ‘|’ (bar) separates alternatives; ‘{’ and ‘}’ enclose optional phrases; ‘{’ and ‘}’* enclose phrases that may be repeated zero or more times; ‘{’ and ‘}+’ enclose phrases that must appear at least once and may be repeated; ‘<’ and ‘>’ enclose nonterminals.

<rcstext> ::= <admin> {<delta>}* <desc> {<deltatext>}*

<admin> ::= head [<num>];
access [<id>]*;
symbols [<id> : <num>]*;
locks [<id> : <num>]*;
comment [<string>];

<delta> ::= <num>
date <num>;
author <id>;
state [<id>];
branches [<num>]*;
next [<num>];

<desc> ::= desc <string>

<deltatext> ::= <num>
log <string>
text <string>

<num> ::= {<digit>{.}}+

digit ::= 0|1|2|3|4|5|6|7|8|9

{id} ::= <letter>{<idchar>}

<letter> ::= A|B|...|Z|a|b|...|z

<idchar> ::= Any printing ASCII character except space, tab, carriage return, new line, and <special>.
<special> ::= ;l:1,1@  
<string> ::= @ {any ASCII character, with '@' doubled}@  

Identifiers are case sensitive. Keywords are in lower case only. The sets of keywords and identifiers may overlap.

The <delta> nodes form a tree. All nodes whose numbers consist of a single pair (e.g., 2.3, 2.1, 1.3, etc.) are on the "trunk", and are linked through the "next" field in order of decreasing numbers. The "head" field in the <admin> node points to the head of that sequence (i.e., contains the highest pair).

All <delta> nodes whose numbers consist of 2n fields (n\geq2) (e.g., 3.1.1.1, 2.1.2.2, etc.) are linked as follows. All nodes whose first (2n)-1 number fields are identical are linked through the "next" field in order of increasing numbers. For each such sequence, the <delta> node whose number is identical to the first 2(n-1) number fields of the deltas on that sequence is called the branchpoint. The "branches" field of a node contains a list of the numbers of the first nodes of all sequences for which it is a branchpoint. This list is ordered in increasing numbers.
Example:

Fig. 1: A revision tree

IDENTIFICATION
Author: Walter F. Tichy, Purdue University, West Lafayette, IN, 47907.
Revision Number: 1.5; Release Date: 89/09/12.
Copyright © 1982 by Walter F. Tichy.
SEE ALSO

ci(1), co(1), ident(1), rcs(1), rcsdiff(1), rcsintro(1), rcsmerge(1), rlog(1).
NAME
reloc – relocation information for a common object file

SYNOPSIS
#include <reloc.h>

DESCRIPTION
Object files have one relocation entry for each relocatable reference in the
text or data. If relocation information is present, it will be in the following
format.

    struct reloc
    {
        long r_vaddr;        /* (virtual) address of
                              reference */
        unsigned r_symndx:24, /* index into symbol
                              table */
                              r_reserved:3,
        r_type:4,
                              rExtern:1; /* relocation type */
                          /* if 1 symndx is an index
                           into the extern symb tab
                           else symndx is a
                           section # */
    
    
    
    
    /* Relocation types */
#define R_ABS     0
#define R_REFHALF 1
#define R_REFWORD 2
#define R_JMPADDR 3
#define R_REFHI   4
#define R_REFLO   5
#define R_GPREL   6
#define R_LITERAL 7

    /* Section numbers */
#define R_SN_NULL 0
#define R_SN_TEXT 1
#define R_SN_RDATA 2
#define R_SN_DATA  3
#define R_SN_SDATA 4
#define R_SN_SBSS  5
#define R_SN_BSS   6
#define R_SN_INIT  7
#define R_SN_LIT8  8
#define R_SN_LIT4 9

The link editor reads each input section and performs relocation. The relocation entries direct how references found within the input section are treated.

If $r_\text{extern}$ is zero, then the reference is a local relocation entry and the $r_\text{symndex}$ is a section number ($R_\text{SN}_*$). For these entries, the starting address for the section referenced by the section number is used in place of an external symbol table entry’s value. The assembler and loader always use local relocation entries if the item to be relocated is defined in the object file.

For every external relocation (except $R_\text{ABS}$), a signed constant is added to the symbols virtual address that the relocation entry refers to. This constant is assembled at the address being relocated.

$R_\text{ABS}$ The reference is absolute and no relocation is necessary. The entry will be ignored.

$R_\text{REFHALF}$ A 16-bit reference to the symbol’s virtual address.

$R_\text{REFWORD}$ A 32-bit reference to the symbol’s virtual address.

$R_\text{JMPADDR}$ A 26-bit jump instruction reference to the symbol’s virtual address.

$R_\text{REFHI}$ A reference to the high 16 bits of the symbol’s 16 bits of the symbol’s virtual address. The next relocation entry must be the corresponding $R_\text{REFLO}$ entry so the proper value of the constant to be added to the symbol’s virtual address can be reconstructed.

$R_\text{REFLO}$ A reference to the low 16-bits of the symbol’s virtual address.

$R_\text{GPREL}$ A 16-bit offset to the symbol’s virtual address from the global pointer register.

$R_\text{LITERAL}$ A 16-bit offset to the literal’s virtual address from the global pointer register.

Relocation entries are generated automatically by the assembler and automatically used by the link editor. Link editor options exist for both preserving and removing the relocation entries from object files.

The number of relocation entries for a section is found in the $s_\text{nreloc}$ field of the section header. This field is a C language short and can overflow with large objects. If this field overflows, the section header $s_\text{flags}$ field has the $S_\text{NRELOC_OVFL}$ bit set. In this case, the true number of
relocation entries is found in the $r_{vaddr}$ field of the first relocation entry for that section. That relocation entry has a type of $R\_ABS$ so it is ignored when the relocation takes place.

SEE ALSO

*Assembly Language Programmer's Guide*, chapter Object File Format, section Section Relocation Information.
as(1), ld(1), a.out(4), scnhdr(4), syms(4).
NAME
esolver – host-address resolver configuration file

SYNOPSIS

/usr/etc/resolv.conf

DESCRIPTION

This file controls the behavior of gethostbyname(3N), gethostbyaddr(3N) and the resolver(3N) routines in the C library. It is read by these routines the first time they are invoked by a process.

The file is designed to be human readable and contains a list of keywords with values that provide various types of resolver information. The keyword and value must appear on a single line, and the keyword (e.g., nameserver) must start the line. The value follows the keyword, separated by white space.

This file is not necessary if there is a name server running on the local machine and the host name contains the domain name. It is necessary, however, if the system administrator wants to override the default ordering of the host lookup services.

The following configuration option applies to gethostbyname(3N) and gethostbyaddr(3N):

hostresorder

A list specifying the ordering of host lookup services used by gethostbyname(3N) and gethostbyaddr(3N). The recognized services and their keywords are Yellow Pages ("yp"), BIND ("bind") and /etc/hosts ("local"). The keywords are separated by white space or a slash (/). Normally, if a service cannot find the answer or is not running, the next service in the list is queried. The slash separator indicates the previous service in the list is authoritative: even if it cannot find the answer, the search is terminated. For example,

    hostresorder  bind  local

specifies that BIND is checked first (bypassing YP) and if no answer is found, the file /etc/hosts is then checked. At least one service keyword must be listed. The default is "yp / bind / local", i.e., YP and BIND have authoritative information if they are available. A user may override the ordering specified by hostresorder with the environment variable HOSTRESORDER set to a string containing the service keywords.
The following options are used by the Internet Domain Name System resolver routines only:

**nameserver**

Internet address (in dot notation) of a name server that the resolver should query. Up to 3 of these lines may be specified; the resolver library queries them in the order listed. If no nameserver entries are present, the default is to use the name server on the local machine. (The algorithm used is to try a name server, and if the query times out, try the next, until out of name servers, then repeat trying all the name servers until a maximum number of retries are made). When specifying a nameserver entry for the local machine, use the address 0 instead of the “localhost” address of 127.1.

**domain** Local domain name. Most queries for names within this domain can use short names relative to the local domain. If no domain entry is present, the domain is determined from the local host name returned by gethostname (2); the domain part is taken to be everything after the first ‘.’. Finally, if the host name does not contain a domain part, the root domain is assumed.

**search** Search list for host-name lookup. The search list is normally determined from the local domain name; by default, it begins with the local domain name, then successive parent domains that have at least two components in their names. This may be changed by listing the desired domain search path following the search keyword with spaces or tabs separating the names. Most resolver queries will be attempted using each component of the search path in turn until a match is found. Note that this process may be slow and will generate a lot of network traffic if the servers for the listed domains are not local, and that queries will time out if no server is available for one of the domains.

The search list is currently limited to six domains with a total of 256 characters. The first item in the list becomes the default domain name; the remaining items are the other domains to search after the default one.

The domain and search keywords are mutually exclusive. If more than one instance of these keywords is present, the last instance will override. These keywords are ignored if the environment variable LOCALDOMAIN is set.
NOTE
The **hostresorder** keyword is specific to IRIX.

FILES
/usr/etc/resolv.conf

ENVIRONMENT
HOSTRESORDER overrides **hostresorder**
LOCALDOMAIN overrides **domain** and **search**

SEE ALSO
gethostbyname(3N), resolver(3N), sethostresorder(3N), hostname(5), named(1M)

Network administration chapters in the *Network Communications Guide*
NAME
rhosts – list of trusted hosts and users

DESCRIPTION
Each user may have a .rhosts file in his home directory. This file contains a
list of users on other hosts in the network that are trusted in the following
sense: when making requests to access the user’s system with rcp(1C),
rdist(1C), rlogin(1C), or rsh(1C), they are allowed to assume the user’s
identity without specifying a password. In other words, the remote user has
exactly the same access privileges on the local system that the owner of the
.rhosts file does and this access is granted without any attempt to verify the
remote user’s identity by requiring him to enter a password. The incoming
request includes the user name that should be used on the local system. The
.rhosts file owned by that local user acts as a logical extension to the
hosts.equiv(4) file when deciding whether to grant permission for the
incoming rcp(1C), rdist(1C), rlogin(1C), or rsh(1C) request.

The .rhosts file has the same format as the hosts.equiv(4) file.

NOTES
The owner of the .rhosts file must be the super-user (i.e., root) or the user in
whose home directory it resides. The contents of the file will be disre-
garded if it is owned by another user or if its permissions allow anyone who
is not the owner to modify the file. Use the chmod(1) command to add the
proper protection:

    chmod go-w .rhosts

Special care should be taken in deciding the contents of the file /.rhosts.
Any host or user added to this file has the ability to become the superuser
on the local system without entering the password. Note that /.rhosts is not
required.

FILES
~/.rhosts

SEE ALSO
rcp(1C), rdist(1C), rlogin(1C), rsh(1C), hosts.equiv(4)
NAME
rpc – RPC program number data base

SYNOPSIS
/etc/rpc

DESCRIPTION
The rpc file contains user readable names that can be used in place of Sun
RPC program numbers. Each line has the following information:

- name of server for the RPC program
- RPC program number
- aliases

Items are separated by any number of blanks and/or tab characters. A “#”
indicates the beginning of a comment; characters up to the end of the line
are not interpreted by routines which search the file.

Here is an excerpt of the /etc/rpc file on IRIX:
#
#    rpc  1.10  87/04/10
#
portmapper  100000  portmap sunrpc
rstatd  100001  rstat rup permeter
rusersd  100002  rusers
nfs  100003  nfsprog
ypserv  100004  ypprog
mountd  100005  mount showmount
ypbind  100007
walld  100008  wall shutdown
sgi_toolkitbus  391001
sgi_fam  391002

FILES
/etc/rpc

SEE ALSO
getrpcent(3N)
NAME

sccsfile – format of SCCS file

DESCRIPTION

An SCCS (Source Code Control System) file is an ASCII file. It consists of six logical parts: the checksum, the delta table (contains information about each delta), user names (contains login names and/or numerical group IDs of users who may add deltas), flags (contains definitions of internal keywords), comments (contains arbitrary descriptive information about the file), and the body (contains the actual text lines intermixed with control lines).

Throughout an SCCS file there are lines which begin with the ASCII SOH (start of heading) character (octal 001). This character is hereafter referred to as the control character and will be represented graphically as @. Any line described below which is not depicted as beginning with the control character is prevented from beginning with the control character.

Entries of the form DDDDDD represent a five-digit string (a number between 00000 and 99999).

Each logical part of an SCCS file is described in detail below.

Checksum

The checksum is the first line of an SCCS file. The form of the line is:

@hDDDDD

The value of the checksum is the sum of all characters, except those of the first line. The @h provides a magic number of (octal) 064001.

Delta table

The delta table consists of a variable number of entries of the form:

@s DDDDDD/DDDDD/DDDDD
@d <type> <SCCS ID> yr/mo/da hr:mi:se <pgmr> DDDDD DDDDD
@i DDDDD ...
@x DDDDD ...
@g DDDDD ...
@m <MR number>
.
.
.
@e <comments> ...
.
The first line (@s) contains the number of lines inserted/deleted/unchanged, respectively. The second line (@d) contains the type of the delta (currently, normal: D, and removed: R), the SCCS ID of the delta, the date and time of creation of the delta, the login name corresponding to the real user ID at the time the delta was created, and the serial numbers of the delta and its predecessor, respectively.

The @i, @x, and @g lines contain the serial numbers of deltas included, excluded, and ignored, respectively. These lines are optional.

The @m lines (optional) each contain one MR number associated with the delta; the @c lines contain comments associated with the delta.

The @e line ends the delta table entry.

**User names**

The list of login names and/or numerical group IDs of users who may add deltas to the file, separated by new-lines. The lines containing these login names and/or numerical group IDs are surrounded by the bracketing lines @u and @U. An empty list allows anyone to make a delta. Any line starting with a! prohibits the succeeding group or user from making deltas.

**Flags**

Keywords used internally. [See admin(1) for more information on their use.] Each flag line takes the form:

```
@f <flag> <optional text>
```

The following flags are defined:

- @ft <type of program>
- @fv <program name>
- @fi <keyword string>
- @fb
- @fm <module name>
- @ff <floor>
- @fc <ceiling>
@f\ d <default-sid>
@f\ n
@f\ j
@f\ l <lock-releases>
@f\ q <user defined>
@f\ z <reserved for use in interfaces>

The t flag defines the replacement for the %Y% identification keyword. The v flag controls prompting for MR numbers in addition to comments; if the optional text is present it defines an MR number validity checking program. The i flag controls the warning/error aspect of the “No id keywords” message. When the i flag is not present, this message is only a warning; when the i flag is present, this message will cause a “fatal” error (the file will not be gotten, or the delta will not be made). When the b flag is present the –b keyletter may be used on the get command to cause a branch in the delta tree. The m flag defines the first choice for the replacement text of the %M% identification keyword. The f flag defines the “floor” release; the release below which no deltas may be added. The c flag defines the “ceiling” release; the release above which no deltas may be added. The d flag defines the default SID to be used when none is specified on a get command. The n flag causes delta to insert a “null” delta (a delta that applies no changes) in those releases that are skipped when a delta is made in a new release (e.g., when delta 5.1 is made after delta 2.7, releases 3 and 4 are skipped). The absence of the n flag causes skipped releases to be completely empty. The j flag causes get to allow concurrent edits of the same base SID. The I flag defines a list of releases that are locked against editing (get(1) with the –e keyletter). The q flag defines the replacement for the %Q% identification keyword. The z flag is used in certain specialized interface programs. Comments Arbitrary text is surrounded by the bracketing lines @t and @T. The comments section typically will contain a description of the file’s purpose.

Body

The body consists of text lines and control lines. Text lines do not begin with the control character, control lines do. There are three kinds of control lines: insert, delete, and end, represented by:

@I DDDDDD
@D DDDDDD
@E DDDDDD
respectively. The digit string is the serial number corresponding to the delta for the control line.

SEE ALSO
admin(1), delta(1), get(1), prs(1).
NAME
scnhdr — section header for a MIPS object file

SYNOPSIS
#include <scnhdr.h>

DESCRIPTION
Every MIPS object file has a table of section headers to specify the layout
of the data within the file. Each section within an object file has its own
header. The C structure appears below:

struct scnhdr
{
    char s_name[8]; /* section name */
    long s_paddr;  /* physical address, aliased s_nlib */
    long s_vaddr;  /* virtual address */
    long s_size;   /* section size */
    long s_scnptr; /* file ptr to raw data for section */
    long s_relptr; /* file ptr to relocation */
    long s_innopr; /* file ptr to gp table */
    unsigned short s_nreloc; /* number of relocation entries */
    unsigned short s_nlnno; /* number of gp table entries */
    long s_flags;   /* flags */
};

File pointers are byte offsets into the file; they can be used as the offset in a
call to fseek [see ldfunc(4)]. If a section is initialized, the file contains the
actual bytes. An uninitialized section is somewhat different. It has a size,
symbols defined in it, and symbols that refer to it. But it can have no relo-
cation entries or data. Consequently, an uninitialized section has no raw
data in the object file, and the values for s_scnptr, s_relptr, and s_nreloc
are zero.

The entries that refer to line numbers (s_innopr and s_nlnno) are not used for
line numbers on MIPS machines. See the header file <sym.h> [line-
num(4)] for the entries to get to the line number table. The entries that were
for line numbers in the section header are used for gp tables on MIPS
machines.

The number of relocation entries for a section is found in the s_nreloc field
of the section header. This field is a C language short and can overflow
with large objects. If this field overflows, the section header s_flags field
has the S_NRELOC_OVFL bit set. In this case, the true number of relo-
cation entries is found in the r_vaddr field of the first relocation entry for that
section. That relocation entry has a type of R_ABS so it is ignored when
the relocation takes place.

April 1990
- 1 -
Version 5.0
The gp table gives the section size corresponding to each applicable value of the compiler option -G num (always including 0), sorted by smallest size first. It is pointed to by the s_innopt field in the section header and its number of entries (including the header) is in the s_ninno field in the section header. This table only needs to exist for the .sdata and .sbss sections. If there is no "small" section then the gp table for it is attached to the corresponding "large" section so the information still gets to the link editor, ld(1). The C union for the gp table appears below.

```c
union gp_table
{
  struct {
    long current_g_value;        /* actual value */
    long unused;
  } header;
  struct {
    long g_value;                /* hypothetical value */
    long bytes;                 /* section size corresponding */
    /* to hypothetical value */
  } entry;
};
```

Each gp table has one header structure that contains the actual value of the -G num option used to produce the object file. An entry must exist for every applicable value of the -G num option. The applicable values are all the sizes of the data items in that section.

For .lib sections, the number of shared libraries is in the s_nlib field (an alias to s_paddr). The .lib section is made up of s_nlib descriptions of shared libraries. Each description of a shared library is a libscn structure followed by the path name to the shared library. The C structure appears below and is defined in <scnhdr.h>.

```c
struct libscn
{
  long size;                  /* size of this entry (including target name) */
  long offset;                /* offset from start of entry to target name */
  long tsize;                 /* text size in bytes, padded to DW boundary */
  long dsize;                 /* data size in bytes, padded to DW boundary */
  long bsize;                 /* bss size in bytes, padded to DW boundary */
  long text_start;            /* base of text used for this library */
  long data_start;            /* base of data used for this library */
  long bss_start;             /* base of bss used for this library */
  /* pathname of target shared library */
};
```
SEE ALSO
   ld(1), fseek(3S), a.out(4), linenum(4), reloc(4).

BUGS
   The s_nrelloc field has been known to overflow on fully linked objects when
   the relocation entries are saved.
NAME

scr_dump – format of curses screen image file.

SYNOPSIS

scr_dump(file)

DESCRIPTION

The curses(3X) function scr_dump() will copy the contents of the screen into a file.  The format of the screen image is as described below.

The name of the tty is 20 characters long and the modification time (the mtime of the tty that this is an image of) is of the type time_t.  All other numbers and characters are stored as ctype (see <curses.h>).  No newlines are stored between fields.

<magic number: octal 0433>
<name of tty>
<mod time of tty>
<columns> <lines>
<line length> <chars in line> for each line on the screen
<line length> <chars in line>

<labels?>
<cursor row> <cursor column>

Only as many characters as are in a line will be listed.  For example, if the <line length> is 0, there will be no characters following <line length>.  If <labels?> is TRUE, following it will be

<number of labels>
<label width>
<chars in label 1>
<chars in label 2>

SEE ALSO

curses(3X).
NAME
services – service name data base

DESCRIPTION
The /etc/services file contains information regarding the known services available in the DARPA Internet. For each service a single line should be present with the following information:

- official service name
- port number
- protocol name
- aliases

Items are separated by any number of blanks and/or tab characters. The port number and protocol name are considered a single item; a "/'" is used to separate the port and protocol (e.g. "'512/tcp'"). A "'#" indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines which search the file.

Service names may contain any printable character other than a field delimiter, newline, or comment character.

YELLOW PAGES
If the NFS option is installed and Yellow Pages is running, the getservent(3N) library routines do not access this file.

FILES
/etc/services

SEE ALSO
getservent(3N)

BUGS
A name server should be used instead of a static file.
NAME

syms – MIPS symbol table

SYNOPSIS

#include <sym.h>
#include <symconst.h>

DESCRIPTION

The MIPS symbol table departs from the standard COFF symbol table. The symbol table consists of many tables unbundling information usually found in the one COFF symbol table. The symbol table should be viewed as a hand-crafted, network-style database designed for space and access efficiency.

The following structures or tables appear in the MIPS symbol table:

<table>
<thead>
<tr>
<th>TABLE</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>symbolic header</td>
<td>sizes and locations of all other tables.</td>
</tr>
<tr>
<td>file descriptors</td>
<td>per file locations for other tables.</td>
</tr>
<tr>
<td>procedure descriptors</td>
<td>frame info and location of procedure info.</td>
</tr>
<tr>
<td>local symbols</td>
<td>local type, local variable, and scoping info.</td>
</tr>
<tr>
<td>local strings</td>
<td>string space for local symbols.</td>
</tr>
<tr>
<td>line numbers</td>
<td>compacted by encoding, contains a line per instruction.</td>
</tr>
<tr>
<td>relative file desc.</td>
<td>indirection for inter-file symbol access.</td>
</tr>
<tr>
<td>optimization symbols</td>
<td>to be defined.</td>
</tr>
<tr>
<td>auxiliary symbols</td>
<td>variable data type info for each local symbol.</td>
</tr>
<tr>
<td>external symbols</td>
<td>loader symbols (global text and data).</td>
</tr>
<tr>
<td>external strings</td>
<td>string space for external symbols.</td>
</tr>
<tr>
<td>dense numbers</td>
<td>(file, symbol) index pairs for compiler use.</td>
</tr>
</tbody>
</table>

External and local symbols contain the standard concept of a "symbol" as follows:

```c
struct {
    long iss;  /* index into string space */
    long value; /* addr, size, etc., depends on sc & st */
    unsigned st: 6; /* symbol type (e.g. local, param, etc.) */
    unsigned sc: 5; /* storage class (e.g. text, bss, etc.) */
    unsigned reserved: 1;
    unsigned index;  /* index to symbol or auxiliary tables */
};
```

SEE ALSO

The chapter on "The Symbol Table" in the Assembly Language Programmer's Guide.
ldfcn(2).

April 1990 - 1 - Version 4.0
NAME

sys_id – system identification file

DESCRIPTION

The file /etc/sys_id contains the name by which the system will be known on communications networks. The name should be no more than eight lower-case letters and digits (to maintain compatibility with foreign networks) and be terminated with a trailing newline. During system startup this file is read by the script /etc/rc2.d/S20syssetup and the contents are passed as a parameter to hostname(1) to initialize the system name. Once this has been done, this name will returned by the commands hostname(1) and uname(1) and the system calls gethostname(2) and uname(2).

FILES

/etc/sys_id

SEE ALSO

hostname(1), uname(1), gethostname(2), uname(2).
NAME

system – system configuration information table

DESCRIPTION

This file is used by the lboot program to obtain configuration information. This file generally contains information used to determine if specified hardware exists, a list of software drivers to include in the load, the assignment of system devices such as pipedev and swapdev, as well as instructions for manually overriding the drivers selected by the self-configuring boot process.

The syntax of the system file is given below. The parser for the /usr/sysgen/system file is case sensitive. All upper case strings in the syntax below should be upper case in the /usr/sysgen/system file as well. Non-terminal symbols are enclosed in angle brackets "<>" while optional arguments are enclosed in square brackets "[]". Ellipses "..." indicate optional repetition of the argument for that line.

\[
\begin{align*}
<\text{name}> & ::= \text{master file name from /master.d directory} \\
<\text{func}> & ::= \text{interrupt function name} \\
<\text{device}> & ::= \text{special device name | DEV(<major>,<minor>)} \\
<\text{major}> & ::= <\text{number}> \\
<\text{minor}> & ::= <\text{number}> \\
<\text{proc}> & ::= \text{processor # as interpreted by runon(1)} \\
<\text{number}> & ::= \text{decimal, octal or hex literal}
\end{align*}
\]

Lboot can determine if hardware exists for a given module by use of probe commands. The syntax for probe commands is:

\[
\begin{align*}
<\text{probe_cmd}> & ::= \text{probe=<number> [ probe_size=<number> ]} \\
& \quad | <\text{extended_probe}> \\
& \quad | <\text{extended_probe}> ::= \text{exprobe=<probe_sequence>} \\
& \quad \quad | \text{exprobe=(<probe_sequence>,<probe_sequence>,...)} \\
& \quad | <\text{probe_sequence}> ::= (\langle\text{seq},<\text{address},<\text{size},<\text{value},<\text{mask}\rangle}) \\
& \quad | <\text{seq}> ::= \text{a sequence of 1 or more r's or w's, indicating a read} \\
& \quad \quad \text{from <address>, or a write to <address>.} \\
& \quad | <\text{address}> ::= <\text{number}> \\
& \quad | <\text{size}> ::= <\text{number}> \\
& \quad | <\text{value}> ::= <\text{number}> \\
& \quad | <\text{mask}> ::= <\text{number}>
\end{align*}
\]

As shown from the grammar, there are two forms of probe commands. The first allows the specification of an address to read, and optionally, a number of bytes to read. If a probe address is specified, the boot program will attempt to read probe_size bytes (default 4) to determine if the hardware...
exists for the module. If the read succeeds, the hardware will be assumed to exist, and the module will be included.

The extended form specifies a sequence of one or more five-tuples used to determine if the hardware exists. Each five-tuple specifies a read/write sequence, an address to read or write, a size of up to four bytes, a value, and a mask. Then, for each five-tuple, the following is performed:

for each element in command do
  if element == 'w' then
    if write(address, value & mask, size) != size then
      failure
  if element == 'r' then
    if read(address, temp, size) != size then
      failure
  if temp & mask != value & mask then
    failure

The lines listed below may appear in any order. Blank lines may be inserted at any point. Comment lines must begin with an asterisk. Entries for VECTOR, EXCLUDE and INCLUDE are cumulative. For all other entries, the last line to appear in the file is used -- any earlier entries are ignored.

VECTOR: (Note: this is one line) module=<fname> [ intr=<func> ]
[ vector=<number> ipl=<number> unit=<number> ] [ base=<number> ]
[ base2=<number> ] [ base3=<number> ]
[ <probe_cmd> ]

specifies hardware to conditionally load. If a probe command is specified, the boot program will perform the probe sequence, as discussed above. If the sequence succeeds, the module is included. If a probe sequence is not specified, the hardware will be assumed to exist. The intr function specifies the name of the module's interrupt handler. If it is not specified, the prefix defined in the module's master file (see master(4)) is concatenated with the string "intr", and, if a routine with that name is found in the module's object (which resides in the directory /usr/sysgen/boot, it is used as the interrupt routine. If the triplet (vector, ipl, unit, base) is specified, a VME interrupt structure is assigned, using the corresponding VME address "vector", priority level "ipl", unit "unit". If the modules’ object contains a routine whose name is the concatenation of the master file prefix and "edtinit", that routine is involved once at startup and passed a pointer to an edt structure which contains the values for base, base2, base3, and a pointer to
the VME interrupt structure.

EXCLUDE: [ <string> ] ... specifies drivers to exclude from the load even if the device is found via VECTOR information.

INCLUDE: [ <string>[[<number>]] ] ... specifies software drivers or loadable modules to be included in the load. This is necessary to include the drivers for software "devices". The optional <number> (parenthesis required) specifies the number of "devices" to be controlled by the driver (defaults to 1). This number corresponds to the builtin variable ##c which may be referred to by expressions in part two of the /usr/sysgen/master file.

ROOTDEV: <device> identifies the device containing the root file system.

SWAPDEV: <device> <number> <number> identifies the device to be used as swap space, the block number the swap space starts at, and the number of swap blocks available.

PIPEDEV: <device> identifies the device to be used for pipe space.

DUMPDEV: <device> identifies the device to be used for kernel dumps.

IPL: <IRQ level> <proc> send VME interrupt at <IRQ level> to <proc>. If <proc> does not exist at run time, the kernel will default to use processor 0.

NETWORKPROC: <proc> select <proc> to handle all of kernel's networking activities. If <proc> does not exist at run time, the kernel will default to use processor 0.

USE: [ <string>[[<number>]] [ <extended_probe> ] ] ... If the driver is present, it is the same as INCLUDE. Behaves like EXCLUDE if the module or driver is not present in /usr/sysgen/boot.

KERNEL: [ <string> ] ... Specifies the module containing the heart of the operating system. It must be present in the system file.

LCOPTS
LDROPTS

are option strings given to cc(1) and ld(1) respectively, to compile the master.c file and link the operating system.

FILES
/usr/sysgen/system
/usr/include/sys/edt.h

April 1990 - 3 - Version 5.0
SEE ALSO
  master(4).
  lboot(1M)
NAME
term - format of compiled term file.

SYNOPSIS
/usr/lib/terminfo/?!/*

DESCRIPTION
Compiled \texttt{terminfo}(4) descriptions are placed under the directory
\texttt{/usr/lib/terminfo}. In order to avoid a linear search of a huge UNIX system
directory, a two-level scheme is used: \texttt{/usr/lib/terminfo/c/name} where \texttt{name}
is the name of the terminal, and \texttt{c} is the first character of \texttt{name}. Thus,\texttt{att4425} can be found in the file \texttt{/usr/lib/terminfo/ata/att4425}. Synonyms for
the same terminal are implemented by multiple links to the same compiled
file.

The format has been chosen so that it will be the same on all hardware. An
8-bit byte is assumed, but no assumptions about byte ordering or sign exten-
sion are made. Thus, these binary \texttt{terminfo}(4) files can be transported to
other hardware with 8-bit bytes.

Short integers are stored in two 8-bit bytes. The first byte contains the least
significant 8 bits of the value, and the second byte contains the most
significant 8 bits. (Thus, the value represented is \texttt{256*second+first}.) The
value \texttt{-1} is represented by \texttt{0377,0377}, and the value \texttt{-2} is represented by
\texttt{0376,0377}; other negative values are illegal. Computers where this does
not correspond to the hardware read the integers as two bytes and compute
the result, making the compiled entries portable between machine types.
The \texttt{-1} generally means that a capability is missing from this terminal. The
\texttt{-2} means that the capability has been cancelled in the \texttt{terminfo}(4) source
and also is to be considered missing.

The compiled file is created from the source file descriptions of the termi-
nals (see the \texttt{-I} option of \texttt{infocmp}(1M)) by using the \texttt{terminfo}(4) compiler,
\texttt{tic}(1M), and read by the routine \texttt{setupterm()}. (See \texttt{curses}(3X).) The file
is divided into six parts: the header, terminal names, boolean flags,
numbers, strings, and string table.

The header section begins the file. This section contains six short integers
in the format described below. These integers are (1) the magic number
(octal \texttt{0432}); (2) the size, in bytes, of the names section; (3) the number of
bytes in the boolean section; (4) the number of short integers in the numbers
section; (5) the number of offsets (short integers) in the strings section; (6)
the size, in bytes, of the string table.

The terminal names section comes next. It contains the first line of the \texttt{ter-
minfo}(4) description, listing the various names for the terminal, separated
by the bar \texttt{(1)} character (see \texttt{term}(5)). The section is terminated with an
ASCII NUL character.

April 1990 - 1 - Version 5.0
The boolean flags have one byte for each flag. This byte is either 0 or 1 as the flag is present or absent. The value of 2 means that the flag has been cancelled. The capabilities are in the same order as the file `<term.h>`.

Between the boolean section and the number section, a null byte will be inserted, if necessary, to ensure that the number section begins on an even byte. All short integers are aligned on a short word boundary.

The numbers section is similar to the boolean flags section. Each capability takes up two bytes, and is stored as a short integer. If the value represented is −1 or −2, the capability is taken to be missing.

The strings section is also similar. Each capability is stored as a short integer, in the format above. A value of −1 or −2 means the capability is missing. Otherwise, the value is taken as an offset from the beginning of the string table. Special characters in `\^X` or `\c` notation are stored in their interpreted form, not the printing representation. Padding information (`$<n+m>` and parameter information (`%x`) are stored intact in uninterpreted form.

The final section is the string table. It contains all the values of string capabilities referenced in the string section. Each string is null terminated.

Note that it is possible for `setupterm()` to expect a different set of capabilities than are actually present in the file. Either the database may have been updated since `setupterm()` has been recompiled (resulting in extra unrecognized entries in the file) or the program may have been recompiled more recently than the database was updated (resulting in missing entries). The routine `setupterm()` must be prepared for both possibilities – this is why the numbers and sizes are included. Also, new capabilities must always be added at the end of the lists of boolean, number, and string capabilities.

Some limitations: total compiled entries cannot exceed 4096 bytes; all entries in the name field cannot exceed 128 bytes.

**FILES**

/usr/lib/terminfo/*

compiled terminal description database

/usr/include/term.h

`terminfo(4)` header file

**SEE ALSO**

curses(3X), terminfo(4), term(5).

infocomp(1M) in the *System Administrator's Reference Manual*.

Chapter 9 of the *Programmer's Guide*. 
NAME
terminfo – terminal capability data base

SYNOPSIS
/usr/lib/terminfo/2/*

DESCRIPTION

terminfo is a compiled database (see tic(1M)) describing the capabilities of
terminals. Terminals are described in terminfo source descriptions by giv-
ing a set of capabilities which they have, by describing how operations are
performed, by describing padding requirements, and by specifying initiali-
ization sequences. This database is used by applications programs, such as
vi(1) and curses(3X), so they can work with a variety of terminals without
changes to the programs. To obtain the source description for a terminal,
use the -I option of infocmp(1M).

Entries in terminfo source files consist of a number of comma-separated
fields. White space after each comma is ignored. The first line of each ter-
minal description in the terminfo database gives the name by which term-
info knows the terminal, separated by bar (|) characters. The first name
given is the most common abbreviation for the terminal (this is the one to
use to set the environment variable TERM in $HOME/.profile; see
profile(4)), the last name given should be a long name fully identifying the
terminal, and all others are understood as synonyms for the terminal name.
All names but the last should contain no blanks and must be unique in the
first 14 characters; the last name may contain blanks for readability.

Terminal names (except for the last, verbose entry) should be chosen using
the following conventions. The particular piece of hardware making up the
terminal should have a root name chosen, for example, for the AT&T 4425
terminal, att4425. Modes that the hardware can be in, or user preferences,
should be indicated by appending a hyphen and an indicator of the mode.
See term(5) for examples and more information on choosing names and
synonyms.

CAPABILITIES

In the table below, the Variable is the name by which the C programmer (at
the terminfo level) accesses the capability. The Capname is the short name
for this variable used in the text of the database. It is used by a person
updating the database and by the put(1) command when asking what the
value of the capability is for a particular terminal. The Termcap Code is a
two-letter code that corresponds to the old termcap capability name.

Capability names have no hard length limit, but an informal limit of 5 char-
acters has been adopted to keep them short. Whenever possible, names are
chosen to be the same as or similar to the ANSI X3.64-1979 standard.
Semantics are also intended to match those of the specification.
All string capabilities listed below may have padding specified, with the exception of those used for input. Input capabilities, listed under the Strings section in the table below, have names beginning with key_. The following indicators may appear at the end of the Description for a variable:

(G) indicates that the string is passed through tparm() with parameters (parms) as given (#i).

(*) indicates that padding may be based on the number of lines affected.

(#i) indicates the i\textsuperscript{th} parameter.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Cap-name</th>
<th>Termcap Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto_left_margin</td>
<td>bw</td>
<td>bw</td>
<td>\texttt{cub1} wraps from column 0 to last column</td>
</tr>
<tr>
<td>auto_right_margin</td>
<td>am</td>
<td>am</td>
<td>Terminal has automatic margins</td>
</tr>
<tr>
<td>back_color_eras</td>
<td>bce</td>
<td>be</td>
<td>Screen erased with background color</td>
</tr>
<tr>
<td>can_change</td>
<td>ccc</td>
<td>cc</td>
<td>Terminal can re-define existing color</td>
</tr>
<tr>
<td>cceol_standout_glitch</td>
<td>xhp</td>
<td>xs</td>
<td>Standout not erased by overwriting (hp)</td>
</tr>
<tr>
<td>eat_newline_glitch</td>
<td>xenl</td>
<td>xn</td>
<td>Newline ignored after 80 cols (Concept)</td>
</tr>
<tr>
<td>erase_overstrike</td>
<td>eo</td>
<td>eo</td>
<td>Can erase overstrikes with a blank</td>
</tr>
<tr>
<td>generic_type</td>
<td>gn</td>
<td>gn</td>
<td>Generic line type (e.g. dialup, switch).</td>
</tr>
<tr>
<td>hard-copy</td>
<td>hc</td>
<td>hc</td>
<td>Hardcopy terminal</td>
</tr>
<tr>
<td>hard_cursor</td>
<td>chts</td>
<td>HC</td>
<td>Cursor is hard to see</td>
</tr>
<tr>
<td>has_meta_key</td>
<td>km</td>
<td>km</td>
<td>Has a meta key (shift, sets parity bit)</td>
</tr>
<tr>
<td>has_status_line</td>
<td>hs</td>
<td>hs</td>
<td>Has extra &quot;status line&quot;</td>
</tr>
<tr>
<td>hue_lightness_saturation</td>
<td>hls</td>
<td>hl</td>
<td>Terminal uses only HLS color notation (Tektronix)</td>
</tr>
<tr>
<td>insert_null_glitch</td>
<td>in</td>
<td>in</td>
<td>Insert mode distinguishes nulls</td>
</tr>
<tr>
<td>memory_above</td>
<td>da</td>
<td>da</td>
<td>Display may be retained above the screen</td>
</tr>
<tr>
<td>memory_below</td>
<td>db</td>
<td>db</td>
<td>Display may be retained below the screen</td>
</tr>
<tr>
<td>move_insert_mode</td>
<td>mir</td>
<td>mi</td>
<td>Safe to move while in insert mode</td>
</tr>
<tr>
<td>move_standout_mode</td>
<td>msgr</td>
<td>ms</td>
<td>Safe to move in standout modes</td>
</tr>
<tr>
<td>needs_xon_xoff</td>
<td>nxon</td>
<td>nx</td>
<td>Padding won't work, xon/xoff required</td>
</tr>
<tr>
<td>no_esc_ctlc</td>
<td>xsb</td>
<td>xb</td>
<td>Beehive (f1=escape, f2=ctrl C)</td>
</tr>
<tr>
<td>non_rev_rncup</td>
<td>nrnc</td>
<td>NR</td>
<td>\texttt{smcup} does not reverse \texttt{rncup}</td>
</tr>
<tr>
<td>no_pad_char</td>
<td>npc</td>
<td>NP</td>
<td>Pad character doesn't exist</td>
</tr>
<tr>
<td>overstrike</td>
<td>os</td>
<td>os</td>
<td>Terminal overstrikes on hard-copy terminal</td>
</tr>
<tr>
<td>ptrtr_silent</td>
<td>mc5i</td>
<td>5i</td>
<td>Printer won't echo on screen</td>
</tr>
<tr>
<td>status_line_esc_ok</td>
<td>eslok</td>
<td>es</td>
<td>Escape can be used on the status line</td>
</tr>
<tr>
<td>dest_tabs_magic_smso</td>
<td>xt</td>
<td>xt</td>
<td>Destructive tabs, magic smso char (1061)</td>
</tr>
<tr>
<td>Term</td>
<td>Value</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------</td>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>tilde_glitch</td>
<td>hz</td>
<td>Hazeltine; can’t print tildes(’).</td>
<td></td>
</tr>
<tr>
<td>transparent_underline</td>
<td>ul</td>
<td>Underline character overstrikes.</td>
<td></td>
</tr>
<tr>
<td>xon_xoff</td>
<td>xon</td>
<td>Terminal uses xon/xoff handshaking.</td>
<td></td>
</tr>
</tbody>
</table>

### Numbers

<table>
<thead>
<tr>
<th>Term</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>columns</td>
<td>cols</td>
<td>Number of columns in a line.</td>
</tr>
<tr>
<td>init_tabs</td>
<td>it</td>
<td>Tabs initially every # spaces.</td>
</tr>
<tr>
<td>label_height</td>
<td>lh</td>
<td>Number of rows in each label.</td>
</tr>
<tr>
<td>label_width</td>
<td>lw</td>
<td>Number of cols in each label.</td>
</tr>
<tr>
<td>lines</td>
<td>li</td>
<td>Number of lines on screen or page.</td>
</tr>
<tr>
<td>lines_of_memory</td>
<td>lm</td>
<td>Lines of memory if &gt; lines; 0 means varies.</td>
</tr>
<tr>
<td>magic_cookie_glitch</td>
<td>xmc</td>
<td>Number blank chars left by smso or rmso.</td>
</tr>
<tr>
<td>max_colors</td>
<td>colors</td>
<td>Maximum number of colors on the screen.</td>
</tr>
<tr>
<td>max_pairs</td>
<td>pairs</td>
<td>Maximum number of color-pairs on the screen.</td>
</tr>
<tr>
<td>no_color_video</td>
<td>ncv</td>
<td>Video attributes that can’t be used with colors.</td>
</tr>
<tr>
<td>num_labels</td>
<td>nlab</td>
<td>Number of labels on screen (start at 1).</td>
</tr>
<tr>
<td>padding_baud_rate</td>
<td>pb</td>
<td>Lowest baud rate where padding needed.</td>
</tr>
<tr>
<td>virtual_terminal</td>
<td>vt</td>
<td>Virtual terminal number (UNIX system).</td>
</tr>
<tr>
<td>width_status_line</td>
<td>wsl</td>
<td>Number of columns in status line.</td>
</tr>
</tbody>
</table>

### Strings

<table>
<thead>
<tr>
<th>Term</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>acs_chars</td>
<td>acsc</td>
<td>Graphic charset pairs aAbBcC - def=vt100+.</td>
</tr>
<tr>
<td>back_tab</td>
<td>cbt</td>
<td>Back tab</td>
</tr>
<tr>
<td>bell</td>
<td>be</td>
<td>Audible signal (bell).</td>
</tr>
<tr>
<td>carriage_return</td>
<td>cr</td>
<td>Carriage return (*).</td>
</tr>
<tr>
<td>change_scroll_region</td>
<td>csr</td>
<td>Change to lines #1 thru #2 (vt100) (G).</td>
</tr>
<tr>
<td>char_padding</td>
<td>rmp</td>
<td>Like ip but when in replace mode.</td>
</tr>
<tr>
<td>clear_all_tabs</td>
<td>tbc</td>
<td>Clear all tab stops.</td>
</tr>
<tr>
<td>clear_margins</td>
<td>mge</td>
<td>Clear left and right soft margins.</td>
</tr>
<tr>
<td>clear_screen</td>
<td>clear</td>
<td>Clear screen and home cursor (*).</td>
</tr>
<tr>
<td>clr_bol</td>
<td>eil</td>
<td>Clear to beginning of line, inclusive.</td>
</tr>
<tr>
<td>clr_eol</td>
<td>el</td>
<td>Clear to end of line.</td>
</tr>
<tr>
<td>clr_eos</td>
<td>ed</td>
<td>Clear to end of display (*).</td>
</tr>
<tr>
<td>column_address</td>
<td>hpa</td>
<td>Horizontal position absolute (G).</td>
</tr>
<tr>
<td>command_character</td>
<td>cmdch</td>
<td>Term. settable cmd char in prototype.</td>
</tr>
<tr>
<td>cursor_address</td>
<td>cup</td>
<td>Cursor motion to row #1 col #2 (G).</td>
</tr>
<tr>
<td>cursor_down</td>
<td>cud1</td>
<td>Down one line.</td>
</tr>
<tr>
<td>cursor_home</td>
<td>home</td>
<td>Home cursor (if no cup).</td>
</tr>
<tr>
<td>cursor_invisible</td>
<td>civis</td>
<td>Make cursor invisible.</td>
</tr>
<tr>
<td>cursor_left</td>
<td>cub1</td>
<td>Move cursor left one space.</td>
</tr>
<tr>
<td>cursor_mem_address</td>
<td>mrcup</td>
<td>Memory relative cursor addressing (G).</td>
</tr>
</tbody>
</table>

April 1990
cursor_normal cnorm ve Make cursor appear normal (undo vs/vi)
cursor_right cufl nd Non-destructive space (cursor right)
cursor_to_ll ll ll Last line, first column (if no cup)
cursor_up cuu1 up Upline (cursor up)
cursor_visible cvvis vs Make cursor very visible
delete_character dch1 dc Delete character (*)
delete_line dl1 dl Delete line (*)
dis_status_line dsl ds Disable status line
down_half_line hd hd Half-line down (forward 1/2 linefeed)
en_acs enacs eA Enable alternate char set
enter_alt_charset_mode smacs as Start alternate character set
enter_am_mode smam SA Turn on automatic margins
enter_blink_mode blink mb Turn on blinking
enter_bold_mode bold md Turn on bold (extra bright) mode
enter_ca_mode smcup ti String to begin programs that use cup
enter_delete_mode smdc dm Delete mode (enter)
enter_dim_mode dim mh Turn on half-bright mode
enter_insert_mode smir im Insert mode (enter);
enter_protected_mode prot mp Turn on protected mode
enter_reverse_mode rev mr Turn on reverse video mode
enter_secure_mode invis mk Turn on blank mode (chars invisible)
enter_standout_mode smso so Begin standout mode
enter_underline_mode smul us Start underscore mode
enter_xon_mode smxon SX Turn on xon/xoff handshaking
erase_chars ech ec Erase #1 characters (G)
exit_alt_charset_mode rmacs ae End alternate character set
exit_am_mode rmam RA Turn off automatic margins
exit_attribute_mode sgr0 me Turn off all attributes
exit_ca_mode mcup te String to end programs that use cup
exit_delete_mode mdc ed End delete mode
exit_insert_mode mmir ei End insert mode;
exit_standout_mode rmso se End standout mode
exit_underline_mode rmul ue End underscore mode
exit_xon_mode mxon RX Turn off xon/xoff handshaking
flash_screen flash vb Visible bell (may not move cursor)
form_feed ff ff Hardcopy terminal page eject (*)
from_status_line fsl fs Return from status line
init_1string is1 i1 Terminal initialization string
init_2string is2 is Terminal initialization string
init_3string is3 i3 Terminal initialization string
init_file if if Name of initialization file containing is
<table>
<thead>
<tr>
<th>Key</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>init_prog</td>
<td>ipro</td>
<td>Path name of program for init.</td>
</tr>
<tr>
<td>initialize_color</td>
<td>initc</td>
<td>Initialize the definition of color</td>
</tr>
<tr>
<td>initialize_pair</td>
<td>initp</td>
<td>Initialize color-pair</td>
</tr>
<tr>
<td>insert_character</td>
<td>ichl</td>
<td>Insert character</td>
</tr>
<tr>
<td>insert_line</td>
<td>ill</td>
<td>Add new blank line (+)</td>
</tr>
<tr>
<td>insert_padding</td>
<td>ip</td>
<td>Insert pad after character inserted (*)</td>
</tr>
<tr>
<td>key_a1</td>
<td>ka1</td>
<td>KEY_A1, 0534, Upper left of keypad</td>
</tr>
<tr>
<td>key_a3</td>
<td>ka3</td>
<td>KEY_A3, 0535, Upper right of keypad</td>
</tr>
<tr>
<td>key_b2</td>
<td>kb2</td>
<td>KEY_B2, 0536, Center of keypad</td>
</tr>
<tr>
<td>key_backspace</td>
<td>kbs</td>
<td>KEY_BACKSPACE, 0407, Sent by backspace key</td>
</tr>
<tr>
<td>key_beg</td>
<td>kbeg</td>
<td>KEY_BEG, 0542, Sent by beginning key</td>
</tr>
<tr>
<td>key_btab</td>
<td>kcbt</td>
<td>KEY_BTAB, 0541, Sent by back-tab key</td>
</tr>
<tr>
<td>key_cl</td>
<td>kcl</td>
<td>KEY_C1, 0537, Lower left of keypad</td>
</tr>
<tr>
<td>key_c3</td>
<td>kcs</td>
<td>KEY_C3, 0540, Lower right of keypad</td>
</tr>
<tr>
<td>key_cancel</td>
<td>kcan</td>
<td>KEY_CANCEL, 0543, Sent by cancel key</td>
</tr>
<tr>
<td>key_catab</td>
<td>ktbc</td>
<td>KEY_CATAB, 0526, Sent by clear-all-tabs key</td>
</tr>
<tr>
<td>key_clear</td>
<td>kclr</td>
<td>KEY_CLEAR, 0515, Sent by clear-screen or erase key</td>
</tr>
<tr>
<td>key_close</td>
<td>kclo</td>
<td>KEY_CLOSE, 0544, Sent by close key</td>
</tr>
<tr>
<td>key_command</td>
<td>kcmd</td>
<td>KEY_COMMAND, 0545, Sent by cmd (command) key</td>
</tr>
<tr>
<td>key_copy</td>
<td>kcpy</td>
<td>KEY_COPY, 0546, Sent by copy key</td>
</tr>
<tr>
<td>key_create</td>
<td>kcr</td>
<td>KEY_CREATE, 0547, Sent by create key</td>
</tr>
<tr>
<td>key_ctab</td>
<td>kctb</td>
<td>KEY_CTAB, 0525, Sent by clear-tab key</td>
</tr>
<tr>
<td>key_dc</td>
<td>kdc1</td>
<td>KEY_DC, 0512, Sent by delete-character key</td>
</tr>
<tr>
<td>key_dl</td>
<td>kdl</td>
<td>KEY_DL, 0510, Sent by delete-line key</td>
</tr>
<tr>
<td>key_down</td>
<td>kcud</td>
<td>KEY_DOWN, 0402, Sent by terminal down-arrow key</td>
</tr>
<tr>
<td>key_eic</td>
<td>kmir</td>
<td>KEY_EIC, 0514, Sent by rmir or smir in insert mode</td>
</tr>
<tr>
<td>key_end</td>
<td>kend</td>
<td>KEY_END, 0550, Sent by end key</td>
</tr>
<tr>
<td>key_enter</td>
<td>kent</td>
<td>KEY_ENTER, 0527, Sent by enter/send key</td>
</tr>
<tr>
<td>key_eol</td>
<td>kel</td>
<td>KEY_EOL, 0517, Sent by clear-to-end-of-line key</td>
</tr>
<tr>
<td>key_eos</td>
<td>ked</td>
<td>KEY_EOS, 0516, Sent by clear-to-end-of-screen key</td>
</tr>
<tr>
<td>key_exit</td>
<td>kext</td>
<td>KEY_EXIT, 0551, Sent by exit key</td>
</tr>
<tr>
<td>key_f0</td>
<td>kf0</td>
<td>KEY_F(0), 0410, Sent by function key f0</td>
</tr>
<tr>
<td>key_f1</td>
<td>kf1</td>
<td>KEY_F(1), 0411, Sent by function key f1</td>
</tr>
<tr>
<td>key_f2</td>
<td>kf2</td>
<td>KEY_F(2), 0412, Sent by function key f2</td>
</tr>
<tr>
<td>key_f3</td>
<td>kf3</td>
<td>k3</td>
</tr>
<tr>
<td>-------------</td>
<td>-----</td>
<td>------</td>
</tr>
<tr>
<td>key_f4</td>
<td>kf4</td>
<td>k4</td>
</tr>
<tr>
<td>key_f5</td>
<td>kf5</td>
<td>k5</td>
</tr>
<tr>
<td>key_f6</td>
<td>kf6</td>
<td>k6</td>
</tr>
<tr>
<td>key_f7</td>
<td>kf7</td>
<td>k7</td>
</tr>
<tr>
<td>key_f8</td>
<td>kf8</td>
<td>k8</td>
</tr>
<tr>
<td>key_f9</td>
<td>kf9</td>
<td>k9</td>
</tr>
<tr>
<td>key_f10</td>
<td>kf10</td>
<td>k10</td>
</tr>
<tr>
<td>key_f11</td>
<td>kf11</td>
<td>F1</td>
</tr>
<tr>
<td>key_f12</td>
<td>kf12</td>
<td>F2</td>
</tr>
<tr>
<td>key_f13</td>
<td>kf13</td>
<td>F3</td>
</tr>
<tr>
<td>key_f14</td>
<td>kf14</td>
<td>F4</td>
</tr>
<tr>
<td>key_f15</td>
<td>kf15</td>
<td>F5</td>
</tr>
<tr>
<td>key_f16</td>
<td>kf16</td>
<td>F6</td>
</tr>
<tr>
<td>key_f17</td>
<td>kf17</td>
<td>F7</td>
</tr>
<tr>
<td>key_f18</td>
<td>kf18</td>
<td>F8</td>
</tr>
<tr>
<td>key_f19</td>
<td>kf19</td>
<td>F9</td>
</tr>
<tr>
<td>key_f20</td>
<td>kf20</td>
<td>FA</td>
</tr>
<tr>
<td>key_f21</td>
<td>kf21</td>
<td>FB</td>
</tr>
<tr>
<td>key_f22</td>
<td>kf22</td>
<td>FC</td>
</tr>
<tr>
<td>key_f23</td>
<td>kf23</td>
<td>FD</td>
</tr>
<tr>
<td>key_f24</td>
<td>kf24</td>
<td>FE</td>
</tr>
<tr>
<td>key_f25</td>
<td>kf25</td>
<td>FF</td>
</tr>
<tr>
<td>key_f26</td>
<td>kf26</td>
<td>FG</td>
</tr>
<tr>
<td>key_f27</td>
<td>kf27</td>
<td>FH</td>
</tr>
<tr>
<td>key_f28</td>
<td>kf28</td>
<td>FI</td>
</tr>
<tr>
<td>key_f29</td>
<td>kf29</td>
<td>FJ</td>
</tr>
<tr>
<td>key_f30</td>
<td>kf30</td>
<td>FK</td>
</tr>
<tr>
<td>key_f31</td>
<td>kf31</td>
<td>FL</td>
</tr>
<tr>
<td>key_f32</td>
<td>kf32</td>
<td>FM</td>
</tr>
<tr>
<td>key_f33</td>
<td>kf33</td>
<td>FN</td>
</tr>
<tr>
<td>key_f34</td>
<td>kf34</td>
<td>FO</td>
</tr>
<tr>
<td>key_f35</td>
<td>kf35</td>
<td>FP</td>
</tr>
<tr>
<td>key_f36</td>
<td>kf36</td>
<td>FQ</td>
</tr>
<tr>
<td>key_f37</td>
<td>kf37</td>
<td>FR</td>
</tr>
<tr>
<td>key_f38</td>
<td>kf38</td>
<td>FS</td>
</tr>
<tr>
<td>key_f39</td>
<td>kf39</td>
<td>FT</td>
</tr>
<tr>
<td>key_f40</td>
<td>kf40</td>
<td>FU</td>
</tr>
<tr>
<td>key_f41</td>
<td>kf41</td>
<td>FV</td>
</tr>
<tr>
<td>key_f42</td>
<td>kf42</td>
<td>FW</td>
</tr>
<tr>
<td>key_f43</td>
<td>kf43</td>
<td>FX</td>
</tr>
<tr>
<td>key_f44</td>
<td>kf44</td>
<td>FY</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>key_f45</td>
<td>kf45</td>
<td>FZ</td>
</tr>
<tr>
<td>key_f46</td>
<td>kf46</td>
<td>Fa</td>
</tr>
<tr>
<td>key_f47</td>
<td>kf47</td>
<td>Fb</td>
</tr>
<tr>
<td>key_f48</td>
<td>kf48</td>
<td>Fc</td>
</tr>
<tr>
<td>key_f49</td>
<td>kf49</td>
<td>Fd</td>
</tr>
<tr>
<td>key_f50</td>
<td>kf50</td>
<td>Fe</td>
</tr>
<tr>
<td>key_f51</td>
<td>kf51</td>
<td>Ff</td>
</tr>
<tr>
<td>key_f52</td>
<td>kf52</td>
<td>Fg</td>
</tr>
<tr>
<td>key_f53</td>
<td>kf53</td>
<td>Fh</td>
</tr>
<tr>
<td>key_f54</td>
<td>kf54</td>
<td>Fi</td>
</tr>
<tr>
<td>key_f55</td>
<td>kf55</td>
<td>Fj</td>
</tr>
<tr>
<td>key_f56</td>
<td>kf56</td>
<td>Fk</td>
</tr>
<tr>
<td>key_f57</td>
<td>kf57</td>
<td>Fl</td>
</tr>
<tr>
<td>key_f58</td>
<td>kf58</td>
<td>Fm</td>
</tr>
<tr>
<td>key_f59</td>
<td>kf59</td>
<td>Fn</td>
</tr>
<tr>
<td>key_f60</td>
<td>kf60</td>
<td>Fo</td>
</tr>
<tr>
<td>key_f61</td>
<td>kf61</td>
<td>Fp</td>
</tr>
<tr>
<td>key_f62</td>
<td>kf62</td>
<td>Fq</td>
</tr>
<tr>
<td>key_f63</td>
<td>kf63</td>
<td>Fr</td>
</tr>
<tr>
<td>key_find</td>
<td>kfind</td>
<td>@0</td>
</tr>
<tr>
<td>key_help</td>
<td>khelp</td>
<td>%1</td>
</tr>
<tr>
<td>key_home</td>
<td>khome</td>
<td>kh</td>
</tr>
<tr>
<td>key_ic</td>
<td>kich</td>
<td>kI</td>
</tr>
<tr>
<td>key_il</td>
<td>kill</td>
<td>kA</td>
</tr>
<tr>
<td>key_left</td>
<td>kcubl</td>
<td>kl</td>
</tr>
<tr>
<td>key_ll</td>
<td>kl</td>
<td>kH</td>
</tr>
<tr>
<td>key_mark</td>
<td>kmrk</td>
<td>%2</td>
</tr>
<tr>
<td>key_message</td>
<td>kmsg</td>
<td>%3</td>
</tr>
<tr>
<td>key_move</td>
<td>kmov</td>
<td>%4</td>
</tr>
<tr>
<td>key_next</td>
<td>knxt</td>
<td>%5</td>
</tr>
<tr>
<td>key_npage</td>
<td>knp</td>
<td>kN</td>
</tr>
<tr>
<td>key_open</td>
<td>kopn</td>
<td>%6</td>
</tr>
<tr>
<td>key_options</td>
<td>kopt</td>
<td>%7</td>
</tr>
<tr>
<td>key_ppage</td>
<td>kpp</td>
<td>kP</td>
</tr>
<tr>
<td>key_previous</td>
<td>kprv</td>
<td>%8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>key_print</td>
<td>kprt</td>
<td>%9</td>
</tr>
<tr>
<td>key_redo</td>
<td>krdo</td>
<td>%0</td>
</tr>
</tbody>
</table>

April 1990 - 7 - Version 5.0
<table>
<thead>
<tr>
<th>Key Reference</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key_reference</td>
<td>kref</td>
<td>&amp;1 KEY_REFERENCE, 0564, Sent by reference key</td>
</tr>
<tr>
<td>key_refresh</td>
<td>krfr</td>
<td>&amp;2 KEY_REFRESH, 0565, Sent by refresh key</td>
</tr>
<tr>
<td>key_replace</td>
<td>krpl</td>
<td>&amp;3 KEY_REPLACE, 0566, Sent by replace key</td>
</tr>
<tr>
<td>key_restart</td>
<td>krest</td>
<td>&amp;4 KEY_RESTART, 0567, Sent by restart key</td>
</tr>
<tr>
<td>key_resume</td>
<td>kres</td>
<td>&amp;5 KEY_RESUME, 0570, Sent by resume key</td>
</tr>
<tr>
<td>key_right</td>
<td>kcurf</td>
<td>kr KEY_RIGHT, 0405, Sent by terminal right-arrow key</td>
</tr>
<tr>
<td>key_save</td>
<td>ksav</td>
<td>&amp;6 KEY_SAVE, 0571, Sent by save key</td>
</tr>
<tr>
<td>key_sbeg</td>
<td>kBEG</td>
<td>&amp;9 KEY_SBEG, 0572, Sent by shifted beginning key</td>
</tr>
<tr>
<td>key_scancel</td>
<td>kCAN</td>
<td>&amp;0 KEY_SCANCEL, 0573, Sent by shifted cancel key</td>
</tr>
<tr>
<td>key_scommand</td>
<td>kCMD</td>
<td>*1 KEY_SCOMMAND, 0574, Sent by shifted command key</td>
</tr>
<tr>
<td>key_scopy</td>
<td>kCPY</td>
<td>*2 KEY_SCOPY, 0575, Sent by shifted copy key</td>
</tr>
<tr>
<td>key_screate</td>
<td>kCRT</td>
<td>*3 KEY_SCREATE, 0576, Sent by shifted create key</td>
</tr>
<tr>
<td>key_sdc</td>
<td>kDC</td>
<td>*4 KEY_SDC, 0577, Sent by shifted delete-char key</td>
</tr>
<tr>
<td>key_sdl</td>
<td>kDL</td>
<td>*5 KEY_SDL, 0600, Sent by shifted delete-line key</td>
</tr>
<tr>
<td>key_select</td>
<td>kslt</td>
<td>*6 KEY_SELECT, 0601, Sent by select key</td>
</tr>
<tr>
<td>key_send</td>
<td>kEND</td>
<td>*7 KEY_SEND, 0602, Sent by shifted end key</td>
</tr>
<tr>
<td>key_seol</td>
<td>kEOL</td>
<td>*8 KEY_SEOL, 0603, Sent by shifted clear-line key</td>
</tr>
<tr>
<td>key_sexit</td>
<td>kEXT</td>
<td>*9 KEY_SEXIT, 0604, Sent by shifted exit key</td>
</tr>
<tr>
<td>key_sf</td>
<td>kind</td>
<td>kF KEY_SF, 0520, Sent by scroll-forward/down key</td>
</tr>
<tr>
<td>key_sfind</td>
<td>kFND</td>
<td>*0 KEY_SFIND, 0605, Sent by shifted find key</td>
</tr>
<tr>
<td>key_shelp</td>
<td>kHELP</td>
<td>#1 KEY_SHELP, 0606, Sent by shifted help key</td>
</tr>
<tr>
<td>key_shome</td>
<td>kHOME</td>
<td>#2 KEY_SHOME, 0607, Sent by shifted home key</td>
</tr>
<tr>
<td>key_sic</td>
<td>kIC</td>
<td>#3 KEY_SIC, 0610, Sent by shifted input key</td>
</tr>
<tr>
<td>key_sleft</td>
<td>kLFT</td>
<td>#4 KEY_SLEFT, 0611, Sent by shifted left-arrow key</td>
</tr>
<tr>
<td>key_smmessage</td>
<td>kMSG</td>
<td>%a KEY_SMMESSAGE, 0612, Sent by shifted message key</td>
</tr>
<tr>
<td>key_smmove</td>
<td>kMOV</td>
<td>%b KEY_SMMOVE, 0613, Sent by shifted move key</td>
</tr>
<tr>
<td>key_snxt</td>
<td>kNXT</td>
<td>%c KEY_SNEXT, 0614, Sent by shifted next key</td>
</tr>
<tr>
<td>key_soptions</td>
<td>kOPT</td>
<td>%d KEY_SOPTIONS, 0615, Sent by shifted options key</td>
</tr>
<tr>
<td>key_sprevious</td>
<td>kPRV</td>
<td>%e KEY_SPREVIOUS, 0616, Sent by shifted prev key</td>
</tr>
<tr>
<td>key_sprint</td>
<td>kPRT</td>
<td>%f KEY_SPRINT, 0617, Sent by shifted print key</td>
</tr>
<tr>
<td>key_sr</td>
<td>kR</td>
<td>%R KEY_SR, 0521, Sent by scroll-backward/up key</td>
</tr>
<tr>
<td>key_sredo</td>
<td>kRDO</td>
<td>%g KEY_SREDO, 0620, Sent by shifted redo key</td>
</tr>
<tr>
<td>key_sreplace</td>
<td>kRPL</td>
<td>%h KEY_SREPLACE, 0621, Sent by shifted replace key</td>
</tr>
<tr>
<td>Key</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>key_sright</td>
<td>kRIT</td>
<td>%i</td>
</tr>
<tr>
<td>key_srsume</td>
<td>kRES</td>
<td>%j</td>
</tr>
<tr>
<td>key_ssave</td>
<td>kSAV</td>
<td>!1</td>
</tr>
<tr>
<td>key_ssuspend</td>
<td>kSPD</td>
<td>!2</td>
</tr>
<tr>
<td>key_stab</td>
<td>khts</td>
<td>kT</td>
</tr>
<tr>
<td>key_sundo</td>
<td>kUND</td>
<td>!3</td>
</tr>
<tr>
<td>key_suspend</td>
<td>kspd</td>
<td>&amp;7</td>
</tr>
<tr>
<td>key undo</td>
<td>kund</td>
<td>&amp;8</td>
</tr>
<tr>
<td>key_up</td>
<td>kcuu1</td>
<td>ku</td>
</tr>
<tr>
<td>keypad_local</td>
<td>rmkx</td>
<td>ke</td>
</tr>
<tr>
<td>keypad_xmit</td>
<td>smkx</td>
<td>ks</td>
</tr>
<tr>
<td>lab_f0</td>
<td>1f0</td>
<td>10</td>
</tr>
<tr>
<td>lab_f1</td>
<td>1f1</td>
<td>11</td>
</tr>
<tr>
<td>lab_f2</td>
<td>1f2</td>
<td>12</td>
</tr>
<tr>
<td>lab_f3</td>
<td>1f3</td>
<td>13</td>
</tr>
<tr>
<td>lab_f4</td>
<td>1f4</td>
<td>14</td>
</tr>
<tr>
<td>lab_f5</td>
<td>1f5</td>
<td>15</td>
</tr>
<tr>
<td>lab_f6</td>
<td>1f6</td>
<td>16</td>
</tr>
<tr>
<td>lab_f7</td>
<td>1f7</td>
<td>17</td>
</tr>
<tr>
<td>lab_f8</td>
<td>1f8</td>
<td>18</td>
</tr>
<tr>
<td>lab_f9</td>
<td>1f9</td>
<td>19</td>
</tr>
<tr>
<td>lab_f10</td>
<td>1f10</td>
<td>la</td>
</tr>
<tr>
<td>label_off</td>
<td>rmln</td>
<td>LF</td>
</tr>
<tr>
<td>label_on</td>
<td>smln</td>
<td>LO</td>
</tr>
<tr>
<td>meta_off</td>
<td>mm</td>
<td>mo</td>
</tr>
<tr>
<td>meta_on</td>
<td>smm</td>
<td>mm</td>
</tr>
<tr>
<td>newline</td>
<td>nel</td>
<td>nw</td>
</tr>
<tr>
<td>orig_colors</td>
<td>oc</td>
<td>oc</td>
</tr>
<tr>
<td>orig_pair</td>
<td>op</td>
<td>op</td>
</tr>
<tr>
<td>pad_char</td>
<td>pad</td>
<td>pc</td>
</tr>
<tr>
<td>parm_dch</td>
<td>dch</td>
<td>DC</td>
</tr>
<tr>
<td>parm_delete_line</td>
<td>dl</td>
<td>DL</td>
</tr>
<tr>
<td>parm_down_cursor</td>
<td>cud</td>
<td>DO</td>
</tr>
<tr>
<td>parm_ich</td>
<td>ich</td>
<td>IC</td>
</tr>
<tr>
<td>parm_index</td>
<td>indn</td>
<td>SF</td>
</tr>
<tr>
<td>parm_insert_line</td>
<td>il</td>
<td>AL</td>
</tr>
<tr>
<td>parm_left_cursor</td>
<td>cub</td>
<td>LE</td>
</tr>
<tr>
<td>parm_right_cursor</td>
<td>cuf</td>
<td>RI</td>
</tr>
<tr>
<td>parm_rindex</td>
<td>rin</td>
<td>SR</td>
</tr>
</tbody>
</table>

April 1990 - 9 - Version 5.0
parm_up_cursor cuu UP Move cursor up #1 lines. (G*)
pkey_key pfkey pk Prog funct key #1 to type string #2
pkey_local pfloc pl Prog funct key #1 to execute string #2
pkey_xmit pfx px Prog funct key #1 to xmit string #2
plab_norm pln pn Prog label #1 to show string #2
print_screen mc0 ps Print contents of the screen
prtr_non mc5p pO Turn on the printer for #1 bytes
prtr_off mc4 pf Turn off the printer
prtr_on mc5 po Turn on the printer
repeat_char rep rp Repeat char #1 #2 times (G*)
req_for_input rfi RF Send next input char (for pts)
reset_1string rs1 r1 Reset terminal completely to sane modes
reset_2string rs2 r2 Reset terminal completely to sane modes
reset_3string rs3 r3 Reset terminal completely to sane modes
reset_file rf rf Name of file containing reset string
restore_cursor rc rc Restore cursor to position of last sc
row_address vpa cv Vertical position absolute (G)
save_cursor sc sc Save cursor position.
scroll_forward ind sf Scroll text up
scroll_reverse ri sr Scroll text down
set_attributes sgr sa Define the video attributes #1-#9 (G)
set_background setb Sb Set current background color
set_bottom_margin smgb Zk Set bottom margin at current line
set_bottom_margin_parm smgbp Zl Set bottom margin at line #1 or #2 lines from bottom
set_color_pair scp sp Set current color-pair
set_foreground_setf Sf Set current foreground color1
set_left_margin smgl ML Set left margin at current line %374%
set_left_margin_setf ML Set left margin at current line %374%
set_left_margin smgl ML Set soft left margin
set_right_margin smgr MR Set soft right margin
set_tab hts st Set a tab in all rows, current column.
set_window wind wi Current window is lines #1-#2 cols #3-#4 (G)
tab ht ta Tab to next 8 space hardware tab stop.
to_status_line tsl ts Go to status line, col #1 (G)
underline_char uc uc Underscore one char and move past it
up_half_line hu hu Half-line up (reverse 1/2 linefeed)
xoff_character xoffc XF X-off character
xon_character xonc XN X-on character

SAMPLE ENTRY
The following entry, which describes the Concept-100 terminal, is among the more complex entries in the terminfo file as of this writing.
Entries may continue onto multiple lines by placing white space at
the beginning of each line except the first.
Lines beginning with "##" are taken as comment lines.

Capabilities in
**terminfo**
are of three types:
boolean capabilities which indicate that the terminal has some particular
feature,
numeric capabilities giving the size of the terminal or particular features,
and string capabilities, which give a sequence which can be used to perform
particular terminal operations.

**Types of Capabilities**
All capabilities have names. For instance, the fact that the **Concept** has
automatic margins (i.e., an automatic return and linefeed when the end of a
line is reached) is indicated by the capability **am**. Hence the description of
the **Concept** includes **am**. Numeric capabilities are followed by the charac-
ter ‘#’ and then the value. Thus **cols**, which indicates the number of
columns the terminal has, gives the value 80 for the **Concept**. The value
may be specified in decimal, octal or hexadecimal using normal C
conventions.

Finally, string-valued capabilities, such as **el** (clear to end of line sequence)
are given by the two- to five-character capname, an ‘=’, and then a string
ending at the next following comma. A delay in milliseconds may appear
anywhere in such a capability, enclosed in `$<..>` brackets, as in
el=\EK$<3>, and padding characters are supplied by tputs() (see curses(3X)) to provide this delay. The delay can be either a number, e.g., 20, or a number followed by an ‘*’ (i.e., 3*), a ‘/’ (i.e., 5/), or both (i.e., 10*/). A ‘*’ indicates that the padding required is proportional to the number of lines affected by the operation, and the amount given is the per-affected-unit padding required. (In the case of insert character, the factor is still the number of lines affected. This is always one unless the terminal has in and the software uses it.) When a ‘*’ is specified, it is sometimes useful to give a delay of the form 3.5 to specify a delay per unit to tenths of milliseconds. (Only one decimal place is allowed.) A ‘/’ indicates that the padding is mandatory. Otherwise, if the terminal has xon defined, the padding information is advisory and will only be used for cost estimates or when the terminal is in raw mode. Mandatory padding will be transmitted regardless of the setting of xon.

A number of escape sequences are provided in the string valued capabilities for easy encoding of characters there. Both \E and \e map to an ESCAPE character, \x maps to a control-x for any appropriate x, and the sequences \n, \r, \t, \b, \f, and \s give a newline, linefeed, return, tab, backspace, formfeed, and space, respectively. Other escapes include: \ for caret (^); \ for backslash (\); \ for comma (,); \: for colon (:); and \x for null. (\x will actually produce \200, which does not terminate a string but behaves as a null character on most terminals.) Finally, characters may be given as three octal digits after a backslash (e.g., \123).

Sometimes individual capabilities must be commented out. To do this, put a period before the capability name. For example, see the second ind in the example above. Note that capabilities are defined in a left-to-right order and, therefore, a prior definition will override a later definition.

Preparing Descriptions

The most effective way to prepare a terminal description is by imitating the description of a similar terminal in terminfo and to build up a description gradually, using partial descriptions with vi(1) to check that they are correct. Be aware that a very unusual terminal may expose deficiencies in the ability of the terminfo file to describe it or the inability of vi(1) to work with that terminal. To test a new terminal description, set the environment variable TERMINO to a pathname of a directory containing the compiled description you are working on and programs will look there rather than in /usr/lib/terminfo. To get the padding for insert-line correct (if the terminal manufacturer did not document it) a severe test is to comment out xon, edit a large file at 9600 baud with vi(1), delete 16 or so lines from the middle of the screen, then hit the u key several times quickly. If the display is corrupted, more padding is usually needed. A similar test can be used for insert-character.
Basic Capabilities
The number of columns on each line for the terminal is given by the \texttt{cols} numeric capability. If the terminal has a screen, then the number of lines on the screen is given by the \texttt{lines} capability. If the terminal wraps around to the beginning of the next line when it reaches the right margin, then it should have the \texttt{am} capability. If the terminal can clear its screen, leaving the cursor in the home position, then this is given by the \texttt{clear} string capability. If the terminal overstrikes (rather than clearing a position when a character is struck over) then it should have the \texttt{os} capability. If the terminal is a printing terminal, with no soft copy unit, give it both \texttt{hc} and \texttt{os}. (\texttt{os} applies to storage scope terminals, such as Tektronix 4010 series, as well as hard-copy and APL terminals.) If there is a code to move the cursor to the left edge of the current row, give this as \texttt{cr}. (Normally this will be carriage return, control M.) If there is a code to produce an audible signal (bell, beep, etc) give this as \texttt{bel}. If the terminal uses the xon-xoff flow-control protocol, like most terminals, specify \texttt{xon}.

If there is a code to move the cursor one position to the left (such as backspace) that capability should be given as \texttt{cub1}. Similarly, codes to move to the right, up, and down should be given as \texttt{cuf1}, \texttt{cuu1}, and \texttt{cud1}. These local cursor motions should not alter the text they pass over; for example, you would not normally use \texttt{cuf1=8} because the space would erase the character moved over.

A very important point here is that the local cursor motions encoded in \texttt{terminfo} are undefined at the left and top edges of a screen terminal. Programs should never attempt to backspace around the left edge, unless \texttt{bw} is given, and should never attempt to go up locally off the top. In order to scroll text up, a program will go to the bottom left corner of the screen and send the \texttt{ind} (index) string.

To scroll text down, a program goes to the top left corner of the screen and sends the \texttt{ri} (reverse index) string. The strings \texttt{ind} and \texttt{ri} are undefined when not on their respective corners of the screen.

Parameterized versions of the scrolling sequences are \texttt{indn} and \texttt{rin} which have the same semantics as \texttt{ind} and \texttt{ri} except that they take one parameter, and scroll that many lines. They are also undefined except at the appropriate edge of the screen.

The \texttt{am} capability tells whether the cursor sticks at the right edge of the screen when text is output, but this does not necessarily apply to a \texttt{cuf1} from the last column. The only local motion which is defined from the left edge is if \texttt{bw} is given, then a \texttt{cub1} from the left edge will move to the right edge of the previous row. If \texttt{bw} is not given, the effect is undefined. This is useful for drawing a box around the edge of the screen, for example. If the terminal has switch selectable automatic margins, the \texttt{terminfo} file
usually assumes that this is on; i.e., am. If the terminal has a command 
which moves to the first column of the next line, that command can be 
given as nel (newline). It does not matter if the command clears the 
remainder of the current line, so if the terminal has no cr and if it may still 
be possible to craft a working nel out of one or both of them.

These capabilities suffice to describe hardcopy and screen terminals. Thus 
the model 33 teletype is described as 

33!tty33!tty!model 33 teletype,
    bel=’G, cols#72, cr=’M, cud1=’J, hc, ind=’J, os,

while the Lear Siegler ADM–3 is described as 

adm3!lsi adm3,
    am, bel=’G, clear=’Z, cols#80, cr=’M, cub1=’H, cud1=’J, 
    ind=’J, lines#24,

Parameterized Strings

Cursor addressing and other strings requiring parameters in the terminal are 
described by a parameterized string capability, with printf(3S)-like escapes 
(\%x) in it. For example, to address the cursor, the cup capability is given, 
using two parameters: the row and column to address to. (Rows and 
columns are numbered from zero and refer to the physical screen visible to 
the user, not to any unseen memory.) If the terminal has memory relative 
cursor addressing, that can be indicated by mrcup.

The parameter mechanism uses a stack and special \% codes to manipulate 
it in the manner of a Reverse Polish Notation (postfix) calculator. Typically 
a sequence will push one of the parameters onto the stack and then print it 
in some format. Often more complex operations are necessary. Binary 
operations are in postfix form with the operands in the usual order. That is, 
to get x–5 one would use \%gx\%{5}\%–.

The \% encodings have the following meanings:

\%o outputs ‘%’
\%[[:flags]][width.[precision]][doxXs]
    as in printf, flags are [+-#] and space
\%c    print pop() gives %c
\%p[1-9]    push \( i^{th} \)
\%P[a-z]    set variable [a-z] to pop()
\%g[a-z]    get variable [a-z] and push it
\%c    push char constant \( c \)
\%{\( nn \)}    push decimal constant \( nn \)
\%    push strlen(pop())
\%+ \%- \%* \%/ \%m
    arithmetic (%m is mod): push(pop() op pop())
%& %! %^ bit operations: push(pop() op pop())
%<= %=> %< logical operations: push(pop() op pop())
%A %O logical operations: and, or
%! %~ unary operations: push(op(pop()))
%i (for ANSI terminals)
   add 1 to first parm, if one parm present,
   or first two parms, if more than one parm present
%? expr %t thenpart %e elsepart %;
   if-then-else, %e elsepart is optional;
else-if's are possible ala Algol 68:
%? c_1 %t b_1 %e c_2 %t b_2 %e c_3 %t b_3 %e c_4 %t b_4 %e b_5 %;
  c_1 are conditions, b_1 are bodies.

If the "--" flag is used with "%[doxFs]", then a colon (:) must be placed between the "%" and the "--" to differentiate the flag from the binary "%--" operator, e.g "%:-16.16s".

Consider the Hewlett-Packard 2645, which, to get to row 3 and column 12, needs to be sent \E&a\03zY padded for 6 milliseconds. Note that the order of the rows and columns is inverted here, and that the row and column are zero-padded as two digits. Thus its \cup capability is "\cup=\E&a%p2%2.2dc%p1%2.2dY$<6>".

The Micro-Term ACT-IV needs the current row and column sent preceded by a "T", with the row and column simply encoded in binary, "\cup=T\%p1%c\%p2%c". Terminals which use "%c" need to be able to backspace the cursor (cu1), and to move the cursor up one line on the screen (cuuu1). This is necessary because it is not always safe to transmit \n, \r, and \t, as the system may change or discard them. (The library routines dealing with \terminfo set tty modes so that tabs are never expanded, so \t is safe to send. This turns out to be essential for the Ann Arbor 4080.)

A final example is the LSI ADM-3a, which uses row and column offset by a blank character, thus "\cup=\E=%p1\
   \s'++%c\%p2\n'++%c'". After sending "\E="", this pushes the first parameter, pushes the ASCII value for a space (32), adds them (pushing the sum on the stack in place of the two previous values), and outputs that value as a character. Then the same is done for the second parameter. More complex arithmetic is possible using the stack.

Cursor Motions

If the terminal has a fast way to home the cursor (to very upper left corner of screen) then this can be given as home; similarly a fast way of getting to the lower left-hand corner can be given as \l; this may involve going up with cuuu1 from the home position, but a program should never do this itself (unless \l does) because it can make no assumption about the effect of
moving up from the home position. Note that the home position is the same as addressing to (0,0): to the top left corner of the screen, not of memory. (Thus, the \texttt{\textbackslash EH} sequence on Hewlett-Packard terminals cannot be used for \texttt{home} without losing some of the other features on the terminal.)

If the terminal has row or column absolute-cursor addressing, these can be given as single parameter capabilities \texttt{hpa} (horizontal position absolute) and \texttt{vpa} (vertical position absolute). Sometimes these are shorter than the more general two-parameter sequence (as with the Hewlett-Packard 2645) and can be used in preference to \texttt{cup}. If there are parameterized local motions (e.g., move \textit{n} spaces to the right) these can be given as \texttt{cud}, \texttt{cub}, \texttt{cuf}, and \texttt{cuu} with a single parameter indicating how many spaces to move. These are primarily useful if the terminal does not have \texttt{cup}, such as the Tektronix 4025.

\textbf{Area Clears}

If the terminal can clear from the current position to the end of the line, leaving the cursor where it is, this should be given as \texttt{el}. If the terminal can clear from the beginning of the line to the current position inclusive, leaving the cursor where it is, this should be given as \texttt{el1}. If the terminal can clear from the current position to the end of the display, then this should be given as \texttt{ed}. \texttt{ed} is only defined from the first column of a line. (Thus, it can be simulated by a request to delete a large number of lines, if a true \texttt{ed} is not available.)

\textbf{Insert/delete line}

If the terminal can open a new blank line before the line where the cursor is, this should be given as \texttt{il}; this is done only from the first position of a line. The cursor must then appear on the newly blank line. If the terminal can delete the line which the cursor is on, then this should be given as \texttt{dl}; this is done only from the first position on the line to be deleted. Versions of \texttt{il} and \texttt{dl} which take a single parameter and insert or delete that many lines can be given as \texttt{il} and \texttt{dl}.

If the terminal has a settable destructive scrolling region (like the VT100) the command to set this can be described with the \texttt{csr} capability, which takes two parameters: the top and bottom lines of the scrolling region. The cursor position is, alas, undefined after using this command. It is possible to get the effect of insert or delete line using this command -- the \texttt{sc} and \texttt{rc} (save and restore cursor) commands are also useful. Inserting lines at the top or bottom of the screen can also be done using \texttt{ri} or \texttt{ind} on many terminals without a true insert/delete line, and is often faster even on terminals with those features.
To determine whether a terminal has destructive scrolling regions or non-destructive scrolling regions, create a scrolling region in the middle of the screen, place data on the bottom line of the scrolling region, move the cursor to the top line of the scrolling region, and do a reverse index (ri) followed by a delete line (dl1) or index (ind). If the data that was originally on the bottom line of the scrolling region was restored into the scrolling region by the dl1 or ind, then the terminal has non-destructive scrolling regions. Otherwise, it has destructive scrolling regions. Do not specify csr if the terminal has non-destructive scrolling regions, unless ind, ri, indn, rin, dl, and dl1 all simulate destructive scrolling.

If the terminal has the ability to define a window as part of memory, which all commands affect, it should be given as the parameterized string wind. The four parameters are the starting and ending lines in memory and the starting and ending columns in memory, in that order.

If the terminal can retain display memory above, then the da capability should be given; if display memory can be retained below, then db should be given. These indicate that deleting a line or scrolling a full screen may bring non-blank lines up from below or that scrolling back with ri may bring down non-blank lines.

Insert/Delete Character

There are two basic kinds of intelligent terminals with respect to insert/delete character operations which can be described using terminfo. The most common insert/delete character operations affect only the characters on the current line and shift characters off the end of the line rigidly. Other terminals, such as the Concept 100 and the Perkin Elmer Owl, make a distinction between typed and untyped blanks on the screen, shifting upon an insert or delete only to an untyped blank on the screen which is either eliminated, or expanded to two untyped blanks. You can determine the kind of terminal you have by clearing the screen and then typing text separated by cursor motions. Type “abc def” using local cursor motions (not spaces) between the abc and the def. Then position the cursor before the abc and put the terminal in insert mode. If typing characters causes the rest of the line to shift rigidly and characters to fall off the end, then your terminal does not distinguish between blanks and untyped positions. If the abc shifts over to the def which then move together around the end of the current line and onto the next as you insert, you have the second type of terminal, and should give the capability in, which stands for “insert null”. While these are two logically separate attributes (one line versus multiline insert mode, and special treatment of untyped spaces) we have seen no terminals whose insert mode cannot be described with the single attribute.
terminfo can describe both terminals which have an insert mode and terminals which send a simple sequence to open a blank position on the current line. Give as smir the sequence to get into insert mode. Give as rmir the sequence to leave insert mode. Now give as ich1 any sequence needed to be sent just before sending the character to be inserted. Most terminals with a true insert mode will not give ich1; terminals which send a sequence to open a screen position should give it here. (If your terminal has both, insert mode is usually preferable to ich1. Do not give both unless the terminal actually requires both to be used in combination.) If post-insert padding is needed, give this as a number of milliseconds padding in ip (a string option). Any other sequence which may need to be sent after an insert of a single character may also be given in ip. If your terminal needs both to be placed into an ‘insert mode’ and a special code to precede each inserted character, then both smir/rmir and ich1 can be given, and both will be used. The ich capability, with one parameter, n, will repeat the effects of ich1 n times.

If padding is necessary between characters typed while not in insert mode, give this as a number of milliseconds padding in rmp.

It is occasionally necessary to move around while in insert mode to delete characters on the same line (e.g., if there is a tab after the insertion position). If your terminal allows motion while in insert mode you can give the capability mir to speed up inserting in this case. Omitting mir will affect only speed. Some terminals (notably Datamedia’s) must not have mir because of the way their insert mode works.

Finally, you can specify dch1 to delete a single character, dch with one parameter, n, to delete n characters, and delete mode by giving smdc and rmde to enter and exit delete mode (any mode the terminal needs to be placed in for dch1 to work).

A command to erase n characters (equivalent to outputting n blanks without moving the cursor) can be given as ech with one parameter.

Highlighting, Underlining, and Visible Bells

If your terminal has one or more kinds of display attributes, these can be represented in a number of different ways. You should choose one display form as standout mode (see curses(3X)), representing a good, high contrast, easy-on-the-eyes, format for highlighting error messages and other attention getters. (If you have a choice, reverse-video plus half-bright is good, or reverse-video alone; however, different users have different preferences on different terminals.) The sequences to enter and exit standout mode are given as smso and rmso, respectively. If the code to change into or out of standout mode leaves one or even two blank spaces on the screen, as the TVI 912 and Teleray 1061 do, then xmc should be given to tell how many spaces are left.
Codes to begin underlining and end underlining can be given as smul and rmul respectively. If the terminal has a code to underline the current character and move the cursor one space to the right, such as the Micro-Term MIME, this can be given as uc.

Other capabilities to enter various highlighting modes include blink (blinking), bold (bold or extra-bright), dim (dim or half-bright), invis (blanking or invisible text), prot (protected), rev (reverse-video), sgr0 (turn off all attribute modes), smacs (enter alternate-character-set mode), and rmacs (exit alternate-character-set mode). Turning on any of these modes singly may or may not turn off other modes. If a command is necessary before alternate character set mode is entered, give the sequence in enacs (enable alternate-character-set mode).

If there is a sequence to set arbitrary combinations of modes, this should be given as sgr (set attributes), taking nine parameters. Each parameter is either 0 or non-zero, as the corresponding attribute is on or off. The nine parameters are, in order: standout, underline, reverse, blink, dim, bold, blank, protect, alternate character set. Not all modes need be supported by sgr, only those for which corresponding separate attribute commands exist. (See the example at the end of this section.)

Terminals with the "magic cookie" glitch (xmc) deposit special "cookies" when they receive mode-setting sequences, which affect the display algorithm rather than having extra bits for each character. Some terminals, such as the Hewlett-Packard 2621, automatically leave standout mode when they move to a new line or the cursor is addressed. Programs using standout mode should exit standout mode before moving the cursor or sending a newline, unless the msgr capability, asserting that it is safe to move in standout mode, is present.

If the terminal has a way of flashing the screen to indicate an error quietly (a bell replacement), then this can be given as flash; it must not move the cursor. A good flash can be done by changing the screen into reverse video, pad for 200 ms, then return the screen to normal video.

If the cursor needs to be made more visible than normal when it is not on the bottom line (to make, for example, a non-blinking underline into an easier to find block or blinking underline) give this sequence as cvvis. The boolean chts should also be given. If there is a way to make the cursor completely invisible, give that as civis. The capability cnorm should be given which undoes the effects of either of these modes.

If the terminal needs to be in a special mode when running a program that uses these capabilities, the codes to enter and exit this mode can be given as smcup and rmcup. This arises, for example, from terminals like the Concept with more than one page of memory. If the terminal has only memory
relative cursor addressing and not screen relative cursor addressing, a one
screen-sized window must be fixed into the terminal for cursor addressing
to work properly. This is also used for the Tektronix 4025, where smcup
sets the command character to be the one used by terminfo. If the smcup
sequence will not restore the screen after an rmcup sequence is output (to
the state prior to outputting rmcup), specify nrrmc.

If your terminal generates underlined characters by using the underline
character (with no special codes needed) even though it does not otherwise
overstrike characters, then you should give the capability ul. For terminals
where a character overstriking another leaves both characters on the screen,
give the capability os. If overstrikes are erasable with a blank, then this
should be indicated by giving eo.

Example of highlighting: assume that the terminal under question needs the
following escape sequences to turn on various modes.

<table>
<thead>
<tr>
<th>tparm parameter</th>
<th>attribute</th>
<th>escape sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>\E[0m</td>
<td></td>
</tr>
<tr>
<td>p1</td>
<td>standout</td>
<td>\E[0;4;7m</td>
</tr>
<tr>
<td>p2</td>
<td>underline</td>
<td>\E[0;3m</td>
</tr>
<tr>
<td>p3</td>
<td>reverse</td>
<td>\E[0;4m</td>
</tr>
<tr>
<td>p4</td>
<td>blink</td>
<td>\E[0;5m</td>
</tr>
<tr>
<td>p5</td>
<td>dim</td>
<td>\E[0;7m</td>
</tr>
<tr>
<td>p6</td>
<td>bold</td>
<td>\E[0;3;4m</td>
</tr>
<tr>
<td>p7</td>
<td>invis</td>
<td>\E[0;8m</td>
</tr>
<tr>
<td>p8</td>
<td>protect</td>
<td>not available</td>
</tr>
<tr>
<td>p9</td>
<td>altcharset</td>
<td>^O (off) ^N(on)</td>
</tr>
</tbody>
</table>

Note that each escape sequence requires a 0 to turn off other modes before
turning on its own mode. Also note that, as suggested above, standout is set
up to be the combination of reverse and dim. Also, since this terminal has
no bold mode, bold is set up as the combination of reverse and underline.
In addition, to allow combinations, such as underline+blink, the sequence
to use would be \E[0;3;5m. The terminal doesn’t have protect mode,
either, but that cannot be simulated in any way, so p8 is ignored. The
altcharset mode is different in that it is either ^O or ^N depending on
whether it is off or on. If all modes were to be turned on, the sequence
would be \E[0;3;4;5;7;8m^N.

Now look at when different sequences are output. For example, ;3 is output
when either p2 or p6 is true, that is, if either underline or bold modes are
turned on. Writing out the above sequences, along with their dependencies,
gives the following:
<table>
<thead>
<tr>
<th>sequence</th>
<th>when to output</th>
<th>terminfo translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>\E[0</td>
<td>always</td>
<td>\E[0</td>
</tr>
<tr>
<td>;3</td>
<td>if p2 or p6</td>
<td>%?%p2%p6%l%t;3%;</td>
</tr>
<tr>
<td>;4</td>
<td>if p1 or p3 or p6</td>
<td>%?%p1%p3%l%p6%l%t;4%;</td>
</tr>
<tr>
<td>;5</td>
<td>if p4</td>
<td>%?%p4%t;5%;</td>
</tr>
<tr>
<td>;7</td>
<td>if p1 or p5</td>
<td>%?%p1%p5%l%t;7%;</td>
</tr>
<tr>
<td>;8</td>
<td>if p7</td>
<td>%?%p7%t;8%;</td>
</tr>
<tr>
<td>m</td>
<td>always</td>
<td>m</td>
</tr>
<tr>
<td>\N or \O</td>
<td>if p9 \N, else \O</td>
<td>%?%p9%t%N%e\O%;</td>
</tr>
</tbody>
</table>

Putting this all together into the sgr sequence gives:

\[ \text{sgr=\E[0%?%p2%p6%l%t;3%;%?%p1%p3%l%p6%l%t;4%;%?%p5%t;}
\quad l5%;%?%p1%p5%l%t;7%;%?%p7%t;8%;m?%p9%t\%N%e\O%; \]

Keypad

If the terminal has a keypad that transmits codes when the keys are pressed, this information can be given. Note that it is not possible to handle terminals where the keypad only works in local (this applies, for example, to the unshifted Hewlett-Packard 2621 keys). If the keypad can be set to transmit or not transmit, give these codes as smkx and rmkx. Otherwise the keypad is assumed to always transmit.

The codes sent by the left arrow, right arrow, up arrow, down arrow, and home keys can be given as keub1, keuf1, kceu1, kced1, and khome respectively. If there are function keys such as f0, f1, ..., f63, the codes they send can be given as kf0, kf1, ..., kf63. If the first 11 keys have labels other than the default f0 through f10, the labels can be given as lf0, lf1, ..., lf10. The codes transmitted by certain other special keys can be given: kll (home down), kbs (backspace), ktbc (clear all tabs), kctab (clear the tab stop in this column), kclr (clear screen or erase key), kdhc1 (delete character), kdll1 (delete line), krmir (exit insert mode), kel (clear to end of line), ked (clear to end of screen), kich1 (insert character or enter insert mode), kill1 (insert line), knp (next page), kpp (previous page), kind (scroll forward/down), kri (scroll backward/up), khts (set a tab stop in this column). In addition, if the keypad has a 3 by 3 array of keys including the four arrow keys, the other five keys can be given as ka1, ka3, kb2, kc1, and kc3. These keys are useful when the effects of a 3 by 3 directional pad are needed. Further keys are defined above in the capabilities list.

Strings to program function keys can be given as pfkey, pfloc, and pfx. A string to program their soft-screen labels can be given as plrn. Each of these strings takes two parameters: the function key number to program (from 0 to 10) and the string to program it with. Function key numbers out of this range may program undefined keys in a terminal-dependent manner. The difference between the capabilities is that pfkey causes pressing the given
key to be the same as the user typing the given string; pttoc causes the string to be executed by the terminal in local mode; and pfx causes the string to be transmitted to the computer. The capabilities nlab, lw and lh define how many soft labels there are and their width and height. If there are commands to turn the labels on and off, give them in smln and rmln. smln is normally output after one or more pln sequences to make sure that the change becomes visible.

Tabs and Initialization

If the terminal has hardware tabs, the command to advance to the next tab stop can be given as ht (usually control I). A “backtab” command which moves leftward to the next tab stop can be given as cbt. By convention, if the teletype modes indicate that tabs are being expanded by the computer rather than being sent to the terminal, programs should not use ht or cbt even if they are present, since the user may not have the tab stops properly set. If the terminal has hardware tabs which are initially set every n spaces when the terminal is powered up, the numeric parameter it is given, showing the number of spaces the tabs are set to. This is normally used by tput init (see tput(1)) to determine whether to set the mode for hardware tab expansion and whether to set the tab stops. If the terminal has tab stops that can be saved in nonvolatile memory, the terminfo description can assume that they are properly set. If there are commands to set and clear tab stops, they can be given as tbc (clear all tab stops) and hts (set a tab stop in the current column of every row).

Other capabilities include: is1, is2, and is3, initialization strings for the terminal; iprog, the path name of a program to be run to initialize the terminal; and if, the name of a file containing long initialization strings. These strings are expected to set the terminal into modes consistent with the rest of the terminfo description. They must be sent to the terminal each time the user logs in and be output in the following order: run the program iprog; output is1; output is2; set the margins using mge, smgl and smgr; set the tabs using tbc and hts; print the file if; and finally output is3. This is usually done using the init option of tput(1); see profile(4).

Most initialization is done with is2. Special terminal modes can be set up without duplicating strings by putting the common sequences in is2 and special cases in is1 and is3. Sequences that do a harder reset from a totally unknown state can be given as rs1, rs2, rf, and rs3, analogous to is1, is2, is3, and if. (The method using files, if and rf, is used for a few terminals, from /usr/lib/tabset/*; however, the recommended method is to use the initialization and reset strings.) These strings are output by tput reset, which is used when the terminal gets into a wedged state. Commands are normally placed in rs1, rs2, rs3, and rf only if they produce annoying effects on the screen and are not necessary when logging in. For example, the
command to set a terminal into 80-column mode would normally be part of is2, but on some terminals it causes an annoying glitch on the screen and is not normally needed since the terminal is usually already in 80-column mode.

If a more complex sequence is needed to set the tabs than can be described by using thc and hts, the sequence can be placed in is2 or if.

If there are commands to set and clear margins, they can be given as mge (clear all margins), smgl (set left margin), and smgr (set right margin).

Dclays

Certain capabilities control padding in the tty(7) driver. These are primarily needed by hard-copy terminals, and are used by tput init to set tty modes appropriately. Delays embedded in the capabilities cr, ind, cub1, ff, and tab can be used to set the appropriate delay bits to be set in the tty driver. If pb (padding baud rate) is given, these values can be ignored at baud rates below the value of pb.

Status Lines

If the terminal has an extra "status line" that is not normally used by software, this fact can be indicated. If the status line is viewed as an extra line below the bottom line, into which one can cursor address normally (such as the Heathkit h19's 25th line, or the 24th line of a VT100 which is set to a 23-line scrolling region), the capability hs should be given. Special strings that go to a given column of the status line and return from the status line can be given as tsl and fsl. (fsl must leave the cursor position in the same place it was before tsl. If necessary, the sc and rc strings can be included in tsl and fsl to get this effect.) The capability tsl takes one parameter, which is the column number of the status line the cursor is to be moved to.

If escape sequences and other special commands, such as tab, work while in the status line, the flag eslok can be given. A string which turns off the status line (or otherwise erases its contents) should be given as dsl. If the terminal has commands to save and restore the position of the cursor, give them as se and rc. The status line is normally assumed to be the same width as the rest of the screen, e.g., cols. If the status line is a different width (possibly because the terminal does not allow an entire line to be loaded) the width, in columns, can be indicated with the numeric parameter wsl.

Line Graphics

If the terminal has a line drawing alternate character set, the mapping of glyph to character would be given in acsc. The definition of this string is based on the alternate character set used in the DEC VT100 terminal, extended slightly with some characters from the AT&T 4410v1 terminal.
<table>
<thead>
<tr>
<th>glyph name</th>
<th>vt100+ character</th>
</tr>
</thead>
<tbody>
<tr>
<td>arrow pointing right</td>
<td>+</td>
</tr>
<tr>
<td>arrow pointing left</td>
<td>,</td>
</tr>
<tr>
<td>arrow pointing down</td>
<td>.</td>
</tr>
<tr>
<td>solid square block</td>
<td>0</td>
</tr>
<tr>
<td>lantern symbol</td>
<td>I</td>
</tr>
<tr>
<td>arrow pointing up</td>
<td>-</td>
</tr>
<tr>
<td>diamond</td>
<td>‘</td>
</tr>
<tr>
<td>checker board (stipple)</td>
<td>a</td>
</tr>
<tr>
<td>degree symbol</td>
<td>f</td>
</tr>
<tr>
<td>plus/minus</td>
<td>g</td>
</tr>
<tr>
<td>board of squares</td>
<td>h</td>
</tr>
<tr>
<td>lower right corner</td>
<td>j</td>
</tr>
<tr>
<td>upper right corner</td>
<td>k</td>
</tr>
<tr>
<td>upper left corner</td>
<td>l</td>
</tr>
<tr>
<td>lower left corner</td>
<td>m</td>
</tr>
<tr>
<td>plus</td>
<td>n</td>
</tr>
<tr>
<td>scan line 1</td>
<td>o</td>
</tr>
<tr>
<td>horizontal line</td>
<td>q</td>
</tr>
<tr>
<td>scan line 9</td>
<td>s</td>
</tr>
<tr>
<td>left tee (├)</td>
<td>t</td>
</tr>
<tr>
<td>right tee (─)</td>
<td>u</td>
</tr>
<tr>
<td>bottom tee (┘)</td>
<td>v</td>
</tr>
<tr>
<td>top tee (┌)</td>
<td>w</td>
</tr>
<tr>
<td>vertical line</td>
<td>x</td>
</tr>
<tr>
<td>bullet</td>
<td>~</td>
</tr>
</tbody>
</table>

The best way to describe a new terminal’s line graphics set is to add a third column to the above table with the characters for the new terminal that produce the appropriate glyph when the terminal is in the alternate character set mode. For example,
glyph name       vt100+       new tty       
              char       char
upper left corner  l          R
lower left corner  m          F
upper right corner k          T
lower right corner j          G
horizontal line   q          
vertical line x          

Now write down the characters left to right, as in "acsc=IRmFkTjGq\x.".

Color Manipulation

Let us define two methods of color manipulation: the Tektronix method and the HP method. The Tektronix method uses a set of N predefined colors (usually 8) from which a user can select "current" foreground and background colors. Thus a terminal can support up to N colors mixed into N*N color-pairs to be displayed on the screen at the same time. When using an HP method the user cannot define the foreground independently of the background, or vice-versa. Instead, the user must define an entire color-pair at once. Up to M color-pairs, made from 2*M different colors, can be defined this way. Most existing color terminals belong to one of these two classes of terminals.

The numeric variables colors and pairs define the number of colors and color-pairs that can be displayed on the screen at the same time. If a terminal can change the definition of a color (for example, the Tektronix 4100 and 4200 series terminals), this should be specified with ccc (can change color). To change the definition of a color (Tektronix method), use initc (initialize color). It requires four arguments: color number (ranging from 0 to colors−1) and three RGB (red, green, and blue) values (ranging from 0 to 1000).

Tektronix 4100 series terminals use a type of color notation called HLS (Hue Lightness Saturation) instead of RGB color notation. For such terminals one must define a boolean variable hls. The last three arguments to the initc string would then be HLS values: H, ranging from 0 to 360; and L and S, ranging from 0 to 100.

If a terminal can change the definitions of colors, but uses a color notation different from RGB and HLS, a mapping to either RGB or HLS must be developed.

To set current foreground or background to a given color, use setf (set foreground) and setb (set background). They require one parameter: the number of the color. To initialize a color-pair (HP method), use initp (initialize pair). It requires seven parameters: the number of a color-pair
(range=0 to \texttt{pairs}-1), and six RGB values: three for the foreground followed by three for the background. (Each of these groups of three should be in the order RGB.) When \texttt{initc} or \texttt{initp} are used, RGB or HLS arguments should be in the order "red, green, blue" or "hue, lightness, saturation"), respectively. To make a color-pair current, use \texttt{scp} (set color-pair). It takes one parameter, the number of a color-pair.

Some terminals (for example, most color terminal emulators for PCs) erase areas of the screen with current background color. In such cases, \texttt{bee} (background color erase) should be defined. The variable \texttt{op} (original pair) contains a sequence for setting the foreground and the background colors to what they were at the terminal start-up time. Similarly, \texttt{oc} (original colors) contains a control sequence for setting all colors (for the Tektronix method) or color-pairs (for the HP method) to the values they had at the terminal start-up time.

Some color terminals substitute color for video attributes. Such video attributes should not be combined with colors. Information about these video attributes should be packed into the \texttt{ncv} (no color video) variable. There is a one-to-one correspondence between the nine least significant bits of that variable and the video attributes. The following table depicts this correspondence.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Bit Position</th>
<th>Decimal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A_STANDOUT</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>A_UNDERLINE</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>A_REVERSE</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>A_BLINK</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>A_DIM</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>A_BOLD</td>
<td>5</td>
<td>32</td>
</tr>
<tr>
<td>A_INVIS</td>
<td>6</td>
<td>64</td>
</tr>
<tr>
<td>A_PROTECT</td>
<td>7</td>
<td>128</td>
</tr>
<tr>
<td>A_ALTCCHARSET</td>
<td>8</td>
<td>256</td>
</tr>
</tbody>
</table>

When a particular video attribute should not be used with colors, the corresponding \texttt{ncv} bit should be set to 1; otherwise it should be set to zero. To determine the information to pack into the \texttt{ncv} variable, you must add together the decimal values corresponding to those attributes that cannot coexist with colors. For example, if the terminal uses colors to simulate reverse video (bit number 2 and decimal value 4) and bold (bit number 5 and decimal value 32), the resulting value for \texttt{ncv} will be 36 (4 + 32).
Miscellaneous
If the terminal requires other than a null (zero) character as a pad, then this
can be given as pad. Only the first character of the pad string is used. If
the terminal does not have a pad character, specify npc.

If the terminal can move up or down half a line, this can be indicated with
hu (half-line up) and hd (half-line down). This is primarily useful for
superscripts and subscripts on hardcopy terminals. If a hardcopy terminal
can eject to the next page (form feed), give this as ff (usually control L).

If there is a command to repeat a given character a given number of times
(to save time transmitting a large number of identical characters) this can be
indicated with the parameterized string rep. The first parameter is the char-
acter to be repeated and the second is the number of times to repeat it.
Thus, tparm(repeat_char, 'x', 10) is the same as xxxxxxxxxx.

If the terminal has a settable command character, such as the Tektronix
4025, this can be indicated with cmdch. A prototype command character is
chosen which is used in all capabilities. This character is given in the
cmdch capability to identify it. The following convention is supported on
some UNIX systems: if the environment variable CC exists, all
occurrences of the prototype character are replaced with the character in
CC.

Terminal descriptions that do not represent a specific kind of known termi-
nal, such as switch, dialup, patch, and network, should include the gn
(generic) capability so that programs can complain that they do not know
how to talk to the terminal. (This capability does not apply to virtual termi-
nal descriptions for which the escape sequences are known.) If the terminal
is one of those supported by the UNIX system virtual terminal protocol, the
terminal number can be given as vt. A line-turn-around sequence to be
transmitted before doing reads should be specified in rfi.

If the terminal uses xon/xoff handshaking for flow control, give xon. Pad-
dding information should still be included so that routines can make better
decisions about costs, but actual pad characters will not be transmitted.
Sequences to turn on and off xon/xoff handshaking may be given in smxon
and rmxon. If the characters used for handshaking are not "S and "Q, they
may be specified with xonc and xoffc.

If the terminal has a "meta key" which acts as a shift key, setting the 8th
bit of any character transmitted, this fact can be indicated with km. Other-
wise, software will assume that the 8th bit is parity and it will usually be
cleared. If strings exist to turn this "meta mode" on and off, they can be
given as smm and rmm.
If the terminal has more lines of memory than will fit on the screen at once, the number of lines of memory can be indicated with \texttt{lm}. A value of \texttt{lm}#0 indicates that the number of lines is not fixed, but that there is still more memory than fits on the screen.

Media copy strings which control an auxiliary printer connected to the terminal can be given as \texttt{mc0}: print the contents of the screen, \texttt{mc4}: turn off the printer, and \texttt{mc5}: turn on the printer. When the printer is on, all text sent to the terminal will be sent to the printer. A variation, \texttt{mc5p}, takes one parameter, and leaves the printer on for as many characters as the value of the parameter, then turns the printer off. The parameter should not exceed 255. If the text is not displayed on the terminal screen when the printer is on, specify \texttt{mc5i} (silent printer). All text, including \texttt{mc4}, is transparently passed to the printer while an \texttt{mc5p} is in effect.

Special Cases

The working model used by \texttt{terminfo} fits most terminals reasonably well. However, some terminals do not completely match that model, requiring special support by \texttt{terminfo}. These are not meant to be construed as deficiencies in the terminals; they are just differences between the working model and the actual hardware. They may be unusual devices or, for some reason, do not have all the features of the \texttt{terminfo} model implemented.

Terminals which can not display tilde (\texttt{\^}) characters, such as certain Hazel-tine terminals, should indicate \texttt{hz}.

Terminals which ignore a linefeed immediately after an \texttt{am} wrap, such as the \texttt{Concept} 100, should indicate \texttt{xenl}. Those terminals whose cursor remains on the right-most column until another character has been received, rather than wrapping immediately upon receiving the right-most character, such as the VT100, should also indicate \texttt{xenl}.

If \texttt{el} is required to get rid of standout (instead of writing normal text on top of it), \texttt{xhp} should be given.

Those Teleray terminals whose tabs turn all characters moved over to blanks, should indicate \texttt{xt} (destructive tabs). This capability is also taken to mean that it is not possible to position the cursor on top of a “magic cookie” therefore, to erase standout mode, it is instead necessary to use delete and insert line.

Those Beehive Superbee terminals which do not transmit the escape or control--C characters, should specify \texttt{xsb}, indicating that the \texttt{f1} key is to be used for escape and the \texttt{f2} key for control--C.
Similar Terminals
If there are two very similar terminals, one can be defined as being just like the other with certain exceptions. The string capability use can be given with the name of the similar terminal. The capabilities given before use override those in the terminal type invoked by use. A capability can be canceled by placing xx@ to the left of the capability definition, where xx is the capability. For example, the entry

att4424-2lTeletype 4424 in display function group ii,
rev@, sgr@, smul@, use=att4424,

defines an AT&T 4424 terminal that does not have the rev, sgr, and smul capabilities, and hence cannot do highlighting. This is useful for different modes for a terminal, or for different user preferences. More than one use capability may be given.

FILES
/usr/lib/terminfo/*
compiled terminal description database
/usr/lib/COREterm/*
subset of compiled terminal description database
/usr/lib/tabset/*
tab settings for some terminals, in a format appropriate to be output to the terminal (escape sequences that set margins and tabs)

SEE ALSO
curses(3X), printf(3S), term(5).
captoinfo(1M), infocmp(1M), tic(1M), tty(7) in the System Administrator's Reference Manual.
Chapter 9 of the Programmer's Guide.

WARNING
As described in the "Tabs and Initialization" section above, a terminal's initialization strings, is1, is2, and is3, if defined, must be output before a curses(3X) program is run. An available mechanism for outputting such strings is tput init (see tput(1) and profile(4)).

Tampering with entries in /usr/lib/COREterm/* or /usr/lib/terminfo/* (for example, changing or removing an entry) can affect programs such as vi(1) that expect the entry to be present and correct. In particular, removing the description for the "dumb" terminal will cause unexpected problems.

NOTE
The termcap database (from earlier releases of UNIX System V) may not be supplied in future releases.
NAME
timezone – set default system time zone

SYNOPSIS
./etc/TIMEZONE

DESCRIPTION
This file sets and exports the time zone environmental variable TZ.
This file is read by init(1) after system boot up and all subsequent processes
inherit TZ in their environment.
The syntax of TZ can be described as follows:

```
TZ         → zone
            / zone signed_time
            / zone signed_time zone
            / zone signed_time zone dst
zone       → letter letter letter
signed_time → sign time
            / time
            / hour
            / hour : minute
            / hour : minute : second
            / signed_time
            / signed_time ; dst_date , dst_date
            / ; dst_date , dst_date
           
dst_date   → julian
            / julian / time
letter     → a / A / b / B / ... / z / Z
hour       → 00 / 01 / ... / 23
minute     → 00 / 01 / ... / 59
second     → 00 / 01 / ... / 59
julian     → 001 / 002 / ... / 366
sign       → - / +
```

EXAMPLES
The contents of /etc/TIMEZONE corresponding to the simple example
below could be

```
# Time Zone
TZ=EST5EDT
export TZ
```

A simple setting for New Jersey could be

April 1990 - 1 - Version 5.0
TZ=EST5EDT

where EST is the abbreviation for the main time zone, 5 is the difference, in hours, between GMT (Greenwich Mean Time) and the main time zone, and EDT is the abbreviation for the alternate time zone.

The most complex representation of the same setting, for the year 1986, is

```
TZ="EST5:00:00EDT4:00:00;117/2:00:00,299/2:00:00"
```

where EST is the abbreviation for the main time zone, 5:00:00 is the difference, in hours, minutes, and seconds between GMT and the main time zone, EDT is the abbreviation for the alternate time zone, 4:00:00 is the difference, in hours, minutes, and seconds between GMT and the alternate time zone, 117 is the number of the day of the year (Julian day) when the alternate time zone will take effect, 2:00:00 is the number of hours, minutes, and seconds past midnight when the alternate time zone will take effect, 299 is the number of the day of the year when the alternate time zone will end, and 2:00:00 is the number of hours, minutes, and seconds past midnight when the alternate time zone will end.

A southern hemisphere setting such as the Cook Islands could be

```
TZ="KDT9:30KST10:00;64/5:00,303/20:00"
```

This setting means that KDT is the abbreviation for the main time zone, KST is the abbreviation for the alternate time zone, KDT is 9 hours and 30 minutes later than GMT, KST is 10 hours later than GMT, the starting date of KST is the 64th day at 5 AM, and the ending date of KST is the 303rd day at 8 PM.

Starting and ending times are relative to the alternate time zone. If the alternate time zone start and end dates and the time are not provided, the days for the United States that year will be used and the time will be 2 AM. If the start and end dates are provided but the time is not provided, the time will be midnight.

NOTES

When the longer format is used, the TZ variable must be surrounded by double quotes as shown.

The system administrator must change the Julian start and end days annually if the longer form of the TZ variable is used.

Setting the time during the interval of change from the main time zone to the alternate time zone or vice versa can produce unpredictable results.

SEE ALSO

environ(5).
ctime(3C) in the Programmer's Reference Manual.
NAME
transferdevice – a shell script specification for extending the WorkSpace menu functions

SYNOPSIS
transferdevice menu
transferdevice versionsOK

DESCRIPTION
transferdevices are shell scripts that implement one or more possible menu actions and are recognized by the standard WorkSpace file typing rules. The transfermanager is used to select and customize transfer devices on a per user bases.

Transfer devices must reside in either the directory /etc/transferDevices or $HOME/.workspace/localTransferLinks, and follow a set of conventions to be recognized as such.

The second line of the transfer device (the line after the shell invocation) must read

#transferDevName

where Name may be any addition to the "transferDev" prefix. If it is desired that a special icon be associated with a device, corresponding FTR and ICON rules must be constructed.

There are two command line arguments that all transfer devices must understand. Both menu and versionsOK are used by WorkSpace or the Transfer Manager. In response to a menu argument, a transfer device returns a number of lines to stdout. Each line consists of a text token, a space and string of text. Each text token corresponds to an action that the particular transfer device is designed to implement. The text string is used to describe that action.

In answer to the versionsOK argument, the device is expected to return (to stdout) either or both of the strings “local” and “remote” (seperated by a space). A response of local indicates that the transfer device may be invoked “as is.” A response of “remote” means that if a symbolic link is created from the file transferdevice.machine to transferdevice, any invocation of transferdevice.machine will be understood by the transfer device to mean that the action should be carried out on the remote machine.

FILES
/etc/transferDevice/
~/workspace/localTransferLinks/
SEE ALSO

transfermanager(1G), cpioDevice(1), rcpDevice(1), tarDevice(1), workspace(1G)

*Programming the IRIS WorkSpace*
NAME
tythe – data base of terminal types by port

SYNOPSIS
/etc/ttytype

DESCRIPTION
TTYtype is a database containing, for each tty port on the system, the kind of terminal that is attached to it. There is one line per port, containing the terminal kind (as a name listed in termcap(4)), a space, and the name of the tty, minus /dev/.

This information is read by tset(1) and by login(1) to initialize the TERM environment variable at login time.

EXAMPLE

iris-ansi console
iris-ansi systty
vt100 ttyd1
?h19 ttyd2
?h19 ttyd3
?v50am ttyd4
?v50am ttyd5
?v50am ttyd6
?v50am ttyd7
?v50am ttyd8
?v50am ttyd9
?v50am ttyd10
?v50am ttyd11
?v50am ttyd12

FILES
/etc/ttytype

SEE ALSO
tset(1), login(1).
NAME
unistd – file header for symbolic constants

SYNOPSIS
#include <unistd.h>

DESCRIPTION
The header file <unistd.h> lists the symbolic constants and structures not
already defined or declared in some other header file.

/* Symbolic constants for the "access" routine: */

#define R_OK 4 /*Test for Read permission */
#define W_OK 2 /*Test for Write permission */
#define X_OK 1 /*Test for eXecute permission */
#define F_OK 0 /*Test for existence of File */

#define F_ULOCK 0 /*Unlock a previously locked region */
#define F_LOCK 1 /*Lock a region for exclusive use */
#define F_TLOCK 2 /*Test and lock a region for exclusive use */
#define F_TEST 3 /*Test a region for other processes locks */

/*Symbolic constants for the "lseek" routine: */

#define SEEK_SET 0 /* Set file pointer to "offset" */
#define SEEK_CUR 1 /* Set file pointer to current plus "offset" */
#define SEEK_END 2 /* Set file pointer to EOF plus "offset" */

/*Pathnames:*/

#define GF_PATH /etc/group /*Pathname of the group file */
#define PF_PATH /etc/passwd /*Pathname of the passwd file */
NAME
utmp, wtmp – utmp and wtmp entry formats

SYNOPSIS
#include <sys/types.h>
#include <utmp.h>

DESCRIPTION
These files, which hold user and accounting information for such commands as who(1), write(1), and login(1), have the following structure as defined by <utmp.h>:

#define UTMP_FILE "/etc/utmp"
#define WTMP_FILE "/etc/wtmp"
#define ut_name ut_user

struct utmp {
  char  ut_user[8];  /* User login name */
  char  ut_id[4];  /* /etc/instab id (usually line #) */
  char  ut_line[12];  /* device name (console, Inxx) */
  short ut_pid;  /* process id */
  short ut_type;  /* type of entry */
  struct exit_status {
    short e_termination;  /* Process termination status */
    short e_exit;  /* Process exit status */
  } ut_exit;  /* The exit status of a process
  * marked as DEAD_PROCESS. */
  time_t ut_time;  /* time entry was made */
};
/* Definitions for ut_type */
#define EMPTY 0
#define RUN_LVL 1
#define BOOT_TIME 2
#define OLD_TIME 3
#define NEW_TIME 4
#define INIT_PROCESS 5 /* Process spawned by "init" */
#define LOGIN_PROCESS 6 /* A "getty" process waiting for login */
#define USER_PROCESS 7 /* A user process */
#define DEAD_PROCESS 8
#define ACCOUNTING 9
#define UTMAXTYPE ACCOUNTING /* Largest legal value of ut_type */

/*/ Special strings or formats used in the "ut_line" field when */
/*/ accounting for something other than a process */
/*/ No string for the ut_line field can be more than 11 chars + */
/*/ a NULL in length */
#define RUNLVL_MSG "run-level %c"
#define BOOT_MSG "system boot"
#define OTIME_MSG "old time"
#define NTIME_MSG "new time"

FILES
/etc/utmp
/etc/wtmp

SEE ALSO
getut(3C).
NAME
uuencode – format of an encoded uuencode file

DESCRIPTION
Files output by uuencode(1C) consist of a header line, followed by a number of body lines, and a trailer line. Uudecode(1C) will ignore any lines preceding the header or following the trailer. Lines preceding a header must not, of course, look like a header.

The header line is distinguished by having the first 6 characters "begin ".
The word begin is followed by a mode (in octal), and a string which names the remote file. A space separates the three items in the header line.

The body consists of a number of lines, each at most 62 characters long (including the trailing newline). These consist of a character count, followed by encoded characters, followed by a newline. The character count is a single printing character, and represents an integer, the number of bytes the rest of the line represents. Such integers are always in the range from 0 to 63 and can be determined by subtracting the character space (octal 40) from the character.

Groups of 3 bytes are stored in 4 characters, 6 bits per character. All are offset by a space to make the characters printing. The last line may be shorter than the normal 45 bytes. If the size is not a multiple of 3, this fact can be determined by the value of the count on the last line. Extra garbage will be included to make the character count a multiple of 4. The body is terminated by a line with a count of zero. This line consists of one ASCII space.

The trailer line consists of "end" on a line by itself.

SEE ALSO
uuencode(1C), uudecode(1C), uucp(1C), mail(1)
NAME
visuallogin, noiconlogin – select and control console login program

DESCRIPTION
The configuration flag visuallogin selects the type of login program used for
the graphics console. If visuallogin is on, the visual login program pandora(1) used for logins. If it is off, the standard IRIX login(1) program is
used.

If the configuration flag noiconlogin is on, pandora(1) displays icons for
each user. If it is off, icons are not displayed.

The value of the flags can be set to on or off using chkconfig(1M).

FILES
/etc/config/visuallogin
/etc/config/noiconlogin

SEE ALSO
login(1), pandora(1), chkconfig(1M)
NAME
intro – introduction to miscellany

DESCRIPTION
This section describes miscellaneous facilities such as macro packages, character set tables, etc.
NAME
ascii – map of ASCII character set

DESCRIPTION

ascii is a map of the ASCII character set, giving both octal and hexadecimal equivalents of each character, to be printed as needed. It contains:

```
1000 null 01 soh 02 stx 03 etx 04 eot 05 enq 06 ack 07 bell 08 bs 09 ht 10 lf 11 cr 12 so 13 si 14 dle 15 dc1 16 dc2 17 dc3 18 dc4 19 nak 20 syn 21 etb 22 can 23 em 24 h Ack 25 sub 26 sc 27 del
```

```
```

April 1990 - 1 - Version 4.0
NAME
charset – description of the standard supported character set

DESCRIPTION
A single 8-bit character set, based on ISO 8859-1, is currently supported. Other character sets may be supported in the future.

ISO 8859-1 is an 8-bit single-byte coded character set. This set, Latin Alphabet #1, contains characters used for general purpose applications in typical office environments in at least the following languages: Danish, Dutch, English, Faeroese, Finnish, French, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Spanish, and Swedish.

(NOTE: please see this man page in the IRIS-4D Programmer's Reference Manual for the table of characters.)

The ASCII 7-bit character set is a proper subset of 8859-1 (see ASCII(5)). The characters added by ISO 8859-1 each have the most significant bit of the byte on.
NAME

environ – user environment

DESCRIPTION

An array of strings called the “environment” is made available by exec(2) when a process begins. By convention, these strings have the form “name=value”. The following names are used by various commands:

CFTIME  The default format string to be used by the date(1) command and the asctime() and cftime() routines (see ctime(3C)). If CFTIME is not set or is null, the default format string specified in the /lib/cftime/LANGUAGE file (if it exists) is used in its place (see cftime(4)).

CHRCLASS  A value that corresponds to a file in /lib/chrclass containing character classification and conversion information. This information is used by commands (such as cat(1), ed(1), sort(1), etc.) to classify characters as alphabetic, printable, upper case, etc. and to convert characters to upper or lower case.

When a program or command begins execution, the tables containing this information are initialized based on the value of CHRCLASS. If CHRCLASS is non-existent, null, set to a value for which no file exists in /lib/chrclass, or errors occur while reading the file, the ASCII character set is used. During execution, a program or command can change the values in these tables by calling the setchrclass() routine. For more detail, see ctype(3C).

These tables are created using the chrribl(1M) command.

HOME  The name of the user’s login directory, set by login(1) from the password file (see passwd(4)).

LANGUAGE  A language for which a printable file by that name exists in /lib/cftime. This information is used by commands (such as date(1), ls(1), sort(1), etc.) to print date and time information in the language specified.

If LANGUAGE is non-existent, null, set to a value for which no file exists in /lib/cftime, or errors occur while reading the file, the last language requested will be used. (If no language has been requested, the language usa_english is assumed.) For a description of the content of files in /lib/cftime, see cftime(4).
PATH
The sequence of directory prefixes that `sh(1)`, `time(1)`, `nice(1)`, `nohup(1)`, etc., apply in searching for a file known by an incomplete path name. The prefixes are separated by colons (:). `login(1)` sets `PATH=::/usr/sbin:/usr/bsd:/bin:/usr/bin:/usr/bin/X11.` (For more detail, see the "Execution" section of the `sh(1)` manual page.)

TERM
The kind of terminal for which output is to be prepared. This information is used by commands, such as `more(1)` or `vi(1)`, which may exploit special capabilities of that terminal.

USER
The user name (from the password file) of the user which is currently running. It is set by `login(1)`, `su(1)`, `rlogind(1M)`, `newgrp(1)`, and `cron(1)`.

LOGNAME
Synonymous with USER.

SHELL
The path to the user's shell, set from the password file.

TZ
Time zone information. The simplest format is `xxxxnzzz` where `xxx` is the standard local time zone abbreviation, `n` is the difference in hours from GMT (Greenwich Mean Time), and `zzz` is the abbreviation for an alternate time zone (usually the daylight-saving local time zone), if any; for example, `TZ="EST5EDT"`

The most complex format allows you to specify the difference in hours of the alternate time zone from GMT and the starting day and time and ending day and time for using this alternate time zone. For example, in 1985 the complex format corresponding to the above simple example is:

`TZ="EST5:00:00EDT4:00:00;118/2:00:00,300/2:00:00"`

When the above complex format is used, it must be surrounded by double quotes. For more details, see `ctime(3C)` and `timezone(4)`.

Further names may be placed in the environment by the `export` command and "name=value" arguments in `sh(1)`, by the `setenv` command in `csh(1)`, or by `exec(2)`. It is unwise to conflict with certain shell variables that are frequently exported by `.profile` or `.cshrc` files: MAIL, PS1, PS2, IFS (see `profile(4)` and `cshrc(4)`).

NOTES
Administrators should note the following: if you attempt to set the current date to one of the dates that the standard and alternate time zones change (for example, the date that daylight time is starting or ending), and you
attempt to set the time to a time in the interval between the end of standard
time and the beginning of the alternate time (or the end of the alternate time
and the beginning of standard time), the results are unpredictable.

SEE ALSO

chrtbl(1M), cftime(4), passwd(4), profile(4), cshrc(4), timezone(4), in the
exec(2), ctime(3C), ctype(3C) in the Programmer's Reference Manual.
cat(1), date(1), ed(1), env(1), ls(1), login(1), nice(1), nohup(1), sh(1),
NAME
fcntl – file control options

SYNOPSIS
#include <fcntl.h>

DESCRIPTION
The fcntl(2) function provides for control over open files. This include file describes requests and arguments to fcntl and open(2).

/ * fcntl(2) requests */
#define F_DUPFD 0 /* Duplicate filedes */
#define F_GETFD 1 /* Get filedes flags */
#define F_SETFD 2 /* Set filedes flags */
#define F_GETFL 3 /* Get file flags */
#define F_SETFL 4 /* Set file flags */
#define F_GETLK 5 /* Get file lock */
#define F_SETLK 6 /* Set file lock */
#define F_SETLKW 7 /* Set file lock and wait */
#define F_CHKFL 8 /* Check legality of file flag changes */

/ * The following apply to sockets only */
#define F_GETOWN 10 /* Get pid receiving SIGIO, SIGURG */
#define F_SETOWN 11 /* Set pid to receive SIGIO, SIGURG */

/ * Flags for F_GETFL and F_SETFL fcntl(2) requests */
#define FNDELAY 0x04 /* Non-blocking I/O */
#define FAPPEND 0x08 /* append (writes guaranteed at the end) */
#define FSYNC 0x10 /* synchronous write option */
#define FRCACHE 0x20 /* Used for file and record locking cache */
#define FASTSYNC 0x40 /* interrupt-driven I/O for sockets */
#define FNONBLK 0x80 /* POSIX Non-blocking I/O */

/ * open-only modes */
#define FCREAT 0x100 /* create if nonexistent */
#define F_TRUNC 0x200 /* truncate to zero length */
#define FEXCL 0x400 /* error if already created */
#define FNOCTTY 0x800 /* POSIX: don’t make this controlling tty */

/ * Flag values accessible to open(2) and fcntl(2) */
#define O_RDONLY 0
#define O_WRONLY 1
#define O_RDWR 2
#define O_ACCMODE 0x3 /* mask for above access bits */
#define O_NDELAY FNDELAY /* Non-blocking I/O */

April 1990

- 1 -

Version 5.0
/* Flag values accessible only to open(2) */
#define O_CREAT  FCREATE  /* open w/ create (uses 3rd open arg) */
#define O_TRUNC  FTRUNC   /* open w/ truncation */
#define O_EXCL   FEXCL     /* exclusive open */
#define O_NOCTTY FNOCTTY   /* don't assign as controlling tty */

/* file segment locking control structure */
struct flock {
    short  l_type;
    short  l_whence;
    long   l_start;
    long   l_len;       /* if 0 then until EOF */
    short  l_sysid;
    short  l_pid;       /* returned with F_GETLK */
}

/* file segment locking types */
#define F_RDLCK  01  /* Read lock */
#define F_WRLCK  02  /* Write lock */
#define F_UNLCK  03  /* Remove locks */
#define FD_CLOEXEC 0x1  /* fcntl 1 in lo bit of arg param */

SEE ALSO
fcntl(2), open(2).
4.3BSD (for socket-related options)
NAME
hostname - host name resolution description

DESCRIPTION
Hostnames are domains, where a domain is a hierarchical, dot-separated list of subdomains; for example, the machine monet, in the Berkeley subdomain of the EDU subdomain of the Internet would be represented as

monet.Berkeley.EDU
(with no trailing dot).

Hostnames are often used with network client and server programs, which must generally translate the name to an address for use. (This function is generally performed by the library routine gethostbyname(3N).) Hostnames are resolved by the Internet name resolver in the following fashion.

If the name consists of a single component, i.e., contains no dot, and if the environment variable "HOSTALIASES" is set to the name of a file, that file is searched for a string matching the input hostname. The file should consist of lines made up of two white-space separated strings, the first of which is the hostname alias, and the second of which is the complete hostname to be substituted for that alias. For example, to refer to the host "matisse.painters.org" with the alias "henri", use

henri matisse.painters.org

If a case-insensitive match is found between the hostname to be resolved and the first field of a line in the file, the substituted name is looked up with no further processing.

If the input name ends with a trailing dot, the trailing dot is removed, and the remaining name is looked up with no further processing.

If the input name does not end with a trailing dot, it is looked up by searching through a list of domains until a match is found. The default search list includes first the local domain, then its parent domains with at least 2 name components (longest first). For example, in the domain CS.Berkeley.EDU, the name lithium.CChem will be checked first as lithium.CChem.CS.Berkeley.EDU and then as lithium.CChem.Berkeley.EDU. Lithium.CChem.EDU will not be tried, as there is only one component remaining from the local domain. The search path can be changed from the default by the resolv.conf system-wide configuration file. See the descriptions of the search keyword in resolver(4).

SEE ALSO
named(1M), gethostbyname(3N), resolver(3N), hosts(4), resolver(4)
NAME
math – math functions and constants

SYNOPSIS
#include <math.h>

DESCRIPTION
This file contains declarations of all the functions in the Math Library (described in Section 3M), as well as various functions in the C Library (Section 3C) that return floating-point values.

It defines the following constant used as an error-return value:

HUGE The maximum value of a single-precision floating-point number

The following mathematical constants are defined for user convenience:

M_E The base of natural logarithms (e)
M_LOG2E The base-2 logarithm of e
M_LOG10E The base-10 logarithm of e
M_LN2 The natural logarithm of 2
M_LN10 The natural logarithm of 10
M_PI Pi (the ratio of the circumference of a circle to its diameter)

M_PI_2 \( \pi/2 \)
M_PI_4 \( \pi/4 \)
M_1_PI \( 1/\pi \)
M_2_PI \( 2/\pi \)
M_2_SQRTPI \( 2/\sqrt{\pi} \)
M_SQRT2 The positive square root of 2
M_SQRT1_2 The positive square root of 1/2

For the definitions of various machine-dependent "constants," see the description of the <values.h> header file.

SEE ALSO
intro(3), values(5).
NAME
regexp – regular expression compile and match routines

SYNOPSIS
#define INIT <declarations>
#define GETC() <getc code>
#define PEEKC() <peekc code>
#define UNGETC(c) <ungetc code>
#define RETURN(pointer) <return code>
#define ERROR(val) <error code>

#include <regexp.h>
char *compile (instring, expbuf, endbuf, eof)
char *instring, *expbuf, *endbuf;
int eof;

int step (string, expbuf)
char *string, *expbuf;
extern char *loc1, *loc2, *locs;
extern int ciref, sed, nbra;

DESCRIPTION
This page describes general-purpose regular expression matching routines in the form of ed(1), defined in <regexp.h>. Programs such as ed(1), sed(1), grep(1), expr(1), etc., which perform regular expression matching use this source file. In this way, only this file need be changed to maintain regular expression compatibility.

The interface to this file is un pleasingly complex. Programs that include this file must have the following five macros declared before the ‘‘#include <regexp.h>’’ statement. These macros are used by the compile routine.

GETC() Return the value of the next character in the regular expression pattern. Successive calls to GETC() should return successive characters of the regular expression.

PEEKC() Return the next character in the regular expression. Successive calls to PEEKC() should return the same character [which should also be the next character returned by GETC()].

UNGETC(c) Cause the argument c to be returned by the next call to GETC() [and PEEKC()]. No more that one character of pushback is ever needed and this character is guaranteed to be the last character read by
GETC(). The value of the macro UNGETC(c) is always ignored.

RETURN(pointer) This macro is used on normal exit of the compile routine. The value of the argument pointer is a pointer to the character after the last character of the compiled regular expression. This is useful to programs which have memory allocation to manage.

ERROR(val) This is the abnormal return from the compile routine. The argument val is an error number (see table below for meanings). This call should never return.

<table>
<thead>
<tr>
<th>ERROR</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Range endpoint too large.</td>
</tr>
<tr>
<td>16</td>
<td>Bad number.</td>
</tr>
<tr>
<td>25</td>
<td>&quot;\digit&quot; out of range.</td>
</tr>
<tr>
<td>36</td>
<td>Illegal or missing delimiter.</td>
</tr>
<tr>
<td>41</td>
<td>No remembered search string.</td>
</tr>
<tr>
<td>42</td>
<td>( ) imbalance.</td>
</tr>
<tr>
<td>43</td>
<td>Too many (.</td>
</tr>
<tr>
<td>44</td>
<td>More than 2 numbers given in { }.</td>
</tr>
<tr>
<td>45</td>
<td>} expected after .</td>
</tr>
<tr>
<td>46</td>
<td>First number exceeds second in { }.</td>
</tr>
<tr>
<td>49</td>
<td>[ ] imbalance.</td>
</tr>
<tr>
<td>50</td>
<td>Regular expression overflow.</td>
</tr>
</tbody>
</table>

The syntax of the compile routine is as follows:

\texttt{compile(instring, expbuf, endbuf, eof)}

The first parameter \texttt{instring} is never used explicitly by the compile routine but is useful for programs that pass down different pointers to input characters. It is sometimes used in the INIT declaration (see below). Programs which call functions to input characters or have characters in an external array can pass down a value of ((char *) 0) for this parameter.

The next parameter \texttt{expbuf} is a character pointer. It points to the place where the compiled regular expression will be placed.

The parameter \texttt{endbuf} is one more than the highest address where the compiled regular expression may be placed. If the compiled expression cannot fit in \texttt{(endbuf–expbuf)} bytes, a call to ERROR(50) is made.
The parameter *eof* is the character which marks the end of the regular expression. For example, in *ed(1)*, this character is usually a `/`

Each program that includes this file must have a `#define` statement for *INIT*. This definition will be placed right after the declaration for the function *compile* and the opening curly brace `{`. It is used for dependent declarations and initializations. Most often it is used to set a register variable to point the beginning of the regular expression so that this register variable can be used in the declarations for *GETC()* and *UNGETC()* otherwise it can be used to declare external variables that might be used by *GETC()* and *UNGETC()* and *PEEKCS()*.* See the example below of the declarations taken from *grep(1)*.

There are other functions in this file which perform actual regular expression matching, one of which is the function *step*. The call to *step* is as follows:

```
step(string, expbuf)
```

The first parameter to *step* is a pointer to a string of characters to be checked for a match. This string should be null terminated.

The second parameter *expbuf* is the compiled regular expression which was obtained by a call of the function *compile*.

The function *step* returns non-zero if the given string matches the regular expression, and zero if the expressions do not match. If there is a match, two external character pointers are set as a side effect to the call to *step*. The variable set in *step* is *loc1*. This is a pointer to the first character that matched the regular expression. The variable *loc2*, which is set by the function *advance*, points to the character after the last character that matches the regular expression. Thus if the regular expression matches the entire line, *loc1* will point to the first character of *string* and *loc2* will point to the null at the end of *string*.

*Step* uses the external variable *circf* which is set by *compile* if the regular expression begins with `\^`. If this is set then *step* will try to match the regular expression to the beginning of the string only. If more than one regular expression is to be compiled before the first is executed the value of *circf* should be saved for each compiled expression and *circf* should be set to that saved value before each call to *step*.

The function *advance* is called from *step* with the same arguments as *step*. The purpose of *step* is to step through the *string* argument and call *advance* until *advance* returns non-zero indicating a match or until the end of *string* is reached. If one wants to constrain *string* to the beginning of the line in all cases, *step* need not be called; simply call *advance*. 
When *advance* encounters a * or \{ \} sequence in the regular expression, it will advance its pointer to the string to be matched as far as possible and will recursively call itself trying to match the rest of the string to the rest of the regular expression. As long as there is no match, *advance* will back up along the string until it finds a match or reaches the point in the string that initially matched the * or \{ \}. It is sometimes desirable to stop this backing up before the initial point in the string is reached. If the external character pointer *locps* is equal to the point in the string at sometime during the backing up process, *advance* will break out of the loop that backs up and will return zero. This is used by *ed(1)* and *sed(1)* for substitutions done globally (not just the first occurrence, but the whole line) so, for example, expressions like s/y*///g do not loop forever.

The additional external variables *sed* and *nbra* are used for special purposes.

EXAMPLES
The following is an example of how the regular expression macros and calls look from *grep(1)*:

```
#define INIT register char *sp = instring;
#define GETC() (*sp++)
#define PEEKC() (*sp)
#define UNGETC(c) (--sp)
#define RETURN(c) return;
#define ERROR(c) regerr()

#include <regexp.h>
...
(void) compile(*argv, expbuf; &expbuf[ESIZE], 'O');
...
if (step(linebuf, expbuf))
    succeed();
```

SEE ALSO
*ed(1)*, *expr(1)*, *grep(1)*, *sed(1)* in the *User's Reference Manual*. 
NAME
stat – data returned by stat system call

SYNOPSIS
#include <sys/types.h>
#include <sys/stat.h>

DESCRIPTION
The system calls stat and fstat return data whose structure is defined by this
include file. The encoding of the field st_mode is defined in this file also.

Structure of the result of stat

struct stat
{
    ino_t     st_ino;
    dev_t     st_dev;
    mode_t    st_mode;
    short     st_nlink;
    ushort    st_uid;
    ushort    st_gid;
    dev_t     st_rdev;
    off_t     st_size;
    time_t    st_atime;
    time_t    st_mtime;
    time_t    st_ctime;
};

#define S_IFMT  0170000 /* type of file */
#define S_IFDIR 0040000 /* directory */
#define S_IFCHR 0020000 /* character special */
#define S_IFBLK 0060000 /* block special */
#define S_IFREG 0100000 /* regular */
#define S_IFIFO 0010000 /* fifo */
#define S_IFLNK 0120000 /* symbolic link */
#define S_ISUID  04000 /* set user id on execution */
#define S_ISGID  02000 /* set group id on execution */
#define S_ISVTX  01000 /* directory permissions control */
#define S_IREAD  004000 /* read permission, owner */
#define S_IWRITE  002000 /* write permission, owner */
#define S_IEXEC  00100 /* execute/search permission, owner */
#define S_ENFMT S_ISGID /* record locking enforcement flag */
#define S_IRWXU  007000 /* read,write, execute: owner */
#define S_IRUSR 00400 /* read permission: owner */
#define S_IWUSR 00200 /* write permission: owner */
#define S_IXUSR 00100 /* execute permission: owner */
#define S_IRWXG 00070 /* read, write, execute: group */
#define S_IRGRP 00040 /* read permission: group */
#define S_IWGRP 00020 /* write permission: group */
#define S_IXGRP 00010 /* execute permission: group */
#define S_IRWXO 00007 /* read, write, execute: other */
#define S_IROTH 00004 /* read permission: other */
#define S_IWOTH 00002 /* write permission: other */
#define S_IXOTH 00001 /* execute permission: other */

SEE ALSO
stat(2), types(5).
NAME
stdarg – variable argument list

SYNOPSIS
#include <stdarg.h>
void va_start (va_list ap, ParmN);
type va_arg (va_list ap, type);
void va_end (va_list ap);

DESCRIPTION
This set of macros provides a means of writing portable procedures that accept variable argument lists. Routines having variable argument lists (such as printf(3)) that do not use stdarg are inherently nonportable, since different machines use different argument passing conventions. The stdarg facility is similar to varargs(5), but is based on the ANSI Standard for C.

A variable argument list contains one or more parameters. The rightmost parameter plays a special role, and is designated ParmN in this discussion.

va_list is a type suitable for storing information needed by the macros va_start, va_arg, and va_end. The called function must declare a variable (referred to as ap) of type va_list, used to access the argument list.

The va_start (ap, ParmN) macro initializes ap for subsequent use by va_arg and va_end. va_start must be called before any use of va_arg.

The va_arg (ap, type) macro will return the next argument in the list pointed to by ap. The first invocation of va_arg returns the value of the argument after that specified by ParmN. Successive invocations return the values of the remaining arguments in succession. type is the type to which the expected argument will be converted when passed as an argument. In standard C, arguments that are char or short should be accessed as int, unsigned char or unsigned short are converted to unsigned int, and float arguments are converted to double. Different types can be mixed, but it is up to the routine to know what type of argument is expected.

va_end (ap) is used to finish up.

Multiple traversals, each bracketed by va_start ... va_end, are possible.

EXAMPLE
#include <stdarg.h>
#define MAXARGS 31
void f1(int nptrs, ...)
{
    va_list ap;
    char *array[MAXARGS];
int ptr_no = 0;
if (nptrs > MAXARGS)
    nptrs = MAXARGS;
va_start(ap, nptrs);
while (ptr_no < nptrs)
    (array[ptr_no++] = va_arg(ap, char *));
va_end(ap);
}

SEE ALSO
varargs(5).

BUGS
Due to the procedure calling convention on the MIPS processor, floating-
point parameters may be inaccessible via stdarg unless they appear after a
parameter of non-floating-point type. Thus, in the code sequence
extern int foo(float, ...);
foo(1.0, 2.0);
the parameter 2.0 may be accessed incorrectly. If the function expected an
intervening non-floating-point parameter, such as
extern int foo(float, ...);
foo(1.0, 4, 2.0);
the second floating-point parameter would be accessible as a double. No
problem is encountered, of course, if the type of the first argument is not
floating-point.

Stdarg cannot be used when passing structures as parameters, as it is impos-
sible to determine their alignment at runtime.

It is up to the calling routine to determine how many arguments there are,
since it is not possible to determine this from the stack frame. For example,
execl passes a 0 to signal the end of the list. Printf can tell how many argu-
ments are supposed to be there by the format.

The macros va_start and va_end may be arbitrarily complex; for example,
va_start might contain an opening brace, which is closed by a matching
brace in va_end. Thus, they should only be used where they could be
placed within a single complex statement.
NAME
term – conventional names for terminals

DESCRIPTION
These names are used by certain commands (e.g., man(1), tabs(1), tput(1),
vi(1) and curses(3X)) and are maintained as part of the shell environment
in the environment variable TERM (see sh(1), profile(4), and environ(5)).

Entries in terminfo(4) source files consist of a number of comma-separated
fields. (To obtain the source description for a terminal, use the –I option of
infocmp(1M).) White space after each comma is ignored. The first line of
each terminal description in the terminfo(4) database gives the names by
which terminfo(4) knows the terminal, separated by bar ( | ) characters.
The first name given is the most common abbreviation for the terminal (this
is the one to use to set the environment variable TERMINFO in
$HOME/.profile; see profile(4)), the last name given should be a long name
fully identifying the terminal, and all others are understood as synonyms for
the terminal name. All names but the last should contain no blanks and
must be unique in the first 14 characters; the last name may contain blanks
for readability.

Terminal names (except for the last, verbose entry) should be chosen using
the following conventions. The particular piece of hardware making up the
terminal should have a root name chosen, for example, for the AT&T 4425
terminal, att4425. This name should not contain hyphens, except that
synonyms may be chosen that do not conflict with other names. Up to 8
characters, chosen from [a–z0–9], make up a basic terminal name. Names
should generally be based on original vendors, rather than local distributors.
A terminal acquired from one vendor should not have more than one dis-
tinct basic name. Terminal sub-models, operational modes that the
hardware can be in, or user preferences, should be indicated by appending a
hyphen and an indicator of the mode. Thus, an AT&T 4425 terminal in 132
column mode would be att4425–w. The following suffixes should be used
where possible:

<table>
<thead>
<tr>
<th>Suffix</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>–w</td>
<td>Wide mode (more than 80 columns)</td>
<td>att4425–w</td>
</tr>
<tr>
<td>–am</td>
<td>With auto. margins (usually default)</td>
<td>vt100–am</td>
</tr>
<tr>
<td>–nam</td>
<td>Without automatic margins</td>
<td>vt100–nam</td>
</tr>
<tr>
<td>–n</td>
<td>Number of lines on the screen</td>
<td>aaa–60</td>
</tr>
<tr>
<td>–na</td>
<td>No arrow keys (leave them in local)</td>
<td>c100–na</td>
</tr>
<tr>
<td>–np</td>
<td>Number of pages of memory</td>
<td>c100–4p</td>
</tr>
<tr>
<td>–rv</td>
<td>Reverse video</td>
<td>att4415–rv</td>
</tr>
</tbody>
</table>
To avoid conflicts with the naming conventions used in describing the different modes of a terminal (e.g., \(-w\)), it is recommended that a terminal’s root name not contain hyphens. Further, it is good practice to make all terminal names used in the `terminfo(4)` database unique. Terminal entries that are present only for inclusion in other entries via the `use=` facilities should have a \('+'\) in their name, as in `4415+nl`.

Some of the known terminal names may include the following (for a complete list, type: `ls -C /usr/lib/terminfo/?`):

- `2621,hp2621`: Hewlett-Packard 2621 series
- `2631`: Hewlett-Packard 2631 line printer
- `2631-c`: Hewlett-Packard 2631 line printer - compressed mode
- `2631-e`: Hewlett-Packard 2631 line printer - expanded mode
- `2640,hp2640`: Hewlett-Packard 2640 series
- `2645,hp2645`: Hewlett-Packard 2645 series
- `3270`: IBM Model 3270
- `33,tty33`: AT&T Teletype Model 33 KSR
- `35,tty35`: AT&T Teletype Model 35 KSR
- `37,tty37`: AT&T Teletype Model 37 KSR
- `4000a`: Trendata 4000a
- `4014,tck4014`: TEKTRONIX 4014
- `40,tty40`: AT&T Teletype Dataspeed 40/2
- `43,tty43`: AT&T Teletype Model 43 KSR
- `4410,5410`: AT&T 4410/5410 terminal in 80-column mode - version 2
- `4410-nfk,5410-nfk`: AT&T 4410/5410 without function keys - version 1
- `4410-nsl,5410-nsl`: AT&T 4410/5410 without pln defined
- `4410-w,5410-w`: AT&T 4410/5410 in 132-column mode
- `4410v1,5410v1`: AT&T 4410/5410 terminal in 80-column mode - version 1
- `4410v1-w,5410v1-w`: AT&T 4410/5410 terminal in 132-column mode - version 1
- `4415,5420`: AT&T 4415/5420 in 80-column mode
- `4415-nl,5420-nl`: AT&T 4415/5420 without changing labels
- `4415-rv,5420-rv`: AT&T 4415/5420 80 columns in reverse video
- `4415-rv-nl,5420-rv-nl`: AT&T 4415/5420 reverse video without changing labels
- `4415-w,5420-w`: AT&T 4415/5420 in 132-column mode
- `4415-w-nl,5420-w-nl`: AT&T 4415/5420 in 132-column mode
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4415–w–rv,5420–w–rv</td>
<td>AT&amp;T 4415/5420 132 columns in reverse video without changing labels</td>
</tr>
<tr>
<td>4415–w–rv–nl,5420–w–rv–nl</td>
<td>AT&amp;T 4415/5420 132 columns reverse video without changing labels</td>
</tr>
<tr>
<td>4418,5418</td>
<td>AT&amp;T 5418 in 80-column mode</td>
</tr>
<tr>
<td>4418–w,5418–w</td>
<td>AT&amp;T 5418 in 132-column mode</td>
</tr>
<tr>
<td>4420</td>
<td>AT&amp;T Teletype Model 4420</td>
</tr>
<tr>
<td>4424</td>
<td>AT&amp;T Teletype Model 4424</td>
</tr>
<tr>
<td>4424–2</td>
<td>AT&amp;T Teletype Model 4424 in display function group ii</td>
</tr>
<tr>
<td>4425,5425</td>
<td>AT&amp;T 4425/5425</td>
</tr>
<tr>
<td>4425–fk,5425–fk</td>
<td>AT&amp;T 4425/5425 without function keys</td>
</tr>
<tr>
<td>4425–nl,5425–nl</td>
<td>AT&amp;T 4425/5425 without changing labels in 80-column mode</td>
</tr>
<tr>
<td>4425–w,5425–w</td>
<td>AT&amp;T 4425/5425 in 132-column mode</td>
</tr>
<tr>
<td>4425–w–fk,5425–w–fk</td>
<td>AT&amp;T 4425/5425 without function keys in 132-column mode</td>
</tr>
<tr>
<td>4425–nl–w,5425–nl–w</td>
<td>AT&amp;T 4425/5425 without changing labels in 132-column mode</td>
</tr>
<tr>
<td>4426</td>
<td>AT&amp;T Teletype Model 4426S</td>
</tr>
<tr>
<td>450</td>
<td>DASI 450 (same as Diablo 1620)</td>
</tr>
<tr>
<td>450–12</td>
<td>DASI 450 in 12-pitch mode</td>
</tr>
<tr>
<td>500,att500</td>
<td>AT&amp;T-IS 500 terminal</td>
</tr>
<tr>
<td>510,510a</td>
<td>AT&amp;T 510/510a in 80-column mode</td>
</tr>
<tr>
<td>513bct,att513</td>
<td>AT&amp;T 513 bct terminal</td>
</tr>
<tr>
<td>5320</td>
<td>AT&amp;T 5320 hardcopy terminal</td>
</tr>
<tr>
<td>5420_2</td>
<td>AT&amp;T 5420 model 2 in 80-column mode</td>
</tr>
<tr>
<td>5420_2–w</td>
<td>AT&amp;T 5420 model 2 in 132-column mode</td>
</tr>
<tr>
<td>5620,dmd</td>
<td>AT&amp;T 5620 terminal 88 columns</td>
</tr>
<tr>
<td>5620–24,dmd–24</td>
<td>AT&amp;T Teletype Model DMD 5620 in a 24x80 layout</td>
</tr>
<tr>
<td>5620–34,dmd–34</td>
<td>AT&amp;T Teletype Model DMD 5620 in a 34x80 layout</td>
</tr>
<tr>
<td>610,610bct</td>
<td>AT&amp;T 610 bct terminal 80-column mode</td>
</tr>
<tr>
<td>610–w,610bct–w</td>
<td>AT&amp;T 610 bct terminal 132-column mode</td>
</tr>
<tr>
<td>7300,pc7300,unix_pc</td>
<td>AT&amp;T UNIX PC Model 7300</td>
</tr>
<tr>
<td>735,ti</td>
<td>Texas Instruments TI735 and TI725</td>
</tr>
<tr>
<td>745</td>
<td>Texas Instruments TI745</td>
</tr>
<tr>
<td>dumb</td>
<td>generic name for terminals that lack reverse line-feed and other special escape sequences</td>
</tr>
<tr>
<td>hp</td>
<td>Hewlett-Packard (same as 2645)</td>
</tr>
<tr>
<td>iris–ansi</td>
<td>SGI wsh(1) ANSI emulator (40 lines)</td>
</tr>
<tr>
<td>iris–ansi–24</td>
<td>SGI wsh(1) ANSI emulator (24 lines)</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>iris-ansi-66</td>
<td>SGI wsh(1) ANSI emulator (66 lines)</td>
</tr>
<tr>
<td>iris-ansi-net</td>
<td>SGI remote login from wsh(1) window</td>
</tr>
<tr>
<td>lp</td>
<td>generic name for a line printer</td>
</tr>
<tr>
<td>pt505</td>
<td>AT&amp;T Personal Terminal 505 (22 lines)</td>
</tr>
<tr>
<td>pt505-24</td>
<td>AT&amp;T Personal Terminal 505 (24-line mode)</td>
</tr>
<tr>
<td>rwsiris</td>
<td>SGI remote login from visual 50 emulator</td>
</tr>
<tr>
<td>sync</td>
<td>generic name for synchronous Teletype Model 4540-compatible terminals</td>
</tr>
<tr>
<td>wsiris</td>
<td>SGI visual 50 terminal emulator</td>
</tr>
</tbody>
</table>

Commands whose behavior depends on the type of terminal should accept arguments of the form \(-T\)term where term is one of the names given above; if no such argument is present, such commands should obtain the terminal type from the environment variable TERM, which, in turn, should contain term.

**FILES**

/usr/lib/terminfo/* compiled terminal description database

**SEE ALSO**
curses(3X), profile(4), terminfo(4), environ(5),
Chapter 9 of the Programmer's Guide.

**NOTES**

Not all programs follow the above naming conventions.
NAME

types – primitive system data types

SYNOPSIS

#include <sys/types.h>

DESCRIPTION

The data types defined in the include file are used in IRIX system code; some data of these types are accessible to user code:

```c
typedef struct { int r[1]; } *physadr;
typedef long daddr_t;
typedef char * caddr_t;
typedef unsigned char uchar;
typedef unsigned short ushort;
typedef unsigned int uint;
typedef unsigned long ulong;
typedef ulong ino_t;
typedef short cnt_t;
typedef long time_t;
typedef int label_t[12];
typedef short dev_t;
typedef long off_t;
typedef unsigned long paddr_t;
typedef int key_t;
typedef unsigned char use_t;
typedef short sysid_t;
typedef short index_t;
typedef unsigned int lock_t;
typedef unsigned int size_t;
typedef unsigned char u_char;
typedef unsigned short u_short;
typedef unsigned int u_int;
typedef unsigned long u_long;
```

The form `daddr_t` is used for disk addresses except in an i-node on disk, see `fs(4)`. Times are encoded in seconds since 00:00:00 GMT, January 1, 1970. The major and minor parts of a device code specify kind and unit number of a device and are installation-dependent. Offsets are measured in bytes from the beginning of a file. The `label_t` variables are used to save the processor state while another process is running.

SEE ALSO

`fs(4)`.

April 1990 - 1 -
NAME
values – machine-dependent values

SYNOPSIS
#include <values.h>

DESCRIPTION
This file contains a set of manifest constants, conditionally defined for particular processor architectures.

The model assumed for integers is binary representation (one’s or two’s complement), where the sign is represented by the value of the high-order bit.

BITS(type) The number of bits in a specified type (e.g., int).

HIBITS The value of a short integer with only the high-order bit set (in most implementations, 0x8000).

HIBITL The value of a long integer with only the high-order bit set (in most implementations, 0x80000000).

HIBITI The value of a regular integer with only the high-order bit set (usually the same as HIBITS or HIBITL).

MAXSHORT The maximum value of a signed short integer (in most implementations, 0x7FFF ≡ 32767).

MAXLONG The maximum value of a signed long integer (in most implementations, 0x7FFFFFFF ≡ 2147483647).

MAXINT The maximum value of a signed regular integer (usually the same as MAXSHORT or MAXLONG).

MAXFLOAT, LN_MAXFLOAT The maximum value of a single-precision floating-point number, and its natural logarithm.

MAXDOUBLE, LN_MAXDOUBLE The maximum value of a double-precision floating-point number, and its natural logarithm.

MINFLOAT, LN_MINFLOAT The minimum positive value of a single-precision floating-point number, and its natural logarithm.
MINDOUBLE, LN_MINDOUBLE

The minimum positive value of a double-precision floating-point number, and its natural logarithm.

FSIGNIF

The number of significant bits in the mantissa of a single-precision floating-point number.

DSIGNIF

The number of significant bits in the mantissa of a double-precision floating-point number.

SEE ALSO

intro(3), math(5).
NAME
varargs – variable argument list

SYNOPSIS

```
#include <varargs.h>

function(va-alist)
va_dcl
va_list pvar;
va_start(pvar);
f = va_arg(pvar, type);
va_end(pvar);
```

DESCRIPTION
This set of macros provides a means of writing portable procedures that accept variable argument lists. Routines having variable argument lists (such as `printf(3)`) that do not use varargs are inherently nonportable, since different machines use different argument passing conventions.

`va-alist` is used in a function header to declare a variable argument list.

`va_dcl` is a declaration for `va-alist`. Note that there is no semicolon after `va_dcl`.

`va_list` is a type which can be used for the variable `pvar`, which is used to traverse the list. One such variable must always be declared.

`va_start(pvar)` is called to initialize `pvar` to the beginning of the list.

`va_arg(pvar, type)` will return the next argument in the list pointed to by `pvar`. Type is the type to which the expected argument will be converted when passed as an argument. In standard C, arguments that are `char` or `short` should be accessed as `int`, `unsigned char` or `unsigned short` are converted to `unsigned int`, and `float` arguments are converted to `double`. Different types can be mixed, but it is up to the routine to know what type of argument is expected, since it cannot be determined at runtime.

`va_end(pvar)` is used to finish up.

Multiple traversals, each bracketed by `va_start ... va_end`, are possible.

EXAMPLE

```
#include <varargs.h>
execl(va-alist)
va_dcl
{
    va_list ap;
    char *file;
    char *args[100];
    int argno = 0;
```
va_start(ap);
file = va_arg(ap, char *);
while (args[argno++] = va_arg(ap, char *))
    ;
va_end(ap);
return execv(file, args);
}

SEE ALSO
stdafx(5)

BUGS
Due to the procedure calling convention on the MIPS processor, floating-
point parameters may be inaccessible via variargs unless they appear after a
parameter of non-floating-point type. Thus, in the code sequence

    extern int foo(float,...);
    foo(1.0,2.0);
the parameter 2.0 may be accessed incorrectly. If the function expected an
intervening non-floating-point parameter, such as
    extern int foo(float,...);
    foo(1.0,4,2.0);
the second floating-point parameter would be accessible as a double.

Variargs cannot be used when passing structures as parameters, as it is
impossible to determine their alignment at runtime.

It is up to the calling routine to determine how many arguments there are,
since it is not possible to determine this from the stack frame. For example,
exec1 passes a 0 to signal the end of the list. Printf can tell how many argu-
ments are supposed to be there by the format.

The macros va_start and va_end may be arbitrarily complex; for example,
va_start might contain an opening brace, which is closed by a matching
brace in va_end. Thus, they should only be used where they could be
placed within a single complex statement.
NAME

winicons – stowed window image mechanism

DESCRIPTION

When an active window is stowed by the user, an RGB image file may be
used to paint the canvas of the stowed window icon. A window icon file
must contain the suffix .icon.

Window icons are assigned to stowed windows by matching the name that
appears in the program’s call to the Graphics Library subroutine, wino-
pen. Thus, an icon for the GL program cedit would have this name:

cedit.icon

WINICON SEARCH PATH

A directory of default window icons exists in $NEWSHOME/icons. You
may add or customize window icons by placing them in
$HOME/.4sight/icons. To find the appropriate window icon for a stowed
window, 4Sight first searches $HOME/.4sight/icons for a name match. If it
is unsuccessful, it searches the default window icon directory
($NEWSHOME/icons). If this is unsuccessful, it uses the predefined icon
$NEWSHOME/icons/default.icon. If this icon is missing for some reason,
4Sight draws the icon without an image.

CREATING A WINICON FILE

A window icon can be created from any image that can be displayed on the
IRIS screen (provided that the tools described below are accessible when
the image is displayed).

The following passage describes one possible way to create an icon image
file. First, display the image you wish to use with the image tools ipaste or
showci, or simply open a window containing a program from which you
want to take an image (make sure that the image is still). Then invoke the
image tool icut from the command line. icut takes a filename as an argu-
ment; the image cut from the screen is written to that file.

icut foo

Place the small rectangular icut window away from the image you wish to
cut. Place the mouse cursor inside the icut window and hold down the
<shift> key. While holding down the key, move the mouse cursor to the
upper left corner of the area you wish to cut. Hold down the left mouse but-
ton while you move to the lower right corner of the area you wish to cut
(the area is not shown on the screen), and when you are ready, let go of the
left button. The image is then written to the file.
**Note:** The image you cut must be scaled to fit the stowed window canvas, so you should attempt to cut an area of the same general shape as the icon. The ratio of window icon width to height is 64:50.

You can use the file obtained with `icut` as a window icon file, and 4Sight will scale and dither it on the fly. However, to increase efficiency and image quality, you may want to scale it yourself. To do so, first run the image tool `istat` on the `icut` file:

```plaintext
istat foo
```

`istat` gives a readout of various image statistics; the important ones for scaling are the first two values, the image width (`xsize`) and image height (`ysize`). The dimension of stowed windows is 50 NeWS (4Sight) points high by 64 NeWS points wide. The ratio of screen pixels to points is 4:3; this yields the following scaling factors:

\[
xscale = \frac{85.33}{xsize} \\
yscale = \frac{66.67}{ysize}
\]

To scale the image, use the image tool `izoom`. `izoom` takes in input file, an output file, and width and height scaling factors as arguments. To scale an image file `foo` whose dimensions are 620 pixels wide by 500 pixels high and make it into a console window icon, you would type the following on the command line:

```plaintext
izoom foo $HOME/./4sight/icons/console.icon .137 .133
```

**NOTES**

Some programs do not use this mechanism to draw their window icons; specifically, those that draw their own icons rather than let 4Sight do it for them. Certain NeWS programs, such as the Calculator draw their own icons in PostScript; Graphics Library programs using `icons` to draw their icons use Graphics Library commands to do so. Both of these methods override the window icon mechanism described here.

**SEE ALSO**

`ipaste(1G)`, `icons(3G)`
PERMUTED INDEX

font names. xfonts: point
maze: an automated maze program...
[ demo ] [ X11 ]
xclock: analog /digital clock for X.
maze program... [ demo ] [ X11 ] maze: an automated
an automated maze program... [ demo ] [ X11 ] maze: an automated
x10tox11: X version 10 to version 11
ik: Ikon
x10tox11: X version 10 to version 11
protocol converter.
x10tox11(1)
protocol converter.
x10tox11(1)
xmt: Xylogics
1/2 inch magnetic tape controller.
xmt(7M)
house: 2D to 3D architecture demo.
house(6D)
set group access list (berkeley 4.3BSD).
t3270: Silicon Graphics
and group of a file (System V and 4.3BSD).
chown, fchown: change owner
dirdl: directory operations (4.3BSD).
CTIME(3C).
kill: send signal to a process (4.3BSD).
send signal to a process group (4.3BSD).
file pointer (System V and 4.3BSD).
process group ID (System V and 4.3BSD).
signals from delivery to process (4.3BSD).
signal: simplified
software signal facilities (4.3BSD).
signals and wait for interrupt (4.3BSD).
set current signal mask (4.3BSD).
GETDTABLESIZE(3). 4.
3-byte integers and long integers.
3D architecture demo.
house(6D)
graphs demographic data in
3D over time.
diff3: 3-way differential file comparison.
diff3(1)
set group access list (berkeley 4.3BSD).
initialize group access list (bsd 4.3BSD).
get group access list (berkeley 4.3BSD).
and group of a file (System V and 4.3BSD).
dirdl: directory operations (4.3BSD).
CTIME(3C). /closedir,
kill: send signal to a process (4.3BSD).
send signal to a process group (4.3BSD).
file pointer (System V and 4.3BSD).
set process group ID (System V and 4.3BSD).
signals from delivery to process (4.3BSD).
signal: simplified
software signal facilities (4.3BSD).
signals and wait for interrupt (4.3BSD).
sigsetmask: set current signal mask (4.3BSD).
GETDTABLESIZE(3).
4.3BSD software signal facilities
sigvec:
/u3b15, vax, m68k, m68000, mips,
/u3b15, vax, m68k, m68000, mips, 4d
/dn_ll, dn_netman
/toolchest, journalchest
4D special files.
dn(7)
4 Sight event record and playback.
journalchest(1W)

@: arithmetic on shell variables.
& click interface for selecting X11.
xfonts(1)
xfontsel(1)
maze(1)

April 1990
Version 5.0
from an/ journalrecord, journalend:
  4Sight event recording and playback . journalplay(1)
toolchest, windowchest, demochest:
  4Sight utility and windowing/ ............ toolchest(1W)
gse: Silicon Graphics
  5080 workstation interface card. ........ gse(7)
cdsio:
  6-port serial I/O ..................... cdsio(7)
integer and base-64 ASCII string.
  a64l, i64a: convert between long .... a64l(3C)
  with a core dump.
  abort: terminate current process .... abort(3C)
  abs: return integer absolute value. .... abs(3C)
  cabs: Euclidean distance, complex
  absolute value. hypot, ............... hypot(3M)
/frunc: floor, ceiling, remainder,
  absolute value, nearest integer, and/
  accept: accept a connection on a socket.
  alloc, almallopt, allocinfo/
  socket. ................................ accept(2)
  accept: accept a connection on a ..... accept(2)
  input. read: accept input from the standard
  requests. ................................ sh(1)
/to basic functions needed to
  access and add to the symbol table. .. stfe(3X)
  file. touch: update access and modification times of a
  utime: set file access and modification times. ... utime(2)
  xhost: server access control program for X. ... xhost(1)
  file. access: determine accessibility of a .... access(2)

INITGROUPS_BSD(3B). set group
  access list (berkeley 4.3 version) ... setgroups:
  SETGROUPS(3C). get group
  access list (berkeley 4.3 version) ... getgroups:
  READV(3C). initialize group
  access list (bsd 4.3 version) ......... initgroups:
  initialize group access list CFGETOSPEED(3T).
  getgroups: get group access list. ....... getgroups(2)
  initialize group access list GETGROUPS(3B).
  setgroups: set group access list. ........ setgroups(2)

machine-independent/ spuit, sgetl:
  access long integer data in a ........ spuit(3X)
randhashinit, randhash, ranlookup:
  access routine for the symbol table/ ... ranhash(3X)
lfcn: common object file
  access routines. ...................... lfcn(4)
  mknetpr: provide access to a remote printer.
  section/ stdfs: routines that provide
  access to per file descriptor ........ stdfs(3X)
  setutent, endutent, utmpname:
  access utmp file entry. /pututiline, ... getut(3C)
  access: determine accessibility of a file. access(2)

acct: enable or disable process
  accounting. .......................... acct(2)
acctcon1, acctcon2: connect-time
  accounting. .......................... acctcon(1M)
acctprc1, acctprc2: process
  accounting. .......................... acctprc(1M)
turnacct: shell procedures for
  accounting. /shutacct, startup, ...... acctsh(1M)
/accton, acctwtmp: overview of
  accounting and miscellaneous
  accounting commands. /overview ...... acct(1M)
diskusg: generate disk
  accounting data by user ID. .......... diskusg(1M)
acct: per-process
  accounting file format. .............. acct(4)
acctcom: search and print process
  accounting file(s). .................. acctcom(1)
acctmerge: merge or add total
  accounting files. .................... acctmerge(1M)
pac: printer/plotter
  accounting information. ............ pac(1M)
command summary from per-process
  accounting records. acctcms: ....... acctcms(1M)
fwtmp, wtmpfix: manipulate connect
  accounting records. ................. fwttmp(1M)
runcacct: run daily
  accounting. .......................... runacct(1M)
  acct: enable or disable process .... acct(2)
  format. ............................. acct(4)
  acct: per-process accounting file .... acct(4)
  per-process accounting records. .... acctcms: command summary from acctcms(1M)

Version 5.0

April 1990
accounting file(s).  acctcom: search and print process  acctcom: acctcom(1)
accounting.  acctcom1, acctcom2: connect-time  acctcom: acctcom(1M)
acctcom1, acctwtmp: overview of accounting/ acctdisk, acctdusg, accton, acct(1M)
of accounting and/ acctdisk, accounting files.
accounting and/ acctdisk, acctdusg,
acctdisk, acctdusg, accton, acctwtmp: overview
acct(1M)
acctwtmp: overview of accounting and/ acct(1M)
times: print  accumulated times.  sh(1)
fasin, facos,/ sin, cos, tan, asin,  acos, atan, atan2, fsin, fcos, ftan, trig(3M)
functions. asinh, acosh, atanh: inverse hyperbolic
not possible. uscspema: attempts to acquire a semaphore, and fails if
uspsema: attempt to acquire a semaphore. uspsema(3P)
arealloc, acalloc, amallopt,/ acreate, adelete, amalloc, afree, amalloc(3P)
tcflo (int fildes, int
action); GETRPCPORT(3R).  int
osview: monitor operating system  activity data.  osview(1)
sa1, sa2, sadc: system  activity report package. sar(1M)
sar: system  activity reporter. sar(1)
print current SCCS file editing  activity. sact: sact(1)
report process data and system  activity. timex: time a command; timex(1)
Add_disk: add a secondary disk to the system. add_disk(1)
 pixie: add profiling code to a program. pixie(1)
environment. export: add shell variables to the sh(1)
basic functions needed to access and
add to the symbol table. /to stfe(3X)
acctmerg: merge or
putenv: change or
clients to connect.
the system. Add_disk: add a secondary disk to
add total accounting files. acctmerg(1M)
add value to environment. putenv(3C)
addclient: allow remote printing addclient(1M)
add client: create and administer SCCS admin(1)

registerinethod: allocate internet
address for workstation. registerinethod(3N)
/inet_inaof, inet_netof: Internet
address manipulation routines. inet(3N)
control. arp: address resolution display and
arp: Address Resolution Protocol. arp(1M)
arp(7P)
acalloc, amallopt, / acreate,
admin: create and administer SCCS admin(1)
files. admin: create and

SA: devices administered by System
SA: devices administered by System
administered by System Administration. sa(7)

sysadm: menu interface to do system administration.
sysad(1)
vadmin: interactive system administration tool.
vadmin(1G)
udadmin: administrative control. uadmin(1M)
udadmin: administrative control. uadmin(2)

madvice: give advise about handling memory. madvice(2)
flock: apply or remove an advisory lock on an open file. flock(3B)
acreate, adelete, amalloc, afree, arealloc, acalloc, amallopt,/ amalloc(3P)
vmsprep: VMS tape preparation
aid.  vmsspre(1)
the flight of any of several aircraft. flight: simulate flight(6D)
<table>
<thead>
<tr>
<th>Permutted Index</th>
<th>Silicon Graphics</th>
</tr>
</thead>
<tbody>
<tr>
<td>or competitive flight simulator and</td>
<td>dog: cooperative</td>
</tr>
<tr>
<td>alarm: set a process</td>
<td>dog(6D)</td>
</tr>
<tr>
<td>alarm clock.</td>
<td>alarm(2)</td>
</tr>
<tr>
<td>alarm: set a process alarm clock.</td>
<td>alarm(2)</td>
</tr>
<tr>
<td>alias: shell macros.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>aliases: aliases file for sendmail.</td>
<td>aliases(4)</td>
</tr>
<tr>
<td>locate a program file including</td>
<td>aliases and path (csh only). which:</td>
</tr>
<tr>
<td>unalias: remove</td>
<td>which(1)</td>
</tr>
<tr>
<td>aliases.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>aliases file for sendmail.</td>
<td>aliases(4)</td>
</tr>
<tr>
<td>rebuild the data base for the mail workstation.</td>
<td>newaliases(1M)</td>
</tr>
<tr>
<td>registerinethost:</td>
<td>registerinethost(3N)</td>
</tr>
<tr>
<td>usnewlock: allocates and initializes a lock.</td>
<td>usnewlock(3P)</td>
</tr>
<tr>
<td>semaphore. usnewsem: allocates and initializes a</td>
<td>usnewsem(3P)</td>
</tr>
<tr>
<td>brk, sbk: change data segment space</td>
<td>brk(2)</td>
</tr>
<tr>
<td>arbitrary arena main memory</td>
<td>malloc(3P)</td>
</tr>
<tr>
<td>free, realloc, calloc: main memory</td>
<td>malloc(3C)</td>
</tr>
<tr>
<td>malloc, mallinfo: fast main memory</td>
<td>malloc(3X)</td>
</tr>
<tr>
<td>usmallinfo: user shared memory</td>
<td>malloc(3P)</td>
</tr>
<tr>
<td>scandir, alphsort: scan a directory.</td>
<td>scandir(3C)</td>
</tr>
<tr>
<td>the set of blocked/ sigprocmask: alter and return previous state of</td>
<td>sigprocmask(2)</td>
</tr>
<tr>
<td>renice: alter priority of running processes.</td>
<td>renice(1M)</td>
</tr>
<tr>
<td>else: alternative commands.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>else: alternative commands.</td>
<td>sh(1)</td>
</tr>
<tr>
<td>/afile, arealloc, acalloc, amallopt, alaminfo: arbitrary arena main/</td>
<td>amalloc(3P)</td>
</tr>
<tr>
<td>amallopt, / acreate, adelete, amalloc, arealloc, acalloc, amallopt,</td>
<td>amalloc(3P)</td>
</tr>
<tr>
<td>/ amalloc, afile, arealloc, acalloc, alaminfo: arbitrary arena/</td>
<td>amalloc(3P)</td>
</tr>
<tr>
<td>xclock: analog / digital clock for X.</td>
<td>xclock(1)</td>
</tr>
<tr>
<td>clock: analog clock in a window.</td>
<td>clock(6D)</td>
</tr>
<tr>
<td>the results of a finite element analysis program. / display</td>
<td>solidview(6D)</td>
</tr>
<tr>
<td>prof: analyze profile data.</td>
<td>prof(1)</td>
</tr>
<tr>
<td>pixstats: analyze program execution.</td>
<td>pixstats(1)</td>
</tr>
<tr>
<td>flush contents of instruction and/or data cache. cacheflush:</td>
<td>cacheflush(2)</td>
</tr>
<tr>
<td>sort: sort and/or merge files.</td>
<td>sort(1)</td>
</tr>
<tr>
<td>polyhedron. ico: animate an ico.sahedron or other</td>
<td>ico(1)</td>
</tr>
<tr>
<td>swapping buffers to display smooth animation.. swap: demonstrates</td>
<td>swap(6T)</td>
</tr>
<tr>
<td>ansi tape: ANSI standard tape handler.</td>
<td>ansi tape(1)</td>
</tr>
<tr>
<td>intro: introduction to commands, application programs, and/</td>
<td>intro(1)</td>
</tr>
<tr>
<td>to maintenance commands and application programs. / introduction</td>
<td>intro(1M)</td>
</tr>
<tr>
<td>appres: list application resource database.</td>
<td>appres(1)</td>
</tr>
<tr>
<td>xlsclients: list client applications running on a display.</td>
<td>xlsclients(1)</td>
</tr>
<tr>
<td>winterm: utility to launch applications that require a terminal/</td>
<td>winterm(1)</td>
</tr>
<tr>
<td>arguments. apply: apply a command to a set of arguments. apply: apply a</td>
<td>apply(1)</td>
</tr>
<tr>
<td>command to a set of arguments. an open file. flock: apply or remove an</td>
<td>apply(1)</td>
</tr>
<tr>
<td>advisory lock on database. appres: list application resource</td>
<td>flock(3B)</td>
</tr>
<tr>
<td>lookup. apropos: locate commands by keyword</td>
<td>apropos(1)</td>
</tr>
<tr>
<td>&quot;whatis&quot; database for use with apropos. / make manual page</td>
<td>makewhatis(1M)</td>
</tr>
<tr>
<td>ar: archive and library maintainer. ar: archive (library) file format.</td>
<td>ar(1)</td>
</tr>
<tr>
<td>ar: archive (library) file format.</td>
<td>ar(4)</td>
</tr>
<tr>
<td>/ amalloc, amallopt, amalinfo: arbitrary arena main memory/</td>
<td>amalloc(3P)</td>
</tr>
<tr>
<td>language. bc: arbitrary-precision arithmetic</td>
<td>bc(1)</td>
</tr>
</tbody>
</table>
Silicon Graphics

Permutated Index

- architecture demo.
- archive and library maintainer.
- archive.
- archive header of a member of an archive file. Idahread: read the archive header of a member of an archive file.
- archive (library) file format.
- archiver.
- archives in and out.
- archives. /access routine for acreate, adelate, amalloc, afree,
- arealloc, acalloc, malloc,.
- arena: a future sport.
- arena configuration operations.
- arena main memory allocator.
- arena USDUMPLOCK(3P). usputinfo: usgetinfo,
- argument list.
- argument vector.
- arguments and invoke commands.
- arguments.
- arguments as an expression.
- arguments.
- arguments.
- arithmetic language.
- arithmetic on shell variables.
- as a NeWS client.
- as an expression.
- as brushes.
- as: MIPS assembler.
- as: MIPS assembler.
- as: MIPS assembler.
- as: MIPS assembler.
- assert: program verification.
- assign buffering to a stream.
- at, at: execute commands at a later time.
- at login time.
- at(1)
- at(1)
- at(1)
- at low priority.

house: 2D to 3D
ar: archive and library maintainer.
cpio: format of cpio the archive header of a member of an archive file. Idahread: read the
ar: archive (library) file format.
tar: tape
the symbol table definition file in
varargs: variable
print formatted output of a variable
command. xargs: construct getopt: get option letter from
getopt: get option letter from
launch: graphical utility to launch arguments and invoke commands.
apply: apply a command to a set of
apply: apply a command to a set of
express: evaluate arguments
express: evaluate arguments
echo: echo
echo: echo
bc: arbitrary-precision arithmetic language.
@:
dogfight. shadow: full-screen
three colored lights bouncing around a scene. bounce:
ARP: address resolution display and
ARP: Address Resolution Protocol.
xstart: start up the X server as a NeWS client.
expr: evaluate arguments
ipaint: Paint using bitmap images
localtime, gmtime, asctime, cftime, between long integer and base-64
asctime, cftime, asctime, strftime,
ftime, fsin, facos, sin, cos, tan,
hyperbolic functions.
messages and commands. help:
ask for help about SCCS error
assembler and link editor output.
asserter: program verification.
/setbuf, setbuffer, setlinebuf:
estabuf(3S)
at, batch: execute commands at a later time.
profile: setting up an environment
nice: run a command at low priority.

April 1990

Version 5.0

-5-
proto: prototype job file for
fasin/, sin, cos, tan, asin, acos,
sin, cos, tan, asin, acos, atan,
asinh, acosh,
file. queuedefs:
utilities for X. bitmap, bmtoa,
double-precision number. strtdc,
strtol, atol,
integer. strtol,
and wait for interrupt/ sigpause:
and wait for interrupt/ sigsuspend:
xslatoms: list interned
uspmesa: attempt to acquire a semaphore.
loginlog: log of failed login
fails if not possible. uscpmsena:
boing: gravitationally attractive bouncing balls.
change login password and password
resources: change RCS file
interface. audio: bi-directional audio channel
audio: bi-directional audio channel interface.
xauth: X authority file utility.
identity. autologin: set autologin user
autologin: set autologin user identity.
that provide scalar interfaces to auxiliaries. staux: routines
ldgetaux: retrieve an auxiliary entry, given an index.
xload: load average display for X.
wait: await completion of process.
language. awk: pattern scanning and processing
wait: wait for background processes to complete.
wait: wait for background processes to complete.
bru: backup and restore utility.
directory. Backup: backup the specified file or
ckbupsed: check file system backup schedule.
list the contents of a given backup tape. List_tape:
directory. Backup: backup the specified file or
gravitationally attractive bouncing balls. boing:
init_barrier, free_barrier:
free_barrier: barrier functions. /new_barrier,
newaliases: rebuild the data base for the mail aliases file.
hosts: host name data base.
networks: network name data base.
ttytype: data base of terminal types by port.
printcap: printer capability data base.
protocols: protocol name data base.
eexisting hostname in yp hosts data base. renamehost: rename the
tcp: RPC program number data base.
services: service name data base.
delete, firstkey, nextkey: data base subroutines. /fetch, store, 
dbm_error, dbm_clearerr: data base subroutines. /dbm_nextkey,
<table>
<thead>
<tr>
<th>Permutated Index</th>
<th>Silicon Graphics</th>
</tr>
</thead>
<tbody>
<tr>
<td>sync: update super</td>
<td>block.</td>
</tr>
<tr>
<td>sigpause: atomically release</td>
<td>blocked signals and wait for/</td>
</tr>
<tr>
<td>sigsuspend: atomically release</td>
<td>blocked signals and wait for/</td>
</tr>
<tr>
<td>return previous state of the set of</td>
<td>blocked signals (POSIX). /alter and</td>
</tr>
<tr>
<td>setblockproc, blockprocall, /unblockproc, setblockprocall,</td>
<td>blockproc, unblockproc,</td>
</tr>
<tr>
<td>df: report number of free disk</td>
<td>blocks.</td>
</tr>
<tr>
<td>/setblockprocentall: routines to</td>
<td>block/unblock processes.</td>
</tr>
<tr>
<td>tasksetblockcent: routines to</td>
<td>block/unblock tasks. /taskunblock,</td>
</tr>
<tr>
<td>converter utilities for X. bitmap, jello: simulates nonrigid</td>
<td>body dynamics.</td>
</tr>
<tr>
<td>bouncing balls.</td>
<td>boing: gravitationally attractive</td>
</tr>
<tr>
<td>mkboottape: make a boot tape.</td>
<td>mkboottape(1M)</td>
</tr>
<tr>
<td>lboot: configure</td>
<td>bootable kernel.</td>
</tr>
<tr>
<td>Protocol.</td>
<td>bootp: server for Internet Bootstrap</td>
</tr>
<tr>
<td>bootp: server for Internet</td>
<td>Bootstrap Protocol.</td>
</tr>
<tr>
<td>bouncing around a scene.</td>
<td>bounce: three colored lights</td>
</tr>
<tr>
<td>bounce: three colored lights</td>
<td>bouncing around a scene.</td>
</tr>
<tr>
<td>boing: gravitationally attractive</td>
<td>bouncing balls.</td>
</tr>
<tr>
<td>switch: multi-way command</td>
<td>branch.</td>
</tr>
<tr>
<td>emulate_branch: MIPS</td>
<td>branch emulation.</td>
</tr>
<tr>
<td>case: multi-way command</td>
<td>branch.</td>
</tr>
<tr>
<td>procedures.</td>
<td>bre, bcheckrc: system initialization</td>
</tr>
<tr>
<td></td>
<td>break: exit while/for loop.</td>
</tr>
<tr>
<td></td>
<td>break: exit while/foreach loop.</td>
</tr>
<tr>
<td></td>
<td>breaksw: exit from switch.</td>
</tr>
<tr>
<td>allocation.</td>
<td>brk, sbk: change data segment space</td>
</tr>
<tr>
<td></td>
<td>bru: backup and restore utility.</td>
</tr>
<tr>
<td>ipaint: Paint using bitmap images as</td>
<td>brushes.</td>
</tr>
<tr>
<td>initialize group access list</td>
<td>(bsd 4.3 version) READV(3C).</td>
</tr>
<tr>
<td>(System V and 4.3BSD). setgpp,</td>
<td>BSDsetgpp: set process group ID</td>
</tr>
<tr>
<td>table.</td>
<td>bsearch: binary search a sorted</td>
</tr>
<tr>
<td>xcutset: interchange between cut</td>
<td>bstream: many buffered filter.</td>
</tr>
<tr>
<td>bstream: many buffered filter.</td>
<td></td>
</tr>
<tr>
<td>psio: NeWS</td>
<td>buffered input/output package.</td>
</tr>
<tr>
<td>stdio: standard</td>
<td>buffered input/output package.</td>
</tr>
<tr>
<td>setbuffer, setlinebuf: assign</td>
<td>buffering to a stream. /setbuf;</td>
</tr>
<tr>
<td>write output gathered from buffers GETRUSAGE(3).</td>
<td>writev:</td>
</tr>
<tr>
<td>swap: demonstrates swapping</td>
<td>buffers to display smooth/</td>
</tr>
<tr>
<td>read input to scattered</td>
<td>buffers WRITEV(3C).</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>(FIFO). mknod:</td>
<td>build font family description.</td>
</tr>
<tr>
<td>for Silicon Graphics demos.</td>
<td>build special file or named pipe</td>
</tr>
<tr>
<td>values between host and network</td>
<td>buttonify: a pretty user interface</td>
</tr>
<tr>
<td>swap_*() - swap/ gethostxss: get the byte order.</td>
<td>byte order. /ntohl, ntohs: convert</td>
</tr>
<tr>
<td>bcopy, bcmcp, blkclr, bzero:</td>
<td>byte string operations.</td>
</tr>
<tr>
<td>swab: swap</td>
<td>bytes.</td>
</tr>
<tr>
<td>bcopy, bcmcp, blkclr, bzero:</td>
<td>byte string operations.</td>
</tr>
<tr>
<td>unifdef: strip or reduce ifdefs in C code.</td>
<td>C code.</td>
</tr>
<tr>
<td>cc: MIPS</td>
<td>C compiler.</td>
</tr>
</tbody>
</table>

Version 5.0  –8–  April 1990
 Silicon Graphics

FORTRAN-callable entry points from a
C file. extentropy: extract extentropy(1)
cflow: generate cflow(1)
cpp: the C language preprocessor. cpp(1)
utility. imake: preprocessor interface to the make imake(1)
ch: program beautifier. ch(1)
lint: a program checker. lint(1)
cxref: generate C program cross-reference. cxref(1)
ctrace: program debugger. ctrace(1)
strings. xstr: extract strings from C programs to implement shared xstr(1)
an error message file by massaging C source. mkstr: create mkstr(1)
cps: construct to PostScript interface. cps(1)
absolute value. hypot, cabs: Euclidean distance, complex hypot(3M)
contents of instruction and/or data cache. cacheflush: flush cacheflush(2)
an executable to facilitate better cache mapping. /procedures in cache(1)
cachectl: mark pages cacheable or uncacheable. cachectl(2)
uncacheable. cachectl: mark pages cacheable or cachectl(2)
instruction and/or data cache. cacheflush: flush contents of cacheflush(2)
cal: print calendar. cal(1)
display computation fluid dynamics calculations. vortex vortex(6D)
dc: desk calculator. dc(1)
xcalc: scientific calculator for X. xcalc(1)
sc: spread sheet calculator. sc(1)
cal: print calendar. cal(1)
ical: calendar. ical(1G)
calendar: reminder service. calendar(1)
compute difference between two calendar times INSQNE(3). difftime:
xcalendar: calendar with a notebook for X11. xcalendar(1)
gamecal: visually check display calibration. gamecal(6D)
cu: call another UNIX system. cu(1C)
rpc: Remote Procedure Call (RPC) library routines. rpc(3R)
schedctl: scheduler control call schedctl(2)
sgisc: SGI graphics system call. sgisc(2)
stat: data returned by stat system call. stat(5)
MIPS Computer Systems Inc. system call. sysmips: sysmips(2)
syssgi: Silicon Graphics Inc. system call. syssgi(2)
texturebind: SGI graphics system call. texturebind(2)
malloc, free, realloc, calloc: main memory allocator. malloc(3C)
memory/ malloc, free, realloc, calloc, malloc, mallinfo: fast main malloc(3X)
tintro: introduction to system calls and error numbers. intro(2)
LP line printer. lp, cancel: send/cancel requests to an cancel, send/lp(1)
printcap: printer capability data base. printcap(4)
terminfo: terminal capability data base. terminfo(4)
description into a terminfo/ convert a termcap captinfo(1M)
protocols. drain: capture unimplemented link-layer drain(7P)
Graphics 5080 workstation interface card. gse: Silicon gse(7)
pnch: file format for card images. pnch(4)
Silicon Graphics 3270 interface card. t3270: t3270(7)
ts: ISI VME-QIC2/X cartridge tape controller. ts(7M)
tps: SCSI 1/4-inch Cartridge tape interface. tps(7M)
case: multi-way command branch. sh(1)
case: selector in switch. csh(1)

April 1990

Version 5.0
<table>
<thead>
<tr>
<th>Permutated Index</th>
<th>Silicon Graphics</th>
</tr>
</thead>
<tbody>
<tr>
<td>esac: terminate</td>
<td>case.</td>
</tr>
<tr>
<td>edit: text editor (variant of ex for casual users).</td>
<td>edit(1)</td>
</tr>
<tr>
<td>cat: catenate and print files.</td>
<td>cat(1)</td>
</tr>
<tr>
<td>default: catchall clause in switch.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>cb: C program beautifier.</td>
<td>cb(1)</td>
</tr>
<tr>
<td>sqrt, fsqrt, cbtr: cube root, square root.</td>
<td>sqrt(3M)</td>
</tr>
<tr>
<td>cc: MIPS C compiler.</td>
<td>cc(1)</td>
</tr>
<tr>
<td>cd: change directory.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>cd: change directory.</td>
<td>sh(1)</td>
</tr>
<tr>
<td>cd: change working directory.</td>
<td>cd(1)</td>
</tr>
<tr>
<td>an SCCS delta.</td>
<td></td>
</tr>
<tr>
<td>cdc: change the delta commentary of cdc(1)</td>
<td></td>
</tr>
<tr>
<td>cdsio: 6-port serial I/O.</td>
<td>cdsio(7)</td>
</tr>
<tr>
<td>cedit: edit colors on the screen.</td>
<td>cedit(6D)</td>
</tr>
<tr>
<td>trunc, ftrunc/: floor, ffloor, /fabs, rint, trunc, ftrunc: floor, LP. mkcentp: register a color Centronics-interface printer with mkcentp(1M)</td>
<td></td>
</tr>
<tr>
<td>cfgetispeed: posix baud rate/ *termios_p, speed_t speed); speed_t rate primitives #include speed_t initialize group access list CGETOSPEED(3T). initgroups: cflow: generate C flowgraph. cflow(1)</td>
<td></td>
</tr>
<tr>
<td>cfgetispeed, cfsetospeed, cfgetispeed: struct terminios / terminios int *termios_p, speed_t speed); speed_t rate primitives/ cfgetispeed, cfgetispeed: struct terminios / baud cfgetospeed, cfgetospeed,</td>
<td></td>
</tr>
<tr>
<td>*termios_p, speed_t speed); speed_t/ cfgetispeed,</td>
<td></td>
</tr>
<tr>
<td>convert/ localtime, gmtime, asctime, cftime, asctime, strftime, tzset: ctime, cftime: language specific strings. cftime(4)</td>
<td></td>
</tr>
<tr>
<td>allocation. brk, sbbrk: change data segment space brk(2)</td>
<td></td>
</tr>
<tr>
<td>cd: change directory. csh(1)</td>
<td></td>
</tr>
<tr>
<td>chdir: change directory. csh(1)</td>
<td></td>
</tr>
<tr>
<td>cd: change directory. sh(1)</td>
<td></td>
</tr>
<tr>
<td>attributes. passwd: change login password and password passwd(1)</td>
<td></td>
</tr>
<tr>
<td>chmod, fchmod: change mode of file. chmod(2)</td>
<td></td>
</tr>
<tr>
<td>putenv: change or add value to environment. putenv(3C)</td>
<td></td>
</tr>
<tr>
<td>mask. umask: change or display file creation csh(1)</td>
<td></td>
</tr>
<tr>
<td>mask. umask: change or display file creation sh(1)</td>
<td></td>
</tr>
<tr>
<td>ulimit: change or display size limits. sh(1)</td>
<td></td>
</tr>
<tr>
<td>(System V and/ chown, fchown: change owner and group of a file chown(2)</td>
<td></td>
</tr>
<tr>
<td>chown, chgrp: change owner or group. chown(1)</td>
<td></td>
</tr>
<tr>
<td>nice: change priority of a process. nice(2)</td>
<td></td>
</tr>
<tr>
<td>rcs: change RCS file attributes. rcs(1)</td>
<td></td>
</tr>
<tr>
<td>chroot: change root directory. chroot(2)</td>
<td></td>
</tr>
<tr>
<td>chroot: change root directory for a command. chroot(1M)</td>
<td></td>
</tr>
<tr>
<td>shutdown: shut down system, change system state. shutdown(1M)</td>
<td></td>
</tr>
<tr>
<td>SCCS delta. cdc: change the delta commentary of an cdc(1)</td>
<td></td>
</tr>
<tr>
<td>newform: change the format of a text file. newform(1)</td>
<td></td>
</tr>
<tr>
<td>rename: change the name of a file. rename(2)</td>
<td></td>
</tr>
<tr>
<td>file or directory. chmod: change the permissions mode of a chmod(1)</td>
<td></td>
</tr>
<tr>
<td>delta: make a delta (change) to an SCCS file. delta(1)</td>
<td></td>
</tr>
<tr>
<td>set: change value of shell variable. csh(1)</td>
<td></td>
</tr>
<tr>
<td>cd: change working directory. cd(1)</td>
<td></td>
</tr>
</tbody>
</table>

Version 5.0

April 1990
<table>
<thead>
<tr>
<th>Silicon Graphics</th>
<th>Permutated Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>chdir: change working directory.</td>
<td>chdir(2)</td>
</tr>
<tr>
<td>open file descriptor, fchdir: change working directory, given an</td>
<td>fchdir(2)</td>
</tr>
<tr>
<td>/number generator; routines for changing generators INITGROUPS(3). random,</td>
<td></td>
</tr>
<tr>
<td>audio: bi-directional audio channel interface.</td>
<td>audio(7)</td>
</tr>
<tr>
<td>pipe: create an interprocess channel.</td>
<td>pipe(2)</td>
</tr>
<tr>
<td>ungetc: push character back into input stream.</td>
<td>ungetc(3S)</td>
</tr>
<tr>
<td>conversion tables. chrtbl: generate character classification and chrtbl(1M)</td>
<td></td>
</tr>
<tr>
<td>_tolower, _toupper, sethclass: character handling. /toascii, ctype(3C)</td>
<td></td>
</tr>
<tr>
<td>cuserid: get character login name of the user. cuserid(3S)</td>
<td></td>
</tr>
<tr>
<td>gete, getchar, fgetc, getw: get character or word from a stream. getc(3S)</td>
<td></td>
</tr>
<tr>
<td>pute, putchar, fpute, putw: put character or word on a stream. pute(3S)</td>
<td></td>
</tr>
<tr>
<td>of the standard supported character set. charset: description charset(5)</td>
<td></td>
</tr>
<tr>
<td>fgrep: search a file for a character string. fgrep(1)</td>
<td></td>
</tr>
<tr>
<td>_tolower, toascii: translate characters. /tolower, _toupper, conv(3C)</td>
<td></td>
</tr>
<tr>
<td>tr: translate characters. tr(1)</td>
<td></td>
</tr>
<tr>
<td>lastlogin, monacct, nulladm/charset: description of the standard charset(5)</td>
<td></td>
</tr>
<tr>
<td>supported character set. chdir: change working directory. chdir(1)</td>
<td></td>
</tr>
<tr>
<td>fsck, dfsc: check and repair file systems. fsck(1M)</td>
<td></td>
</tr>
<tr>
<td>logical volumes. lvck: check and restore consistency of lvck(1M)</td>
<td></td>
</tr>
<tr>
<td>gamcal: visually check display calibration. gamcal(6D)</td>
<td></td>
</tr>
<tr>
<td>cbkupsd: check file system backup schedule. cbkupsd(1M)</td>
<td></td>
</tr>
<tr>
<td>ci: check in RCS revisions. ci(1)</td>
<td></td>
</tr>
<tr>
<td>co: check out RCS revisions. co(1)</td>
<td></td>
</tr>
<tr>
<td>check: check RCS status of a file. check(1)</td>
<td></td>
</tr>
<tr>
<td>permissions file. uuchek: check the uucp directories and uuchek(1M)</td>
<td></td>
</tr>
<tr>
<td>chkconfig: configuration state checker. chkconfig(1M)</td>
<td></td>
</tr>
<tr>
<td>lint: a C program checker. lint(1)</td>
<td></td>
</tr>
<tr>
<td>pwck, grpck: password/group file checkers. pwck(1M)</td>
<td></td>
</tr>
<tr>
<td>type rules. isSuper: supertype checking utility for use with file issuper(1)</td>
<td></td>
</tr>
<tr>
<td>sum: print checksum and block count of a file. sum(1)</td>
<td></td>
</tr>
<tr>
<td>chown, chgrp: change owner or group. chown(1)</td>
<td></td>
</tr>
<tr>
<td>times: get process and child process times. times(2)</td>
<td></td>
</tr>
<tr>
<td>wait, waitpid, wait3: wait for child processes to stop or/ wait(2)</td>
<td></td>
</tr>
<tr>
<td>checker. chkconfig: configuration state chkconfig(1M)</td>
<td></td>
</tr>
<tr>
<td>of a file or directory. chmod: change the permissions mode chmod(1)</td>
<td></td>
</tr>
<tr>
<td>chmod, fchmod: change mode of file. chmod(2)</td>
<td></td>
</tr>
<tr>
<td>chown, chgrp: change owner or group. chown(1)</td>
<td></td>
</tr>
<tr>
<td>group of a file (System V and/ chown, fchown: change owner and chown(2)</td>
<td></td>
</tr>
<tr>
<td>command. chroot: change root directory. chroot(2)</td>
<td></td>
</tr>
<tr>
<td>classification and conversion/ chroot: change root directory for a chroot(1M)</td>
<td></td>
</tr>
<tr>
<td>chrtbl: generate character chrtbl(1M)</td>
<td></td>
</tr>
<tr>
<td>ci: check in RCS revisions. ci(1)</td>
<td></td>
</tr>
<tr>
<td>schedule. cbkupsd: check file system backup cbkupsd(1M)</td>
<td></td>
</tr>
<tr>
<td>nulladm, protmp, /chargefee, cplacet, dodisk, lastlogin, monacct, acctsh(1M)</td>
<td></td>
</tr>
<tr>
<td>values. fp_class: classes of IEEE floating-point fp_class(3C)</td>
<td></td>
</tr>
<tr>
<td>tables. chrtbl: generate character classification and conversion chrtbl(1M)</td>
<td></td>
</tr>
<tr>
<td>default: catchall clause in switch. csh(1)</td>
<td></td>
</tr>
<tr>
<td>uucleanup: uucp spool directory clean-up. uucleanup(1M)</td>
<td></td>
</tr>
</tbody>
</table>
clear: clear terminal screen.
clear-i-node: clear i-node.
gclear: clear IRIS graphics screen.
clear: clear terminal screen.
inquiries, ferror, feof, ferrorr, fileno: stream status
font names. xfontsel: point &
display. xlsclients: list
client applications running on a
client by its X resource.
kxkill: kill a
xclipboard: X clipboard
client. client
start up the sgi X server as a XigS
addclient: allow remote printing
clients to connect.
a shell (command interpreter) with
C-like syntax. csh: csh
allow synchronization of the system
clock. adjtime: correct the time to
clock.
clock: analog clock in a window.
futime: control
clock and timer resolution.
cron: clock daemon.
xclock: analog / digital
clock for X.
clock: analog
clock in a window.
clock: report CPU time used.
timeslave: 'slave' local
clock to a better one.
links. link: clone a file using symbolic
STREAMS driver.
clone: open any minor device on a
ldclose, ldaclose: close a common object file.
close: close a file descriptor.
close: close a file descriptor.
fclose, flush:
close or flush a stream.
/telldir, seekdir, rewinddir,
closedir: directory operations/
closedir, diffd: directory/ readdir,
control system/ syslog, openlog,
closeup: zoom in on an image.
clear: clear i-node.
cmp: compare two files.
co: check out RCS revisions.
pixie: add profiling
code to a program.
unifdef: strip or reduce #ifdef in C
code.
col: filter reverse line-feeds.
between libraries.
collapse: look for name collisions
between libraries.
with LP. mkcentpr: register a
color Centronics-interface printer
mkcentpr(1M)
tobw: convert a
color image to black and white.
tobw(1G)
scanner: scan
color images.
scanner(1)
makemap: make the default
color map.
makemap(1G)
showmap: display the contents of the
color map.
showmap(6D)
interp: gamma-corrected
color ramp generator.
interp(6D)
scene. bounce: three
colored lights bouncing around a
loadmap: loads the
colormap from a file.
saves the current contents of the
colormap.
showmap: colormap utility.
xmap(1)
xshowmap: show
colormap.
xshowmap(1)

Si...
silicon graphics

execute: overlay shell with specified command.

time: time command.

eenv: set environment for command execution.

uxx: UNIX-to-UNIX system command execution.

system-wide csh initialization command file. cshrc:

rehash: recompute command hash table.

unhash: discard command hash table.

whois: describe what a command is.

runon: run a command on a particular cpu.

getopt: parse command options.

getoptv: parse command options.

rsh: shell, the standard/restricted command for returning a stream to a remote system.

repeat: execute command repeatedly.

system activity. time: time a command; report process data and system requests.

uxxqt: execute remote command.

rexec: return stream to a remote command.

onintr: process interrupts in command scripts.

trap: process interrupts in command scripts.

:: null command.

exec: overlay shell with specified command.

test: condition evaluation command.

accounting records. acctcms: command summary from per-process accounting.

system: issue a shell command.

test: condition evaluation command.

time: time a command.

apply: apply a command to a set of arguments.

goto: command transfer.

argument list(s) and execute commands. /overview of accounting

intro: introduction to maintenance programming/ intro: introduction to

at, batch: execute commands at a later time.

apropos: locate commands by keyword lookup.

while: repeat commands conditionally.

until: repeat commands conditionally.

else: alternative commands.

source: read commands from file.
wait for background processes to complete. wait: 
wait: await completion of process.

hypot, cabs: Euclidean distance, complex absolute value. hypot(3M)
compress, uncompress, zcat: compress and expand data. compress(1)
pack, pcat, unpack: compress and expand files. pack(1)
and expand data. compress, uncompress, zcat: compress compress(1)
calculations. vortex: display computation fluid dynamics vortex(6D)
an image file. hist compute and display the histogram of hist(6D)
calendar times INSQUE(3). compute difference between two diffitime:

mkdepend: compute header file dependencies. mkdepend(1)
entry of a common object/ ldlibindex:

sysmips: MIPS Computer Systems Inc. system call. sysmips(2)
disk driver. dks: Small Computer Systems Interface (SCSI) dks(7M)
cat: concatenate and print files. cat(1)
test: condition evaluation command. sh(1)
test: condition evaluation command. test(1)
endif: terminate conditional. csh(1)
fi: terminate conditional. sh(1)
if: conditional statement. csh(1)
if, then: conditional statement. sh(1)
while: repeat commands conditionally. csh(1)
until, while: repeat commands conditionally. sh(1)
pathconf, fpathconf: get configurable pathname variables. pathconf(2)
(POSIX). sysconf: get configurable system variables sysconf(2)
master: master configuration database. master(4)
login: login configuration file. login(4)
resolver: host-address resolver configuration file. resolver(4)
system: system configuration information table. system(4)
usconfig: semaphore and lock arena configuration operations. usconfig(3P)
chkconfig: configuration state checker. chkconfig(1M)
lboot: configure bootable kernel. lboot(1M)
parameters. ifconfig: configure network interface ifconfig(1M)
lpadmin: configure the LP spooling system. lpadmin(1M)
window and request a response. confirm: display a message in a confirm(1G)
fwtmp, wtmpfix: manipulate connect accounting records. fwtmp(1M)
allow remote printing clients to connect. addclient: addclient(1M)
socket. connect: initiate a connection on a connect(2)
getpeermode: get name of connected peer. getpeermode(2)
socketpair: create a pair of connected sockets. socketpair(2)
establish an out-going terminal line connection. dial: dial(3C)
accept: accept a connection on a socket. accept(2)
connect: initiate a connection on a socket. connect(2)
shut down part of a full-duplex connection. shutdown: shutdown(2)
listen: listen for connections on a socket. listen(2)
acctcon1, acctcon2: connect-time accounting. acctcon(1M)
lvck: check and restore consistency of logical volumes. lvck(1M)
console: console interface. console(7)
noiconlogin: select and control console login program. visuallogin, visuallogin(4)
pandora: login on the graphics console. pandora(1)
MIT X Consortium. xconsortium(1)
header for implementation-specific
math: math functions and constants. limits: file
unistd: file header for symbolic constants.
mksfs: construct a file system.
exe command. xargs: construct argument list(s) and
volume. mkfsv: construct or extend a logical
control maximum system resource consumer. getrlimit, setrlimit:
on. Utry: try to contact remote system with debugging
on. List_tape: list the contents of a given backup tape.
showsnf: print contents of an SNF file.
ls: list contents of directory.
cache. cacheflush: flush contents of instruction and/or data
showmap: display the contents of the color map.
savemap: saves the current contents of the colormap.
xev: print contents of X events.
csplit: context split.
continue: cycle in loop.
continue: cycle in loop.
mpadmin: control and report processor status.
arp: address resolution display and control. arp(1M)
schedctl: scheduler control call. schedctl(2)
ftime: control clock and timer resolution.
visuallog in, noiconlogin: select and control console login program.
ioctl: control device.
fcntl: file and descriptor control.
init, teliux: process control initialization.
consumption. getrlimit, setrlimit: control maximum system resource
icmp: Internet Control Message Protocol.
newhost: NeWS network security control. newhost(1)
nooniconlogin: login process control. nooniconlogin(5)
msgctl: message control operations. msgctl(2)
semctl: semaphore control operations.
shmctl: shared memory control operations.
usctllock: lock control operations.
usctlsema: semaphore control operations.
fcntl: file control options.
tcdrain, tcflush, tcflow: post x line control primitives #include int/ tcp control primitives
xhost: server access control program for X.
lpcl: printer control program.
timed: timed control program.
tcp: Internet Transmission Control Protocol.
swapRM, swapINX: floating-point control registers. /set_fpc led, fpc(3C)
symp: multiprocessing control.
closelog, setlogmask, vsyslog: control system log. /openlog, syslog(3B)
virtually "hangup" the current control terminal. vhangup: vhangup(2)
uadmin: administrative control.
uadmin: administrative control.
uadmin: administrative control.
uadmin: administrative control.
uadmin: administrative control.
usstat: uucp status inquiry and job control.
vc: version control.

Version 5.0 -16- April 1990
visuallogin: login process          control.      ......... visuallogin(5)
National Instruments VME IEEE-488     controller. gpib: driver for gpib(7M)
  ik: Ikon 10088 hardcopy interface controller. ik(7)
ts: ISI VME-QIC2/X cartridge tape controller. ts(7M)
xmt: Xylogics 1/2 inch magnetic tape controller. xmt(7M)
  ipi, xypi: Xylogics IPI disk      controllers and driver. ipi(7M)
  ips, dkip: Interphase disk        controllers and driver. ips(7M)
  xyl, xyl754: Xylogics disk       controllers and driver. xyl(7M)
  ethernet: IRIS-4D Series ethernet controllers. ethernet(7)
    tty: controlling terminal interface. tty(7)
    term: conventional names for terminals. term(5)
    units: conversion program. units(1)
  character classification and conversion tables. chrtbl: generate chrtbl(1M)
   white. tobw: convert a color image to black and tobw(1G)
  terminfo description. capinfo: convert a termcap description into a capinfo(1M)
    dd: convert and copy a file. dd(1M)
  long integers. 13tol, ltol3: convert between 3-byte integers and l3tol(3C)
  base-64 ASCII string. a64l, l64a: convert between long integer and a64l(3C)
    /ctime, asctime, strftime, tzset: convert date and time to string/ ctime,
      string. ecvt, fcvt, gcvt: convert floating-point number to ecvt(3C)
      scanf, fscanf, sscanf: convert formatted input. scanf(3S)
      number. strtol, atoi: convert string to double-precision strtol(3C)
      strtol, atol, atoi: convert string to integer. strtol(3C)
  network/ htnol, htnol, ntohl, ntohs: convert values between host and byteorder(3N)
  bmtoa, atomb: bitmap editor and converter utilities for X. bitmap. bitmap(1)
  X version 10 to version 11 protocol converter. x10to11: x10to11(1)
  simulator and airshow/ dog: cooperative or competitive flight dog(6D)
    dd: convert and copy a file. dd(1M)
  cpio: copy file archives in and out. cpio(1)
  cp, ln, mv: copy, link or move files. cp(1)
  rcp: remote file copy. rcp(1C)
  distcap: copy software distribution. distcap(1M)
  uulog, uuname: UNIX-to-UNIX system copy. uucp, uucp(1C)
  public UNIX-to-UNIX system file copy. uuto, upickup: uuto(1C)
  copysign, remainder, exponent/ copysign, drem, finite, logb, scalb: copysign(3M)
    copysign, remainder, exponent/ copysign, drem, finite, logb, scalb: ieee(3M)
    copysign, drem, finite, logb, scalb: copysign, remainder, exponent/ copysign(3M)
    copysign, drem, finite, logb, scalb: ieee(3M)
    ftoc: interface between prog and cord. ftoc(1)
    executable to facilitate better/ cord: rearranges procedures in an cord(1)
    terminate current process with a core dump. abort. abort(3C)
  savecore: save a core dump of the operating system. savecore(1M)
    core: format of core image file. core(4)
    mem, kmem, memm: core memory. mem(7)
    synchronization of the/ adjtime: correct the time to allow adjtime(2)
      fsin, feos, ftan, fasin/ sin, cos, tan, asin, acos, atan, atan2, trig(3M)
      hyperbolic functions. sinh, cosh, tanh, fasin, fcos, ftanh: sinh(3M)
      sum: print checksum and block count of a file. sum(1)
      wc: word count. wc(1)
      files. cp, ln, mv: copy, link or move cp(1)
### Permutated Index

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpio: format of</td>
<td>cpio archive</td>
<td>cpio(4)</td>
</tr>
<tr>
<td>cpio: copy file archives in and out</td>
<td>cpio(1)</td>
<td></td>
</tr>
<tr>
<td>cpio: format of cpio archive</td>
<td>cpio(4)</td>
<td></td>
</tr>
<tr>
<td>/transferdevice for performing cpio/transferdevice for performing cpio</td>
<td>cpioarchive(1)</td>
<td></td>
</tr>
<tr>
<td>transferdevice for performing cpio/transferdevice for performing cpio</td>
<td>cpioarchive(1)</td>
<td></td>
</tr>
<tr>
<td>interface</td>
<td>cpio: construct C to PostScript</td>
<td>cps(1)</td>
</tr>
<tr>
<td>runon: run a command on a particular cpu.</td>
<td>runon(1)</td>
<td></td>
</tr>
<tr>
<td>clock: report CPU time used.</td>
<td>clock(3C)</td>
<td></td>
</tr>
<tr>
<td>display processes having highest CPU usage in a window.</td>
<td>gr_top(1)</td>
<td></td>
</tr>
<tr>
<td>display processes having highest CPU usage.</td>
<td>top(1)</td>
<td></td>
</tr>
<tr>
<td>an existing one. creat: create a new file or rewrite</td>
<td>creat(2)</td>
<td></td>
</tr>
<tr>
<td>mkfile: create a file.</td>
<td>mkfile(1M)</td>
<td></td>
</tr>
<tr>
<td>tmpnam, tempnam: create a name for a temporary file.</td>
<td>tmpnam(3S)</td>
<td></td>
</tr>
<tr>
<td>existing one. creat: create a new file or rewrite an</td>
<td>creat(2)</td>
<td></td>
</tr>
<tr>
<td>fork: create a new process.</td>
<td>fork(2)</td>
<td></td>
</tr>
<tr>
<td>sproc: create a new share group process.</td>
<td>sproc(2)</td>
<td></td>
</tr>
<tr>
<td>taskcreate: create a new task.</td>
<td>taskcreate(3P)</td>
<td></td>
</tr>
<tr>
<td>sockeptpair: create a pair of connected sockets.</td>
<td>sockeptpair(2)</td>
<td></td>
</tr>
<tr>
<td>pcreateve, pcreatevp, pcreatevev: create a process ./pcreateve,</td>
<td>pcreateve(3C)</td>
<td></td>
</tr>
<tr>
<td>mkshlib: create a shared library.</td>
<td>mkshlib(1)</td>
<td></td>
</tr>
<tr>
<td>ctags: create a tags file.</td>
<td>ctags(1)</td>
<td></td>
</tr>
<tr>
<td>tmpfile: create a temporary file.</td>
<td>tmpfile(3S)</td>
<td></td>
</tr>
<tr>
<td>communication. socket: create an endpoint for</td>
<td>socket(2)</td>
<td></td>
</tr>
<tr>
<td>messaging C source. mkstr: create an error-message file by</td>
<td>mkstr(1)</td>
<td></td>
</tr>
<tr>
<td>pipe: create an interprocess channel</td>
<td>pipe(2)</td>
<td></td>
</tr>
<tr>
<td>admin: create and administer SCCS files.</td>
<td>admin(1)</td>
<td></td>
</tr>
<tr>
<td>makedepend: create dependencies in makefiles.</td>
<td>makedepend(1)</td>
<td></td>
</tr>
<tr>
<td>MAKEDEV: Create device special files.</td>
<td>makedev(1M)</td>
<td></td>
</tr>
<tr>
<td>of font files.. mkfontdir: create fonts.dir file from directory</td>
<td>mkfontdir(1)</td>
<td></td>
</tr>
<tr>
<td>pmake, snake: create programs in parallel.</td>
<td>pmake(1)</td>
<td></td>
</tr>
<tr>
<td>IDs. setsid: create session and set process group</td>
<td>setsid(2)</td>
<td></td>
</tr>
<tr>
<td>shell. sh: creates and specifies a window</td>
<td>sh(1G)</td>
<td></td>
</tr>
</tbody>
</table>
| umask: change or display file
| umask: change or display file
| umask: set and get file
| simulates a walking, six-legged
| cron: clock daemon.                                               | cron(1M) |
| crontab: user crontab file.                                       | crontab(1) |
| crontab: user crontab file.                                       | crontab(1) |
| cxref: generate C program cross-reference.                        | cxref(1) |
| more, page: file perusal filter for crt viewing.                   | more(1) |
| pg: file perusal filter for C RTs.                                 | pg(1) |
| crypt: encode/decode.                                             | crypt(1) |
| functions. crypt: password and file encryption                     | crypt(3X) |
| hashing encryption.                                               | crypt(3C) |
| with C-like syntax.                                               | crypt(3C) |
| cshrc: system-wide csh initialization command file.               | cshrc(4) |
| file including aliases and path
| initialization command file.                                      | cshrc(4) |
| csplit: context split.                                            | csplit(1) |

**Version 5.0**

April 1990
Permuted Index

printcap: printer capability
protocols: protocol name
data base.
the existing hostname in yp hosts
rpc: RPC program number
services: service name
store, delete, firstkey, nextkey:
dbm_error, dbm_clearerr:
terminfo: terminal capability
the existing host entry in yp hosts
diskusg: generate disk accounting
flush contents of instruction and/or
zcat: compress and expand
eval: re-evaluate shell
gview: viewer for radiosity
demograph: graphs demographic
spul, sgctl: access long integer
lock: lock process, text, or
monitor system activity
prof: analyze profile
routines M_FORK(3P), external
data representation (xdr) library
stat:

brk, sbrk: change
eval: re-evaluate shell
types: primitive system
appres: list application resource
/master:

join: relational
da terminal or query terminfo
data segment space allocation.
sh(1)
data types.
types(5)
database.
/appres:
database for use with apropos.
makewhatis(1M)
database.
master(4)
join(1)
database. tput: initialize
xdm(1)
Datagram Protocol.
udp(7P)
date and time.
gettimeofday(3B)
date and time to string/ctime
ctime,
date:
date(): print and set date.

motic: message of the
day.
o'clock: display time of
day.
date:
date(1)

date(1)

/dbm_nextkey, dbm_error,
dbm_clearerr: data base subroutines.

/dbm_delete, dbm_firstkey, dbm_open,
dbm_close, dbm_fetch, dbm_store,
/dbm_firstkey, dbm_nextkey,
/dbm_fetch, dbm_store, dbm_delete,
dbm_firstkey, dbm_nextkey,
dbm_store, dbm_delete,
/dbm_open, dbm_close, dbm_fetch,
dbm_open, dbm_delete, dbm_firstkey,
/dbm_store, dbm_delete, dbm_firstkey,
/dbm_open, dbm_close, dbm_fetch,
<table>
<thead>
<tr>
<th>Permutated Index</th>
<th>Silicon Graphics</th>
</tr>
</thead>
<tbody>
<tr>
<td>captninfo: convert a termcap</td>
<td>description into a terminfo/.....captninfo(1M)</td>
</tr>
<tr>
<td>supported character set. charset:</td>
<td>description of the standard.....charset(5)</td>
</tr>
<tr>
<td>compare or print out terminfo</td>
<td>descriptions. infocomp:.....infocomp(1M)</td>
</tr>
<tr>
<td>close: close a file descriptor.</td>
<td>close(2)</td>
</tr>
<tr>
<td>fcntl: file and descriptor control.</td>
<td>fcntl(2)</td>
</tr>
<tr>
<td>dup: duplicate an open file descriptor.</td>
<td>dup(2)</td>
</tr>
<tr>
<td>dup2: duplicate an open file descriptor.</td>
<td>dup2(2(3C)</td>
</tr>
<tr>
<td>directory, given an open file descriptor.</td>
<td>fchdir: change working directory.....fchdir(2)</td>
</tr>
<tr>
<td>descriptor given a procedure/.....ldgetpdp(3X)</td>
<td></td>
</tr>
<tr>
<td>descriptor index. /procedure.....ldgetpdp(3X)</td>
<td></td>
</tr>
<tr>
<td>/that provide access to per file</td>
<td>descriptor section of the symbol/.....stfd(3X)</td>
</tr>
<tr>
<td>INITGROUPS(3X). get descriptor table size</td>
<td>getdtablesize:</td>
</tr>
<tr>
<td>dc: desk calculator.</td>
<td>dc(1)</td>
</tr>
<tr>
<td>taskdestroy: destroy a task.</td>
<td>taskdestroy(3P)</td>
</tr>
<tr>
<td>access: determine accessibility of a file.</td>
<td>access(2)</td>
</tr>
<tr>
<td>fsyp: determine file system identifier.</td>
<td>fsyp(1M)</td>
</tr>
<tr>
<td>file: determine file type.</td>
<td>file(1)</td>
</tr>
<tr>
<td>a demo with error output directed to /dev/console. redirect: run</td>
<td>redirect(6D)</td>
</tr>
<tr>
<td>long lines for finite width output device. fold: fold</td>
<td>fold(1)</td>
</tr>
<tr>
<td>imon: inode monitor device.</td>
<td>imon(7M)</td>
</tr>
<tr>
<td>ioctl: control device.</td>
<td>ioctl(2)</td>
</tr>
<tr>
<td>devmm: device name.</td>
<td>devmm(1M)</td>
</tr>
<tr>
<td>clone: open any minor device on a STREAMS driver.</td>
<td>clone(7)</td>
</tr>
<tr>
<td>MAKEDEV: Create device special files.</td>
<td>makedev(1M)</td>
</tr>
<tr>
<td>Administration. SA: devices administered by System</td>
<td>sa(7)</td>
</tr>
<tr>
<td>communicate with generic SCSI devices. dsopen, dsclose:</td>
<td>dslib(3)</td>
</tr>
<tr>
<td>lvinit: initialize logical volume devices...</td>
<td>lvinit(1M)</td>
</tr>
<tr>
<td>devmm: device name.</td>
<td>devmm(1M)</td>
</tr>
<tr>
<td>blocks. df: report number of free disk</td>
<td>df(1)</td>
</tr>
<tr>
<td>systems. fsck, dfsg: check and repair file</td>
<td>fsck(1M)</td>
</tr>
<tr>
<td>server. dgl: Distributed Graphics Library</td>
<td>dgl(1M)</td>
</tr>
<tr>
<td>dglfax: electronic fax program.</td>
<td>dglfax(1)</td>
</tr>
<tr>
<td>running across a network. dglnewton: a physical modeling demo</td>
<td>dglnewton(6D)</td>
</tr>
<tr>
<td>running across a network. dglray: a visualized raytracer</td>
<td>dglray(6D)</td>
</tr>
<tr>
<td>terminal line connection. dial: establish an out-going</td>
<td>dial(3C)</td>
</tr>
<tr>
<td>parameters. mousewarps, dialwarps, keywarps: set input warping</td>
<td>mousewarps(6D)</td>
</tr>
<tr>
<td>bdiff: big diff.</td>
<td>bdiff(1)</td>
</tr>
<tr>
<td>directory comparator. diff: differential file and</td>
<td>diff(1)</td>
</tr>
<tr>
<td>comparison. diff3: 3-way differential file</td>
<td>diff3(1)</td>
</tr>
<tr>
<td>times INSQUE(3). compute</td>
<td>difftime:</td>
</tr>
<tr>
<td>sdiff: side-by-side difference program.</td>
<td>sdiff(1)</td>
</tr>
<tr>
<td>greyscale: make different patterns.</td>
<td>greyscale(6D)</td>
</tr>
<tr>
<td>comparator. diff: differential file and directory</td>
<td>diff(1)</td>
</tr>
<tr>
<td>diff3: 3-way</td>
<td>diff3(1)</td>
</tr>
<tr>
<td>differential file comparison.</td>
<td>differential file comparison.</td>
</tr>
<tr>
<td>send signal to executing program DIFFTIME(3C).</td>
<td>raise:</td>
</tr>
<tr>
<td>xclock: analog /</td>
<td>xclock(1)</td>
</tr>
<tr>
<td>files. mkfontdir: create fonts.</td>
<td>mkfontdir(1)</td>
</tr>
<tr>
<td>dir from directory of font</td>
<td>dir(4)</td>
</tr>
<tr>
<td>dir: format of EFS directories.</td>
<td>dir(4)</td>
</tr>
<tr>
<td>dircmp: directory comparison.</td>
<td>dircmp(1)</td>
</tr>
<tr>
<td>run a demo with error output directly to /dev/console. redirect:</td>
<td>redirect(6D)</td>
</tr>
<tr>
<td>xdrp: dump an X window</td>
<td>xdrp(1)</td>
</tr>
</tbody>
</table>

Version 5.0 —22— April 1990
ucheck: check the uucp directories and permissions file. ... ucheck(1M)
dir: format of EFS directories. ... dir(4)
install: install files in directories. ... install(1)
unlink: link and unlink files and directories. link, ... link(1M)
mkdir: make directories. ... mkdir(1)
rn, rmdir: remove files or directories. ... rm(1)
Backup: backup the specified file or directory. ... backup(1)
cd: change working directory. ... cd(1)
chdir: change working directory. ... chdir(2)
the permissions mode of a file or directory. chmod: change ... chmod(1)
chroot: change root directory. ... chroot(2)
uucleanup: uucp spool directory clean-up. ... uucleanup(1M)
diff: differential file and directory comparator. ... diff(1)
dircmp: directory comparison. ... dircmp(1)
cd: change directory. ... csh(1)
chdir: change directory. ... csh(1)
system independent/ getdents: read directory entries and put in a file ... getdents(2)
dirent: file system independent directory entry. ... dirent(4)
unlink: remove directory entry. ... unlink(2)
chroot: change root directory for a command. ... chroot(1M)
restore the specified file or directory from tape. Restore: ... restore(1)
get path-name of current working directory. getcwd: ... getcwd(3C)
descriptor. fchdir: change working directory, given an open file ... fchdir(2)
ls: list contents of directory. ... ls(1)
mkdir: make a directory. ... mkdir(2)
mvdir: move a directory. ... mvdir(1M)
pwd: working directory name. ... pwd(1)
/create fonts.dir file from directory of font files... ... mkfoni(1)
 seekdir, rewinddir, closedir, dirfd: directory operations (4.3bsd)/opendir,
 /seekdir, rewinddir, closedir: directory operations (System V). ... directory(3C)
 file. mknod: make a directory, or a special or ordinary ... mknod(2)
 getwd: get current working directory pathname. ... getwd(3C)
 rmdir: remove a directory. ... rmdir(2)
scandir, alphasort: scan a directory. ... scandir(3C)
 cd: change directory. ... sh(1)
popt: pop shell directory stack. ... csh(1)
push: push shell directory stack. ... csh(1)
about a specific semaphore DIRECTORY BSD(3B). /out information usdumpsem:
directory entry. dirent: file system independent ... dirent(4)
 /seekdir, rewinddir, closedir, dirfd: directory operations (4.3bsd)/opendir,
 names. basename, dimname: deliver portions of path ... basename(1)
system. dirview: graphical interface to file dis: disassemble an object file. ... dis(1)
enable, disassemble a MIPS instruction and ... disassemble(3X)
acct: enable or disassemble an object file. ... acct(2)
print the results. disassembler: disassemble a MIPS dis: disassemble a MIPS ... disassembler(3X)
 instruction and print the results. disassembler: disassemble a MIPS ... disassembler(3X)
unhash: discard command hash table. ... csh(1)
unset: discard shell variables. ... csh(1)
type, modes, speed, and line discipline. getty: set terminal ... getty(1M)
<table>
<thead>
<tr>
<th>Command/Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type, modes, speed, and line discipline</td>
<td>ugetty: set terminal</td>
</tr>
<tr>
<td>diskusg: generate disk accounting data by user ID.</td>
<td>diskusg(1M)</td>
</tr>
<tr>
<td>df: report number of free disk blocks.</td>
<td>df(1)</td>
</tr>
<tr>
<td>ipi, xylip: Xylitics IPI disk controllers and driver.</td>
<td>ipi(7M)</td>
</tr>
<tr>
<td>ips, dkip: Interphase disk controllers and driver.</td>
<td>ips(7M)</td>
</tr>
<tr>
<td>xyl, xyl754: Xylitics disk controllers and driver.</td>
<td>xyl(7M)</td>
</tr>
<tr>
<td>Computer Systems Interface (SCSI) disk driver. dks: Small dks(7M)</td>
<td></td>
</tr>
<tr>
<td>lv: logical volume Disk driver.</td>
<td>lv(7M)</td>
</tr>
<tr>
<td>smfd: SCSI floppy disk driver.</td>
<td>smfd(7M)</td>
</tr>
<tr>
<td>a file's in-core state with that on disk. fsync: synchronize fsync(2)</td>
<td></td>
</tr>
<tr>
<td>disks: interactive local and network</td>
<td>disks(1G)</td>
</tr>
<tr>
<td>Add_disk: add a secondary disk to the system.</td>
<td>add_disk(1)</td>
</tr>
<tr>
<td>du: summarize disk usage.</td>
<td>du(1M)</td>
</tr>
<tr>
<td>fx: disk utility.</td>
<td>fx(1)</td>
</tr>
<tr>
<td>dvhtool: modify and obtain disk volume header information.</td>
<td>dvhtool(1M)</td>
</tr>
<tr>
<td>vh: disk volume header.</td>
<td>vh(7M)</td>
</tr>
<tr>
<td>disk mounts tool. disks: interactive local and network</td>
<td>disks(1G)</td>
</tr>
<tr>
<td>data by user ID. diskusg: generate disk accounting</td>
<td>diskusg(1M)</td>
</tr>
<tr>
<td>mount, umount: mount and dismount filesystems.</td>
<td>mount(1M)</td>
</tr>
<tr>
<td>for input. manwsh: display a man page and then prompt manwsh(6D)</td>
<td></td>
</tr>
<tr>
<td>request a response. confirm: display a message in a window and inform: display a message in a window.</td>
<td>confirm(1G)</td>
</tr>
<tr>
<td>ipaste: display an image.</td>
<td>ipaste(1G)</td>
</tr>
<tr>
<td>arp: address resolution display and control.</td>
<td>arp(1M)</td>
</tr>
<tr>
<td>gamcal: visually check display calibration.</td>
<td>gamcal(6D)</td>
</tr>
<tr>
<td>calculations. vortex: display computation fluid dynamics vortex(6D)</td>
<td></td>
</tr>
<tr>
<td>xditview: display ditroff DVI files.</td>
<td>xditview(1)</td>
</tr>
<tr>
<td>vedit: screen-oriented (visual) display editor based on ex. /view, vi(1)</td>
<td></td>
</tr>
<tr>
<td>umask: change or display file creation mask.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>umask: change or display file creation mask.</td>
<td>sh(1)</td>
</tr>
<tr>
<td>xload: load average display for X.</td>
<td>xload(1)</td>
</tr>
<tr>
<td>xdpyminfo: display information utility for X.</td>
<td>xdpyminfo(1)</td>
</tr>
<tr>
<td>xdm: X Display Manager.</td>
<td>xdm(1)</td>
</tr>
<tr>
<td>cube: real-time display of famous cube puzzle.</td>
<td>cube(6D)</td>
</tr>
<tr>
<td>values. symster: meter display of system performance</td>
<td>sysmeter(1)</td>
</tr>
<tr>
<td>usage in a window. gr_top: display processes having highest CPU gr_top(1)</td>
<td></td>
</tr>
<tr>
<td>usage. top: display processes having highest CPU top(1)</td>
<td></td>
</tr>
<tr>
<td>System... xman: Manual page display program for the X Window xman(1)</td>
<td></td>
</tr>
<tr>
<td>slides: slide display program.</td>
<td>slides(6D)</td>
</tr>
<tr>
<td>xmessage: X window system message display program.</td>
<td>xmessage(1)</td>
</tr>
<tr>
<td>ulimit: change or display size limits.</td>
<td>sh(1)</td>
</tr>
<tr>
<td>demonstrates swapping buffers to display smooth animation.. swap:</td>
<td>swap(6T)</td>
</tr>
<tr>
<td>map, showmap: display the contents of the color showmap(6D)</td>
<td></td>
</tr>
<tr>
<td>file.. hist: compute and display the histogram of an image</td>
<td>hist(6D)</td>
</tr>
<tr>
<td>element analysis/ solidview: display the results of a finite solidview(6D)</td>
<td></td>
</tr>
<tr>
<td>oclcock: display time of day.</td>
<td>oclcock(1)</td>
</tr>
<tr>
<td>client applications running on a display. xlsclients: list xlsclients(1)</td>
<td></td>
</tr>
<tr>
<td>xfd: font display font for X.</td>
<td>xfd(1)</td>
</tr>
<tr>
<td>xlsfonts: server font list display font for X.</td>
<td>xlsfonts(1)</td>
</tr>
<tr>
<td>xswins: server window list display for X. xswins(1)</td>
<td></td>
</tr>
<tr>
<td>xprop: property display for X.</td>
<td>xprop(1)</td>
</tr>
</tbody>
</table>

Version 5.0 — April 1990
Silicon Graphics

xwud: image display for X ............... xwud(1)
hypot, cabs: Euclidean distance, complex absolute value .......... hypot(3M)
distcp: copy software distribution ........ distcp(1M)
dgld: Distributed Graphics Library server .......... dgld(1M)

/seed48, lcong48: generate uniformly distributed pseudo-random numbers .......... drand48(3C)
distcp: copy software distribution .......... distcp(1M)
rdist: remote file distribution program .......... rdist(1C)
xditview: display ditroff DVI files .......... xditview(1)
and driver.. ips, dkip: Interphase disk controllers .......... ips(7M)
Interface (SCSI) disk driver ....... dks: Small Computer Systems .......... dks(7M)
/res_mkquery, res_send, res_init, dn_comp, dn_expand: resolver/ resolver(3N)
/res_send, res_init, dn_comp, files .......... dn_il, dn_netman: 4DDN special .......... dn_il(7)

pretmp, / chargefee, cpacket, flight simulator and airshow/ acctsh(1M)
armchair pilot’s view of the dog: cooperative or competitive .......... dog(6D)
w: who is on and what they are doing .......... w(1)
whodo: who is doing what .......... whodo(1M)
get/set name of current domain. /setdomainname: .......... getdomainname(2)
named: Internet domain name server .......... named(1M)
done: terminate loop .......... sh(1)
strtod, atof: convert string to double-precision number .......... strtod(3C)
connection. shutdown: shut down part of a full-duplex .......... shutdown(2)
shutown: shut down system, change system state .......... shutown(1M)
script for shutting the system down... systemdow: interactive .......... systemdow(1G)
Julia sets. dragon: generates Mandelbrot and .......... dragon(6D)
link-layer protocols. drain: capture unimplemented .......... drain(7P)
mrand48, jrand48, srand48, seed48, / draw interesting patterns in an X .......... muncher(1)
window. muncher: remainder, exponent/ copysign, drem, finite, logb, scalb: copysign .......... copysign(3M)
remainder, exponent/ copysign, drem, finite, logb, scalb: copysign .......... copysign(3M)
of liquid. liquid: A faucet dripping into a multi-colored pool .......... liquid(6D)
open any minor device on a STREAMS driver. clone .......... clone(7)
Systems Interface (SCSI) disk driver. dks: Small Computer .......... dks(7M)
ds: generic (user mode) SCSI driver .......... ds(7M)
IEEE-488 controller. gpib: driver for National Instruments VME .......... gpib(7M)
hl: hardware spinlocks driver .......... hl(7M)
Xylogics IPI disk controllers and driver. .. ipi, xylipi: .......... ipi(7M)
Interphase disk controllers and driver. .. ips, dkip: .......... ips(7M)
Iv: logical volume Disk driver .......... lv(7M)
pty: pseudo terminal driver .......... pty(7M)
smfd: SCSI floppy disk driver .......... smfd(7M)
Xylogics disk controllers and driver. .. xyl, xyl754: .......... xyl(7M)
SCSI devices. dsopen, ds: generic (user mode) SCSI driver .......... ds(7M)
generic SCSI devices. dslclose: communicate with generic .......... dslib(3)
dsdopen, dsclose: communicate with generic .......... dslib(3)
du: summarize disk usage .......... du(1M)
du: on-board serial ports .......... duart(7)
dsduart(7)
symbolic information. stdump: dump a file of intermediate-code .......... stdump(1)
current process with a core dump. abort: terminate .......... abort(3C)
April 1990 -25- Version 5.0
Permut Index

xdm: dump an image of an X window.... xwm(1)
printer. xdrp: dump an X window directly to a... xdrp(1)
dumpfont: dump font out in some other format... dumpfont(1)
cell: dump memory cell log... cell(1)
od: octal dump... od(1)
nm: name list dump of MIP's object files... nm(1)
savecore: save a core dump of the operating system... savecore(1M)
specific lock USDUMPSEMA(3P), dump out information about a... usdumplock:
specific semaphore/ file. odump: dump selected parts of an object... odump(1)
xpr: print an X window dump... xpr(1)
other format. descriptor. dup: duplicate an open file... dup(2)
descriptor. dup2: duplicate an open file... dup2(3C)
dup: duplicate an open file descriptor. dup2: duplicate an open file descriptor... dup2(3C)
dup2: duplicate an open file descriptor
int tsenbreak (int filedes, int duration): /primitives #include
tcsenbreak, volume header information. dvhtool: modify and obtain disk disk... dvhtool(1M)
xditview: display ditloff DVI files... xditview(1)
vortex: display computation fluid dynamics calculations... vortex(6D)
jello: simulates nonrigid body dynamics... jello(6D)
cut: cut out selected fields of each line of a file... cut(1)
ecc: dump memory ecc log... ecc(1)
ecc: dump memory ecc log... ecc(1)
echo: echo arguments... csh(1)
echo: echo arguments... echo(1)
echo: echo arguments... csh(1)
echo: echo arguments... echo(1)
hosts. ping: send ICMP ECHO_REQUEST packets to network ping(1M)
floating-point number to string. edcvt, fcvt, gcvt: convert... edcvt(3C)
ed, red: text editor... ed(1)
_procedure_table/, end, eteet, edata, eprol; _fext; _fdata, _fbs, end(3C)
ed, red: text editor... edge(1)
cedit: edit colors on the screen... cedit(6D)
csact: print current SCSS file editing activity... cact(1)
X. bitmap, bmtoa, atobm: bitmap editor and converter utilities for... bitmap(1)
lid, uld: MIPS link editor and ucode link editor... lid(1)
screen-oriented (visual) display editor based on ex. /view, vedit:... vi(1)
ed, red: text editor... ed(1)
ex: text editor... ex(1)
xedit: simple text editor for X... xedit(1)
inged: small image editor... inged(1G)
job: a simple mouse-based text editor... jot(1G)
uld: MIPS link editor and ucode link editor... uld(1)
a.out: assembler and link editor output... a.out(4)
ld: stream editor... led(1)
users). edit: text editor (variant of ex for casual... edit(1)
whoami: print effective current user id... whoami(1)
ssetregid: set real and effective group ID... ssetregid(2)
effective user, real group, and effective group IDs. /get real user, getuid(2)

Version 5.0 April 1990
setreuid: set real and effective user ID’s. ........... setreuid(2)
/getgid, getegid: get real user, and/or effective user, real group, and/or getuid(2)
new process in a virtual memory efficient way. vfork: spawn vfork(2)
dir: format of system. EFS directories. ........... dir(4)
using full regular expressions. efs: layout of the Extent file fs(4)
dglfax: electronic fax program. dglfax(1)
/display the results of a finite element analysis program. solidview(6D)
remque: insert/remove element from a queue USGETINFO(3P). insque,
elif, else: alternative commands. sh(1)
else: alternative commands. csh(1)
elif, else: alternative commands. sh(1)
emulation. emulate_branch: MIPS branch emulate_branch(3X)
emulate_branch: MIPS branch emulation. .............. emulate_branch(3X)
xterm: terminal emulator for X. xterm(1)
psterm: NeWS terminal emulator. psterm(1)
applications that require a terminal emulator.. /utility to launch winterm(1)
printers. enable, disable: enable/disable LP enable(1)
accounting. acct: enable or disable process acct(2)
enable, disable: enable/disable LP printers. enable(1)
uuencode: format of an encoded uuencode file. uuencode(4)
transmission/ uuencode, uudecode: encode/decode a binary file for uuencode(1C)
crypt: encode/decode. ................. crypt(1)
image to be stored using run length encoding. rle: force an rle(6D)
to be stored without run length encoding. verbatim: force an image verbatim(6D)
encryption. crypt, setkey, encrypt: generate hashing encrypt(3C)
setkey, encrypt: generate hashing encryption. crypt, encrypt(3C)
crypt: password and file encryption functions. crypt(3X)
makekey: generate encryption key. makekey(1)
_fdata, _fbss, _procedure_table,/end, etext, edata, eprol, _ftext, end(3C)
logout: end session. ................. csh(1)
end: terminate loop. ................. csh(1)
/setgrgid, getgrnam, setgrent, endgrent, fgetgrent: get group file/ getgrent(3C)
entry. /gethostent, sethostent, endhostent, herror: get network host gethostbyname(3N)
setgrent(3C)endif: terminate conditional. csh(1)
inventory/ getinvent, setinvent, endinvent, scaninvent: get hardware getinvent(3)
/getnetbyname, setnetent, endnetent: get network entry. getnetent(3N)
socket: create an endpoint for communication. socket(2)
/getprotobyname, setprotonet, endprotoent: get protocol entry. getprotoent(3N)
file/ getpwuid, getpwnam, setpwent, endpwent, fgetpwent: get password getpwent(3C)
/set servbyname, setservent, endservent: get service entry. getservent(3N)
endsw: terminate switch. csh(1)
/endutline, pututline, setutent, endutent, utmpname: access utmp file/ getut(3C)
launch: graphical utility to enter arguments and invoke commands. launch(1)
getdents: read directory entries and put in a file system/ getdents(2)
reference manuals; find manual entries by keyword. /the on-line man(1)
list: get entries from name list. nlist(3X)
manuals; find manual/ man: print entries from the on-line reference man(1)
linewidth: line number entries in a MIPS object file. linenum(4)
/ a visual interface for selecting entries in the workspace transfer/ transformer(1G)
flditem: manipulate line number entries of a common object file/ ldldread(3X)
Permutated Index

entries of a section of a common/
entries of a section of a common
entry. dirent: ....... dirent(4)
entry formats: ....... utmp(4)
entry. /getnamam, setgrent,
grent, fgetgrent: get group file
entry. /gethostent, sethostent,
endhostent, herror: get network host
entry. /setinvent, endinvent,
scaninvent: get hardware inventory
entry. /getnetbyaddress, getnetbyname,
setnetent, endnetent: get network
entry. /getprotobase, getprotobyname,
setprotoent, endprotoent: get protocol
entry. /setprotoent, endprotoent,
fgetpwent: get password file
entry. /setpwent, endpwent,
getpwnambyname: get RPC
entry. /getpwnambyname,
setservent, endservent: get service
entry. /getservbyname,
endutent, utmpname: access utmp file
entry. /pututline, pututent,
lgetaux_x: retrieve an auxiliary
entry in yp hosts data base.
/remove the existing host
name for object file symbol table
/compute the index of a symbol table
/read an indexed symbol table
exctenent: extract FORTRAN-callable
putpwent: write password file
execute:
unlink: remove directory
entry. ....... unirlink(2)
profile: setting up an
setenviron: set variable in
env: set
getenv: return value for
putenv: print out the
commands performed for multi-user
export: add shell variables to the
exportonews: Pass a login shell's
generate a string for the NEWSERVER
unsetenviron: remove
end, etext, edata,
deroff: remove rroff/roff, tbl, and
eqn constructs.
hosts. equiv: list of trusted hosts.
rand48, srand48, seed48, / rand48,
complementary error function.
erf, cerfc: error function and
erf: complementary error function.
erf, cerfc: error function and
erf: error function and complementary
error messages. perror, sterror,
error function. erf, cerfc:
erf: error function and complementary
klog: kmem
source. mkstr: create an
help: ask for help about SCCS
errno, sys_errno, sys_nerr: system
introduction to system calls and
oserror, setoserror: get/set system
error.
<table>
<thead>
<tr>
<th><strong>Silicon Graphics</strong></th>
<th><strong>Permutted Index</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>redirect: run a demo with error output directed to/ redirect(6D)</td>
<td></td>
</tr>
<tr>
<td>spellin, spellout: find spelling errors. spell, spell(1)</td>
<td></td>
</tr>
<tr>
<td>connection. dial: esc: terminate case. sh(1)</td>
<td></td>
</tr>
<tr>
<td>_fbss, <em>procedure_table/ end</em> establish an out-going terminal line dial(3C)</td>
<td></td>
</tr>
<tr>
<td>setmnt: establish mount table. setmnt(1M)</td>
<td></td>
</tr>
<tr>
<td>ethernet: IRIS-4D Series etext, edata, eprol, _fdata, _ftext, end(3C)</td>
<td></td>
</tr>
<tr>
<td>controllers. ethernet controllers. ethernet(7)</td>
<td></td>
</tr>
<tr>
<td>value. hypot, cabs: Euclidean distance, complex absolute hypot(3M)</td>
<td></td>
</tr>
<tr>
<td>eval: re-evaluate shell data. csh(1)</td>
<td></td>
</tr>
<tr>
<td>eval: re-evaluate shell data. sh(1)</td>
<td></td>
</tr>
<tr>
<td>expr: evaluate arguments as an expression. expr(1)</td>
<td></td>
</tr>
<tr>
<td>test: condition evaluation command. sh(1)</td>
<td></td>
</tr>
<tr>
<td>test: condition evaluation command. test(1)</td>
<td></td>
</tr>
<tr>
<td>history: print history event list. csh(1)</td>
<td></td>
</tr>
<tr>
<td>journaldigest: 4Sight event record and playback toolchest. journaldigest(1W)</td>
<td></td>
</tr>
<tr>
<td>journalrecord, journalend: 4Sight event recording and playback from an/ journalplay(1)</td>
<td></td>
</tr>
<tr>
<td>xev: print contents of X xev(1)</td>
<td></td>
</tr>
<tr>
<td>edit: text editor (variant of ex for casual users). edit(1)</td>
<td></td>
</tr>
<tr>
<td>(visual) display editor based on ex: text editor. ex(1)</td>
<td></td>
</tr>
<tr>
<td>lpq: spool queue examination program. lpq(1)</td>
<td></td>
</tr>
<tr>
<td>SGI- /signal set manipulation and examination routines (POSIX, with sigsetops(3)</td>
<td></td>
</tr>
<tr>
<td>TCSETATTR(3T). floating-point exception handler package handle_sigfpe:</td>
<td></td>
</tr>
<tr>
<td>USDUMPLOCK(3P). uspinfo: exchange information though an arena uspinfo,</td>
<td></td>
</tr>
<tr>
<td>command: exec: overlay shell with specified csh(1)</td>
<td></td>
</tr>
<tr>
<td>command. exec: overlay shell with specified sh(1)</td>
<td></td>
</tr>
<tr>
<td>execlp, execvp: execute a file. exec, execv, exec, execvp, exec(2)</td>
<td></td>
</tr>
<tr>
<td>exec: a file. exec, execvp, exec, execvp, exec, execvp, exec(2)</td>
<td></td>
</tr>
<tr>
<td>exec, execvp, exec, execvp, exec, execvp, exec, execvp, exec(2)</td>
<td></td>
</tr>
<tr>
<td>identifying number. tag: tag a MIPS executable or shell script with an tag(1)</td>
<td></td>
</tr>
<tr>
<td>cord: rearranges procedures in an executable to facilitate better/ cord(1)</td>
<td></td>
</tr>
<tr>
<td>exec, execve, execvp, execvp, exec, execvp, exec, execvp, exec(2)</td>
<td></td>
</tr>
<tr>
<td>repeat: execute command repeatedly. csh(1)</td>
<td></td>
</tr>
<tr>
<td>construct argument list(s) and execute command. xargs: xargs(1)</td>
<td></td>
</tr>
<tr>
<td>at, batch: execute commands at a later time. at(1)</td>
<td></td>
</tr>
<tr>
<td>say: execute PostScript. say(1)</td>
<td></td>
</tr>
<tr>
<td>regexp, regex: compile and execute regular expression. regexp(3X)</td>
<td></td>
</tr>
<tr>
<td>uuxqt: execute remote command requests. uuxqt(1M)</td>
<td></td>
</tr>
<tr>
<td>send signal to executing program DIFFTIME(3C). raise:</td>
<td></td>
</tr>
<tr>
<td>env: set environment for command execution. env(1)</td>
<td></td>
</tr>
<tr>
<td>sleep: suspend execution for an interval. sleep(1)</td>
<td></td>
</tr>
<tr>
<td>sleep: suspend execution for interval. sleep(3)</td>
<td></td>
</tr>
<tr>
<td>pixstats: analyze program execution. pixstats(1)</td>
<td></td>
</tr>
<tr>
<td>monstartup, moncontrol: prepare execution profile. monitor, monitor(3X)</td>
<td></td>
</tr>
<tr>
<td>rexecd: remote execution server. rexecd(1M)</td>
<td></td>
</tr>
<tr>
<td>profil: execution time profile. profil(2)</td>
<td></td>
</tr>
<tr>
<td>uux:UNIX-to-UNIX system command execution. uux(1C)</td>
<td></td>
</tr>
<tr>
<td>execvp: execute a file. exec, file. exec, exec, exec, exec, exec(2)</td>
<td></td>
</tr>
<tr>
<td>execvp: execute a file. exec, exec, execvp: execute a exec(2)</td>
<td></td>
</tr>
<tr>
<td>execvp: execute a file. exec, exec, execvp: execute a exec(2)</td>
<td></td>
</tr>
<tr>
<td>April 1990</td>
<td>Version 5.0</td>
</tr>
</tbody>
</table>
base. unregisterhost: remove the existing host entry in yp hosts data . unregisterhost(3N)
base. renamehost: rename the existing hostname in yp hosts data . renamehost(3N)
create a new file or rewrite an

fexp, fexpm1, flog, flog10, flog1p/

uncompress, zcat: compress and
glob: filename

distribute data. compress,

pack, pcat, unpack: compress and

expand a filesystem .

growfs

an image... imgexp:

expand the range of pixel values in .

imgexp(6D)

fexpm1, flog, flog10, flog1p/

expand files .

pack(1)

expression.

regroup: regular expression;

expression compile and match .

regexp(5)

expr: evaluate arguments as an

expression compile .

regcmp(1)

regexp: compile and execute regular

expression handler .

regex(3B)

for a pattern using full regular

expression. regcmp .

regcmp(3X)

entry points from a C file.

execute: extract FORTRAN-callable .

exitcentry(1)

mkv: construct or

extend a logical volume .

mkv(1M)

/a shell script specification for

efs: layout of an

Extant file system .

fs(4)

inode: format of an

Extant File System inode .

inode(4)

library routines M_FORK(3P).

external data representation (xdr) .

xdr:

points from a C file. exitcentry.

extract FORTRAN-callable entry .

exitcentry(1)

implement shared strings. xstr:

extract strings from C programs to .

xstr(1)

floor, ffloor, ceil, fceil, fmod,

fabs, rint, trunc, ftrunc: floor .

floor(3M)

procedures in an executable to

facilitate better cache mapping .

cord(1)

signal: simplified software signal

facilities (4.3BSD) .

signal(3B)

4.3bsd software signal

facilities GETDTABLESIZE(3) .

sigvec:

introduction to networking

facilities: networking .

neintro(7)

sigaction: software signal

facilities (POSIX) .

sigaction(2)

report inter-process communication

facilities status. ipcs .

ipcs(1)

of figures under X11. xfig:

facilities (System V) .

signal(2)

bindkey: function key binding

facility for use with .

bindkey(1)

.atan2, fsin, fcos, ftan, fasin,

facos, fatan, atan2: trigonometric .

trig(3M)

factor: obtain the prime factors of

factor(1)

loginlog: log of

failed login attempts .

login(4)
attempts to acquire a semaphore, and
true, fails if not possible. uscsemna:....usc-semna(3P)
bldfamily: build font family: provides true(1)
inet: Internet protocol family. inet(7)
raw: raw network protocol family. raw(7P)
cube: real-time display of famous cube puzzle. cube(6D)
data in a machine-independent fashion. /sgetl: access long integer sputl(3X)
acos, atan, atan2, fsin, fc0s, ftan, fast family, facos, fastan, fastan2: /asinf astan(3M)
realloc, calloc, malloc, mallocf: fast main memory allocator. /free, malloc(3X)
/fc0s, ftan, fasin, facos, fastan, fastan2: trigonometric /atan2, trig(3M)
/facos, ftan, fasin, facos, fastan, fastan2: trigonometric functions and / trig(3M)
pool of liquid. liquid: A faucet dripping into a multi-colored liquid(6D)
dglfax: electronic fax program. dgglfax(1)
etext, edata, eprol, _fext, _fdata, _fbs, _procedure_table, end, _fbs, _procedure_table, end, end(3C)
frunc: floor/ floor, ffloor, cell, fceil, fmod, fabs, rint, trunc, floor(3M)
given an open file descriptor. fchdir: change working directory, fchdir(2)
chmod, chmod: change mode of file. fchmod(2)
file (System V and 4.3BSD). chown, chmod, chown: change owner and group of file chown(2)
stream, fchown: change owner and group of a file. fclosel(3S)
/fclose: flush close or flush a file. fclosel(3S)
fcntl: file and descriptor control. fcntl(2)
fcntl: file control options. fcntl(5)
<table>
<thead>
<tr>
<th>Action</th>
<th>File Access</th>
<th>Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>utime</td>
<td></td>
<td>file access and modification times.</td>
<td>utime(2)</td>
</tr>
<tr>
<td>ldfcn</td>
<td></td>
<td>file access routines.</td>
<td>ldfcn(4)</td>
</tr>
<tr>
<td>access</td>
<td></td>
<td>determine accessibility of a file.</td>
<td>access(2)</td>
</tr>
<tr>
<td>fcntl</td>
<td></td>
<td>file and descriptor control.</td>
<td>fcntl(2)</td>
</tr>
<tr>
<td>diff</td>
<td></td>
<td>differential file and directory comparator.</td>
<td>diff(1)</td>
</tr>
<tr>
<td>epio</td>
<td></td>
<td>file archives in and out.</td>
<td>epio(1)</td>
</tr>
<tr>
<td>rcs</td>
<td></td>
<td>change RCS file attributes.</td>
<td>rcs(1)</td>
</tr>
<tr>
<td>mkstr</td>
<td></td>
<td>create an error message file by massaging C source.</td>
<td>mkstr(1)</td>
</tr>
<tr>
<td>check</td>
<td></td>
<td>check RCS status of a file.</td>
<td>check(1)</td>
</tr>
<tr>
<td>pwck, grpck</td>
<td></td>
<td>password/group file checkers.</td>
<td>pwck(1M)</td>
</tr>
<tr>
<td>chmod, fchmod</td>
<td></td>
<td>change mode of file.</td>
<td>chmod(2)</td>
</tr>
<tr>
<td>diff3</td>
<td></td>
<td>3-way differential file comparison.</td>
<td>diff3(1)</td>
</tr>
<tr>
<td>fcntl</td>
<td></td>
<td>control options.</td>
<td>fcntl(5)</td>
</tr>
<tr>
<td>rcp</td>
<td></td>
<td>remote file copy.</td>
<td>rcp(1C)</td>
</tr>
<tr>
<td>uupick</td>
<td></td>
<td>public UNIX-to-UNIX system file copy. uuto.</td>
<td>uuto(1C)</td>
</tr>
<tr>
<td>core</td>
<td></td>
<td>format of core image file.</td>
<td>core(4)</td>
</tr>
<tr>
<td>umask</td>
<td></td>
<td>change or display file creation mask.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>umask</td>
<td></td>
<td>change or display file creation mask.</td>
<td>sh(1)</td>
</tr>
<tr>
<td>umask</td>
<td></td>
<td>set and get file creation mask.</td>
<td>umask(2)</td>
</tr>
<tr>
<td>crontab</td>
<td></td>
<td>user crontab file.</td>
<td>crontab(1)</td>
</tr>
<tr>
<td>source</td>
<td></td>
<td>read commands from file.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>csh</td>
<td></td>
<td>initialization command file.</td>
<td>cshrc: system-wide cshrc(4)</td>
</tr>
<tr>
<td>ctags</td>
<td></td>
<td>create a tags file.</td>
<td>ctags(1)</td>
</tr>
<tr>
<td>selected fields of each line of a file</td>
<td>cut: cut out</td>
<td>cut(1)</td>
<td></td>
</tr>
<tr>
<td>dd</td>
<td></td>
<td>convert and copy a file.</td>
<td>dd(1M)</td>
</tr>
<tr>
<td>make a delta</td>
<td></td>
<td>(change) to an SCCS file.</td>
<td>delta(1)</td>
</tr>
<tr>
<td>mkdepend</td>
<td></td>
<td>compute header file dependencies.</td>
<td>mkdepend(1)</td>
</tr>
<tr>
<td>close</td>
<td></td>
<td>close a file descriptor.</td>
<td>close(2)</td>
</tr>
<tr>
<td>dup</td>
<td></td>
<td>duplicate an open file descriptor.</td>
<td>dup(2)</td>
</tr>
<tr>
<td>dup2</td>
<td></td>
<td>duplicate an open file descriptor.</td>
<td>dup2(3C)</td>
</tr>
<tr>
<td>working directory, given an open file</td>
<td>fchdir: change</td>
<td>fchdir(2)</td>
<td></td>
</tr>
<tr>
<td>/routines that provide access to per file descriptor section of the file</td>
<td>stdf(3X)</td>
<td>stdf(3X)</td>
<td></td>
</tr>
<tr>
<td>dis</td>
<td></td>
<td>disassemble an object file.</td>
<td>dis(1)</td>
</tr>
<tr>
<td>rdist</td>
<td></td>
<td>file distribution program.</td>
<td>rdist(1C)</td>
</tr>
<tr>
<td>sact</td>
<td></td>
<td>print current SCCS file editing activity.</td>
<td>sact(1)</td>
</tr>
<tr>
<td>crypt</td>
<td></td>
<td>password and encryption functions.</td>
<td>crypt(3X)</td>
</tr>
<tr>
<td>endgrent</td>
<td></td>
<td>get group file entry.</td>
<td>getgrent(3C)</td>
</tr>
<tr>
<td>endpwent, fgetpwent</td>
<td></td>
<td>get password file entry.</td>
<td>getpwent(3C)</td>
</tr>
<tr>
<td>endutent, utmpname</td>
<td></td>
<td>access utmp file entry.</td>
<td>getut(3C)</td>
</tr>
<tr>
<td>putpwent</td>
<td></td>
<td>write password file entry.</td>
<td>putpwent(3C)</td>
</tr>
<tr>
<td>freelvent</td>
<td></td>
<td>get lvtab file entry.</td>
<td>freelvent</td>
</tr>
<tr>
<td>execve, execdp, execvp</td>
<td></td>
<td>execute a file.</td>
<td>exec(2)</td>
</tr>
<tr>
<td>entry points from a C</td>
<td></td>
<td>file /extract FORTRAN-callable</td>
<td>extcentry(1)</td>
</tr>
<tr>
<td>remove an advisory lock on an open file</td>
<td>flock: apply or</td>
<td>flock(3B)</td>
<td></td>
</tr>
<tr>
<td>fgrep</td>
<td></td>
<td>search a file for a character string.</td>
<td>fgrep(1)</td>
</tr>
<tr>
<td>grep</td>
<td></td>
<td>search a file for a pattern.</td>
<td>grep(1)</td>
</tr>
<tr>
<td>regular/egrep</td>
<td></td>
<td>search a file for a pattern using full</td>
<td>egrep(1)</td>
</tr>
<tr>
<td>proto</td>
<td></td>
<td>prototype job file for at.</td>
<td>proto(4)</td>
</tr>
<tr>
<td>ldaopen</td>
<td></td>
<td>open a common object file for reading. ldaopen</td>
<td>ldaopen(3X)</td>
</tr>
<tr>
<td>aliases</td>
<td></td>
<td>aliases file for sendmail.</td>
<td>aliases(4)</td>
</tr>
</tbody>
</table>
Silicon Graphics

/uudecode: encode/decode a binary file for transmission via mail...
acct: per-process accounting file format...
ar: archive (library) file format...
pnch: file format for card images...
intro: introduction to file formats...
mkfontdir: create fonts.dir file from directory of font files...
number entries of a common object file function. /manipulate line
get: get a version of an SCCS file.
group: group membership file.
implementation-specific/ limits: file header for
filehdr:
unistd: file header for MIPS object files.
unistd:
ldfthread: read the file header of a common object file.
Idohseek: seek to the optional file header of a common object file.
display the histogram of an image file... hist: compute and
a part of the screen in an image file... icut: save
for the symbol table definition file in archives. /access routine
only) which: locate a program file including aliases and path (csh)
split: split a file into pieces
header of a member of an archive file. ldahread: read the archive
ldaclose: close a common object file. ldclose,
the file header of a common object file. ldfthread: read
of a section of a common object file. /seek to line number entries
file header of a common object file. /seek to the optional
of a section of a common object file. /seek to relocation entries
section header of a common object file. /read an indexed/named
section of a common object file. /seek to an indexed/named
table entry of a common object file. /compute the index of a symbol
symbol table of a common object file. /read an indexed symbol
line number entries in a MIPS object file. lnumen:
link: link to a file.
loadmap: loads the colormap from a file.
login: login configuration file.
passmgmt: password file management.
merge: three-way file merge.
mkfifo: make a FIFO special file.
mkfile: create a file.
directory, or a special or ordinary file. mknod: make a
ctermid: generate file name for terminal.
mkttemp, mktstemp: make a unique file name.
the database for the mail aliases file. newaliases: rebuild
newform: change the format of a text file.
nul: the null file.
dump selected parts of an object file. odump:
information. stdump: dump a file of intermediate-code symbolic
stdout: find the slot in the umtp file of the current user.
Backup: backup the specified file or directory.
change the permissions mode of a file or directory. chmod:
Restore: restore the specified file or file structure.
fuser: identify processes using a file or file structure.

April 1990

Version 5.0
mknod: build special file or named pipe (FIFO).
create: create a new file or rewrite an existing one.
password: password file.
files or subsequent lines of one file. /merge same lines of several
more, page: file personal filter for crn viewing.
pg: file personal filter for CRTs.
seek, rewind, ftell: reposition a file pointer in a stream.
Iseek: move read/write file pointer (System V and 4.3BSD).
pr: print an SCCS file.
at/batch/cron queue description file. queuedefs:
remove a file RAISE(3C).
read: read from file.
information for a common object file. reloc: relocation
rename: change the name of a file.
host-address resolver configuration file. resolver:
rmdev: remove a delta from an SCCS file.
bf: big file scanner.
compare two versions of an SCCS file. scsdiff:
scsfile: format of an SCCS file.
section header for a MIPS object file. scndir:
format of curses screen image file.. scr_dump:
a part of the screen in an image file. scrsave: save
: read commands from file.
showdns: print contents of an SNF file.
print the section sizes of an object file. size:
an portion of the screen in an image file. snapshot: save
stat, lstat, fstat: get file status.
in an object, or other binary file. /find the printable strings
identify processes using a file or file structure. fuser:
print checksum and block count of a file. sum:
retrieve symbol name for object file symbol table entry. Ildgetname: Ildgetname(3X)
symlink: make symbolic link to a file.
sys_id: system identification file.
ckbupsd: check file system backup schedule.
DBG, debug: the debug file system.
dirview: graphical interface to file system.
efs: layout of the Extent file system.
afs: layout of the file system.
fs: identifier.
entry. direct: file system independent directory
/read directory entries and put in a file system independent format.
statfs, fstatfs: get file system information.
inode: format of an Extent File System inode.
mkfs: construct a file system.
mount: mount a file system.
rmount: get file system statistics.
fsstat: report file system status.
mount: mounted file system table.
syfs: get file system type information.
unmount: unmount a file system.
fchown: change owner and group of a file (System V and 4.3BSD). chown.
<table>
<thead>
<tr>
<th>Silicon Graphics</th>
<th>Permutated Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>WorkSpace: graphical interface</td>
<td>workspace(1G)</td>
</tr>
<tr>
<td>to file system.</td>
<td></td>
</tr>
<tr>
<td>fsck, dsfsck: check and</td>
<td>fsck(1M)</td>
</tr>
<tr>
<td>repair file systems.</td>
<td></td>
</tr>
<tr>
<td>labelit: provide labels for</td>
<td>labelit(1M)</td>
</tr>
<tr>
<td>file systems.</td>
<td></td>
</tr>
<tr>
<td>umountall: mount, umount</td>
<td>mountall(1M)</td>
</tr>
<tr>
<td>multiple file systems. moun-</td>
<td></td>
</tr>
<tr>
<td>tail: deliver the last part of</td>
<td>tail(1)</td>
</tr>
<tr>
<td>a file.</td>
<td></td>
</tr>
<tr>
<td>term: format of compiled term</td>
<td>term(4)</td>
</tr>
<tr>
<td>file.</td>
<td></td>
</tr>
<tr>
<td>utimes: set file times.</td>
<td>utimes(3B)</td>
</tr>
<tr>
<td>tmpfile: create a temporary</td>
<td>tmpfile(3S)</td>
</tr>
<tr>
<td>file.</td>
<td></td>
</tr>
<tr>
<td>create a name for a temporary</td>
<td>tmnpnam(3S)</td>
</tr>
<tr>
<td>file.</td>
<td></td>
</tr>
<tr>
<td>truncate, fttruncate: truncate</td>
<td>truncate(2)</td>
</tr>
<tr>
<td>a file to a specified length.</td>
<td></td>
</tr>
<tr>
<td>routeprint: route file to</td>
<td>routeprint(1)</td>
</tr>
<tr>
<td>printer.</td>
<td></td>
</tr>
<tr>
<td>access and modification times</td>
<td>touch(1)</td>
</tr>
<tr>
<td>of a file.</td>
<td></td>
</tr>
<tr>
<td>ftp: Internet file transfer</td>
<td>ftp(1C)</td>
</tr>
<tr>
<td>program.</td>
<td></td>
</tr>
<tr>
<td>tftp: trivial file transfer</td>
<td>tftp(1C)</td>
</tr>
<tr>
<td>program.</td>
<td></td>
</tr>
<tr>
<td>tftp: Internet File Transfer</td>
<td>tftp(1M)</td>
</tr>
<tr>
<td>Protocol server.</td>
<td></td>
</tr>
<tr>
<td>system: uucico: file transport</td>
<td>uucico(1M)</td>
</tr>
<tr>
<td>program for the uucp.</td>
<td></td>
</tr>
<tr>
<td>uusched: the scheduler for the</td>
<td></td>
</tr>
<tr>
<td>uucp file transport program.</td>
<td>uusched(1M)</td>
</tr>
<tr>
<td>ftw: walk a file tree.</td>
<td>ftw(3C)</td>
</tr>
<tr>
<td>tlink: clone a file tree using</td>
<td>tlink(1)</td>
</tr>
<tr>
<td>symbolic links.</td>
<td></td>
</tr>
<tr>
<td>file: determine file type.</td>
<td>file(1)</td>
</tr>
<tr>
<td>checking utility for use with</td>
<td></td>
</tr>
<tr>
<td>file type rules. isSuper:</td>
<td>isSuper(1)</td>
</tr>
<tr>
<td>supertype.</td>
<td></td>
</tr>
<tr>
<td>undo a previous get of an SCCS</td>
<td>unget(1)</td>
</tr>
<tr>
<td>file. unget:</td>
<td></td>
</tr>
<tr>
<td>uniq: report repeated lines in</td>
<td>uniq(1)</td>
</tr>
<tr>
<td>a file.</td>
<td></td>
</tr>
<tr>
<td>xauth: X authority file utility.</td>
<td>xauth(1)</td>
</tr>
<tr>
<td>the uucp directories and</td>
<td></td>
</tr>
<tr>
<td>permissions format of an encoded</td>
<td></td>
</tr>
<tr>
<td>uucode file.</td>
<td>uuencode(4)</td>
</tr>
<tr>
<td>val: validate SCCS file.</td>
<td>val(1)</td>
</tr>
<tr>
<td>write: write on a file.</td>
<td>write(2)</td>
</tr>
<tr>
<td>umask: set file-creation mode</td>
<td>umask(1)</td>
</tr>
<tr>
<td>mask.</td>
<td></td>
</tr>
<tr>
<td>files. filehdr: file header for</td>
<td>filehdr(4)</td>
</tr>
<tr>
<td>MIPS object.</td>
<td></td>
</tr>
<tr>
<td>glob: filename expand argument</td>
<td>csh(1)</td>
</tr>
<tr>
<td>list.</td>
<td></td>
</tr>
<tr>
<td>ferror, feof, clearerr,</td>
<td>ferror(3S)</td>
</tr>
<tr>
<td>search and print process</td>
<td></td>
</tr>
<tr>
<td>accounting file(s). acctcom:</td>
<td>acctcom(1)</td>
</tr>
<tr>
<td>merge or add total accounting</td>
<td></td>
</tr>
<tr>
<td>files. acctmerge:</td>
<td>acctmerge(1M)</td>
</tr>
<tr>
<td>admin: create and administer</td>
<td>admin(1)</td>
</tr>
<tr>
<td>SCCS files.</td>
<td></td>
</tr>
<tr>
<td>link, unlink: link and unlink</td>
<td>link(1M)</td>
</tr>
<tr>
<td>files and directories.</td>
<td></td>
</tr>
<tr>
<td>cat: concatenate and print</td>
<td>cat(1)</td>
</tr>
<tr>
<td>files.</td>
<td></td>
</tr>
<tr>
<td>cmp: compare two files.</td>
<td>cmp(1)</td>
</tr>
<tr>
<td>or reject lines commen to two</td>
<td></td>
</tr>
<tr>
<td>sorted files.</td>
<td>comm(1)</td>
</tr>
<tr>
<td>cp, ln, mv: copy, link or move</td>
<td>cp(1)</td>
</tr>
<tr>
<td>files.</td>
<td></td>
</tr>
<tr>
<td>dn_ssl, dn_netman: 4DNN special</td>
<td>dn_ssl(7)</td>
</tr>
<tr>
<td>files.</td>
<td></td>
</tr>
<tr>
<td>filehdr: file header for MIPS</td>
<td>filehdr(4)</td>
</tr>
<tr>
<td>object files.</td>
<td></td>
</tr>
<tr>
<td>find: find files.</td>
<td>find(1)</td>
</tr>
<tr>
<td>fspec: format specification in</td>
<td>fspec(4)</td>
</tr>
<tr>
<td>text files.</td>
<td></td>
</tr>
<tr>
<td>ident: identify files.</td>
<td>ident(1)</td>
</tr>
<tr>
<td>install: install files in</td>
<td>install(1)</td>
</tr>
<tr>
<td>directories.</td>
<td></td>
</tr>
<tr>
<td>disk. fsync: synchronize a file</td>
<td></td>
</tr>
<tr>
<td>'s in-core state with that on</td>
<td>fsync(2)</td>
</tr>
<tr>
<td>intro: introduction to special</td>
<td>intro(7)</td>
</tr>
<tr>
<td>files.</td>
<td></td>
</tr>
<tr>
<td>information of a list of image</td>
<td>istat(6D)</td>
</tr>
<tr>
<td>files. istat: print the header</td>
<td></td>
</tr>
<tr>
<td>lockf: record locking on files.</td>
<td>lockf(3C)</td>
</tr>
</tbody>
</table>
Permutated Index Silicon Graphics

MAKEDEV: Create device special files. makedev(1M)
file from directory of font files. mkfondir: create fonts.dir mkfondir(1)
nm: name list dump of MIPS object files. nm(1)
rm, rmdir: remove files or directories. rm(1)
paste: merge same lines of several files or subsequent lines of one/paste(1)
pr: print files. pack(1)
and other information about RCS files. rlog: print log messages rlog(1)
sort: sort and/or merge files. sort(1)
what: identify SCCS files. what(1)
xditview: display ditoff DVI files. xditview(1)
growfs: expand a filesystem. growfs(1M)
fstab: static information about filesystems. fstab(4)
mount, umount: mount and dismount filesystems. mount(1M)
bstream: many buffered filter. bstream(1)
more, page: file perusal filter for crt viewing. more(1)
pg: file perusal filter for CRTs. pg(1)
ln: line numbering filter. nl(1)
col: filter reverse line-feeds. col(1)
/find files. find(1)
/find: find files. find(1)
/find: find files. find(1)
/from the on-line reference manuals;
/ttyname, isatty: find name of a terminal. ttyname(3C)
library. lorder: find ordering relation for an object. lorder(1)
spell, spellin, spellout: find spelling errors. spell(1)
object, or other binary/strings: find the printable strings in an
the current user. tyslot: find the slot in the utmp file of
program. finger: user information lookup finger(1)
server. finger: remote user information finger(1M)
solidview: display the results of a
remainder, exponent / copysign, drem,
remainder, exponent / copysign, drem,
fold: fold long lines for
head: give first few lines. head(1)
setup: initialize system for first user. setup(1)
dbmminit, fetch, store, delete,
xbiff: mailbox flag for X. xbiff(1)
set: set shell flags or positional parameters. sh(1)
flight: simulate the flight of any of several aircraft.
of several aircraft. flight: simulate the flight of any
flight simulator and airshow/flight(6D)
dog: cooperative or competitive
/set_fpc_led, swapRM, swapINX: flip: spin one or more objects. flip(6D)
package TCSETATTR(3T). floating-point control registers. fpc(3C)
ecvt, fvct, gvct: convert floating-point exception handler handle_sigfpe:
ldexp, modf: manipulate parts of floating-point numbers. frexp
fp_class: classes of IEEE floating-point values. fp_class(3C)
lock on an open file. flock: apply or remove an advisory flock(3B)
/log10, log1p, pow, exp, expm1, log10, flog10, flog1p: exponential,
/log1p, pow, exp, expm1, flog10, flog10, flog1p: exponential,
/pow, exp, expm1, flog, flog10, flog1p: exponential, logarithm,

Version 5.0 April 1990
Silicon Graphics

/filemod, fabs, rint, trunc, ftrunc: floor, ceiling, remainder, absolute/
/fabs, rint, trunc, ftrunc: floor/
/smifd: SCSI
/cflow: generate C
data cache, cacheflush:
/floor, /floor, ffloor, ceil, fceil,
/floor (3M)
/floor (3M)
/floor (3M)
/smifd (7M)
cflow(1)
vortex: display computation
/fclose, fflush: close or
/fluid dynamics calculations.
/fclose (3S)
data cache, cacheflush:
/flush a stream.
/flush contents of instruction and/or
/flyray: a visualized raytracer.
/flyray (6D)
/flyray (6D)
/xfl: font display for X.
/bldfamily: build
/fonts, dir file from directory of
/font family description.
/bldfamily (1)
/xfl (1)
/xfl (1)
/xfl (1)
/font list display for X.
/xfont (1)
/xfont (1)
fold: fold long lines for finite
/fold (1)
/fold (1)
/bdfftsosn: BDF to SNF
/bdfftsosn (1)
/font compiler for X11.
/font compiler for X11.
/font compiler for X11.
/font compiler for X11.
/fopen, freopen, fopen, open a
/fopen (3S)
/fopen (3S)
runtime encoding. rle:
/force an image to be stored using
/force an image to be stored without
/force an image to be stored without
/force an image to be stored without
/fetch: loop over list of names.
/fetch: loop over list of names.
/acct: per-process accounting file
/correct: create a new process.
/acct (4)
/fork (2)
ar: archive (library) file
/format. ar(4)
dumpfont: dumpfont
/format. dumpfont:
/dumpfont (1)
dumpfont (1)
pch: file
/format for card images.
pch (4)
dumpfont (1)
and put in a file system independent
/format. /read directory entries
/getdents (2)
gameform: change the
/format of a text file.
gameform (1)
/uencode: format of an encoded uencode file.
uencode (4)
/inode, ino: format of an Extent File System
/inode (4)
/term: format of compiled term file.
term (4)
core: format of core image file.
core (4)
cpio: format of cpio archive.
cpio (4)
scr_dump: format of curses screen image file.
scr_dump (4)
dir: format of EPS directories.
dir (4)
rcsfile: format of RCS file.
r dirs (4)
scsfile: format of SCCS file.
scsfile (4)
set the default monitor video output
/format. setmon:
/setmon (1)
fpsec: format specification in text files.
fpsec (4)
intro: introduction to file
/intro (4)
ulm, wtmp: utmp and wtmp entry
/utmp (4)
scanf, fscanf, scanf: convert
/scanf (3S)
scanf (3S)
vpprintf, vprintf, vprintf: print
/vprintf (3S)
print, fprintf, printf: print
/print (3S)
print (3S)
fmt: simple text
/fmt (1)
fmt (1)
mkf2c: generate
/FORTRAN-C interface routines.
mkf2c (1)
C file, extcency: extract
/FORTRAN-callable entry points from a
/extcency (1)

April 1990

Version 5.0
variables. pathconf, fpargc, get configurable pathname. pathconf(2)
get_fpc_irr, get_fpc_csr, floating-point values. fp_class: classes of IEEE fp_class(3C)
output. printf, fprintf, fprintf: print formatted printf(3S)
on a stream. putc, putcchar, putc, putw: put character or word putc(3S)
puts. puts: put a string on a stream. puts(3S)
freed, fwrite: binary input/output. fwrite(3S)
usfrellock: free a lock. usfrellock(3P)
usfreesema: free a semaphore. usfreesema(3P)
df: report number of free disk blocks. df(1)
malloc: free, realloc, calloc: main memory malloc(3C)
malinfo: fast main memory/ malloc, free, realloc, calloc, malloc, malloc(3X)
barrier, new_barrier, init_barrier, barrier: barrier functions. barrier(3P)
REMOVE(3C). freelist: get lvtab file entry freelist
usvsem: frees a resource to a semaphore. usvsem(3P)
foopen, fopen, fdopen: open a stream. fopen(3S)
of floating-point numbers. frexp, ldexp, modf: manipulate parts frexp(3C)
input. scanf, fscanf, scanf: convert formatted scanf(3S)
system. fsck, df, fsck: check and repair file fsck(1M)
file pointer in a stream. fseek, rewind, ftell: reposition a fseek(3S)
/cos, tan, asin, acos, atan, atan2, fsin, fcos, tan, fasin, facos, / trig(3M)
sinh, cosh, tanh, fsinh, fcosh, f sinh, sinh, f cos, f cos
functions. sinh, cosh, tanh, tray, facos, / trig(3M)
files. fspec: format specification in text fspec(4)
sqrt, sqrt, asqrt, sqrt: cube root, square root. sqrt(3M)
fsstat: report file system status. fsstat(1M)
filesystems. fstab: static information about fstab(4)
stat, lstat, fs: get file status. stat(2)
information. fstatfs: get file system fstatfs(2)
identifier. fsyp: determine file system fsyp(1M)
state with that on disk. fsync: synchronize a file's in-core fsync(2)
/asin, acos, atan, atan2, fsin, fcos, tan, fasin, facos, f cos, / trig(3M)
sinh, cosh, tanh, fsinh, fcosh, f sinh, f cos, f cos
a stream. fseek, rewind, ftell: reposition a file pointer in ftell(3S)
end, etext, edata, epro, _next, _data, _fbs / end(3C)
resolution. ftimer: control clock and itimer ftimer(1)
cord. ftoc: interface between prof and ftoc(1)
communication package. ftoc: standard interprocess stdtpc(3C)
front: Internet file transfer program. ftp(1C)
Protocol server. ftp: Internet File Transfer ftpd(1M)
/ceil, fm, fabs, int, trunc, specified length. truncate, ftfunc: floor, ceil, remainder, floor(3M)
trunc: truncate a file to a truncate(2)
ftw: walk a file tree. ftw(3C)
search a file for a pattern using full regular expressions. egrep egrep(1)
shutdown: shut down part of a full-duplex connection. shutdown(2)
the dogfight. shadow: full-screen armchair pilot's view of shadow(6D)
function. erf, erfc: error function and complementary error erf(3M)
function and complementary error erf(3M)
gamma: log gamma function. gamma(3M)
use with. bindkey: function key binding facility for bindkey(1)
entries of a common object file function. /manipulate line number ldread(3X)
timed sleep and processor yield function. sginap sginap(2)
Silicon Graphics

math: math functions and constants. ............ math(5)
/lacos, atan, atan2: trigonometric functions and their inverses. ....... trig(3M)
acosh, atanh: inverse hyperbolic functions. asinh, ............ asinh(3M)
init_barrier, free_barrier: barrier functions. barrier, new_barrier, ...... barrier(3P)
j0, j1, yn, y0, y1, yn: bessel functions. .................. bessel(3M)
crypt: password and file encryption functions. .................. crypt(3X)
nearest integer, and truncation functions /abs value, ............ floor(3M)
introduction to mathematical library functions. math: ............ math(3M)
to /a a high-level interface to basic things. stfe(3X)
fsinh, fsinh, ftanh: hyperbolic functions. sinh, cosh, tanh, .......... sinh(3M)
for extending the Workspace menu functions /script specification .......... transferdevice(4)
file or file structure. fuser: identify processes using a .......... fuser(1M)
arena: a future sport. ............ arena(6D)
fread, fwrite: binary input/output. ............ fread(3S)
accounting records. fwtmp, wtmpinx: manipulate connect .......... fwtmp(1M)
fx: disk utility. .................. fx(1)
calibration. gamcal: visually check display ............ gamcal(6D)
puzzle: puzzle game for X. .................. puzzle(1)
tintro: introduction to games and demos. ............ intro(6)
gamma: log gamma function. ............ gamma(3M)
or set the gamma value stored in /. gamma. gamma: get .......... gamma(6D)
get or set the gamma value stored in /.gamma. gamma: get or set the gamma value .......... gamma(6D)
stored in /.gamma. gamma: get or set the gamma value .......... gamma(6D)
gamma: log gamma function. ............ gamma(3M)
gamma value stored in /.gamma. gamma: get or set the gamma value .......... gamma(6D)
generator. interp: gamma-corrected color ramp .......... interp(6D)
gated: gateway routing daemon. ............ gated(1M)
gated: gateway routing daemon. ............ gated(1M)
/print_unaligned_summary: gather statistics on unaligned/ ............ unaligned(3X)
write output gathered from buffers GETRUSAGE(3). writev: ............ writev(2)
geclear: clear IRIS graphics screen. ............ gelear(1G)
ecvt: convert floating-point number ............ ecvt(3C)
environment/ setnewhost, sn: generate a string for the NEWSERVER ............ setnewhost(1)
cflow: generate C flowgraph. .................. cflow(1)
cxref: generate C program cross-reference. ............ cxref(1)
and conversion tables. chrbl: generate character classification ....... chrbl(1M)
user ID. diskusg: generate disk accounting data by ............ diskusg(1M)
makekey: generate encryption key. ............ makekey(1)
cermitd: generate file name for terminal. ............ cermitd(3S)
routines. mkf2c: generate FORTRAN-C interface ............ mkf2c(1)
crypt, setkey, encrypt: generate hashing encryption. ............ crypt(3C)
lptest: generate lineprinter ripple pattern. ............ lptest(1)
ncheck: generate path names from i-numbers. ............ ncheck(1M)
tasks. lex: generate programs for simple lexical ............ lex(1)
/jrand48, srand48, seed48, icong48: generate uniformly distributed/ ............ rand(48)(3C)
dragon: generates Mandelbrot and Julia sets. ............ dragon(6D)
xfig: Facility for Interactive Generation of figures under X11. .......... xfig(1)
flight simulator and airshow generator. /or competitive ............ dog(6D)
interp: gamma-corrected color ramp generator. ............ interp(6D)
rand, srand: simple random-number generator. ............ rand(3C)
/setstate: better random number generator; routines for changing/ ............ random,
Permutted Index

generator; routines for changing
dsopen, dsclose: communicate with
ds: generic SCSI devices.
character or word from a stream.
or word from a stream. getc, getwchar, fgetc, getw: get
gwchar, fgetwc, getchar: get character
working directory.
put in a file system independent/
get/set name of current domain.
4.3bsd software signal facilities
user, real/ getuid, geteuid, getgid,
name.
user, effective user, real/ getuid,
getfpc_rir, get_fpc_eir/ fpc,
swapINX/ /set_fpc_eir, get_fpc_eir,
fpcc, get_fpc_eir, set_fpc_eir,
effective user/ getuid, geteuid,
setgrent, endgrent, fgetsgrent/ get;
endgrent, fgetsgrent/ get/ getgrent,
getgrent: get/ getgrent, getgrgid,
getgrgid: return value for environment
getenv: return value for environment
getenv: return value for environment
groups: get group access list.
initialize group access list
sethostent/ /gethostbyname,
gethostent, gethostent, gethostent, /gethostent,
gethostent, hostent, /gethostent,
gethostent, /gethostbyname, gethostbyname,
gethostbyname, gethostbyname, /gethostbyname,
identifer of current host.
names of current host.
host machine swap_.*() - swap the/
scaninvent: get hardware inventory/
of interval timer.
termios *termios_p, speed_1 speed;
setnetent, endnetent/ getnetent,
etnetbyname, setnetent, endnetent,
etnetbyname, setnetent, endnetent/
argument vector.
getopt: get option letter from
getopt: parse command options.
getopts, getoptcv: parse command options.
getopts, getoptcv: parse command options.
getopt: get option letter from
getoptcv: parse command options.
getpass: get system page size.
getpeername: get name of connected
getpeername: get name of connected
getppgid, getppid: get process,
getppid: get process, getppid:
program scheduling priority.
getpriority, setpriority: get/set
getprotoent, getprotobynumber,
getprotobyname, getprotobynumber,
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
getprotobyname, setprotoent/
setprotoent/ getprotoent,
Silicon Graphics

Permutated Index

fgetpwent: get/ getpwent, getpwuid, endpwent, fgetpwent: get/ getpwent, maximum system resource/
RPC entry, getpcent, getpbyname, getpbynumber: get RPC entry, getpbynumber: get RPC entry.
tcflow (int filedes, int action); write output gathered from buffers stream.
getservent, getservbyport, getprotobyname, setprotobyname, getprotobynumber, setprotobynumber.
gettimeofday, settimeofday: get/set date and time.
primitives/ tcesetgrp: posix
getdomainname, setdomainname: get/set name of current domain.
gethostname, sethostname: get/set name of current host.
getpriority, setpriority: get/set system error.
#include int/ tcesetattr: posix
host. gethostid, gethostid:
getitimer, setitimer: get/set value of interval timer.
getsockname: get socket name.
getsockopt, setsockopt: get and set
gettimeofday, settimeofday: get/set
gettimeofday, settimeofday: date and time.
speed and terminal settings used by speed, and line discipline.
cr spawn.
settings used by getty.
get real user, effective user, real/ pututline, setutent, endutent/ setutent, endutent/.
getutent, getutid, getutid, getutid:
getutid, getutline, getutline, getutline,
getutline, pututline,
stream. getc, getchar, fgetc,
pathname.
madvise:
give advise about handling memory.
head: give first few lines.
/retrieve procedure descriptor retrieve an auxiliary entry, fchdir: change working directory.
List_tape: list the contents of a
glob: filename expand argument list.
strtime, tzset: convert/ localtime,
gmtime, asctime, ctime, asctime,
goto: command transfer.
_setjmp, _longjmp: non-local program. \_procedure_string_table,
 Instruments VME IEEE-488/
dirview: WorkSpace:
 and invoke commands. launch:
t3270: Silicon Graphics 3270 interface card.

getpwnam, setpwent, endpwent, getpwnam, setpwent, endpwent, getpwnam, setpwent, endpwent, limit, setlimit: control
getpbyname, getpbynumber: get
getpbynumber: get RPC entry.
getpbynumber, get RPC entry.
getpbyname, getpbynumber:
getpbynumber, get RPC entry.
getpbyname, getpbynumber, get
getpbynumber, get RPC entry.
getpbyname, getpbynumber, get

gettimeofday, settimeofday: get/set date and time.
getset foreground process group.
getset name of current domain.
get/set name of current host.
get/set program scheduling priority.
get/set system error.
set/get terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
get/set terminal type, modes.
get/set terminal state primitives.
get/set unique identifier of current
get/set value of interval timer.
Silicon Graphics

gview: viewer for radiosity data.  
halt: halt the system.  

halt: halt the system.  

powerdown: stop all processes and 
ansi_tape: ANSI standard tape 

halt the system.  

handlertape floating-point exception 

handlertape parallel programming primitives 

handlertape print_unaligned_summary: gather/ 
curses: terminal screen 

handlertape handling and optimization package. 
curses(3X) 

handlertape _toupper, setclass: character 

handlertape handling. /toascii, _tolower, 

cyrtype(3C) 

handlertape madvise: give advise about 

handlertape handling memory. 

madvise(2) 
xmh: X interface to the MH message 

clearing. 

hangups: virtually 

"hangup" the current control 

Hangups(2) 
nohup: run a command immune to 

hangups and quits. 

nohup(1) 
nohup: run command immune to 

hangups.  

nohup(1) 

ik: Ikon 10088 

hardcopy interface controller. 

ik(7) 

hinv: hardware inventory command. 

hinv(1M) 

/endview, scanview: get 

hardware inventory entry. 

getinv(3) 

hl: hardware spinlocks driver. 

hl(7M) 

hsearch, hcreate, hdestroy: manage 

hash search tables.  

hsearch(3C) 

rehash: recompute command 

hash table.  

csh(1) 

unhash: discard command 

hash table.  

csh(1) 

crypt, setkey, encrypt: generate 

hashing encryption.  

crypt(3C) 

window. gr_top: display processes 

having highest CPU usage in a 

gr_top(1) 

top: display processes 

having highest CPU usage. 

top(1) 

search tables. hsearch, 
hsearch, hcreate, 
hdestroy: manage hash hash 

hsearch(3C) 

head: give first few lines.  

head(1) 

mkdepend: compute 

header file dependencies. 

mkdepend(1) 

cshdr: section 

header for a MIPS object file.  

cshdr(4) 

constants. limits: file 

header for implementation-specific 

limits(4) 

filehdr: file 

header for MIPS object files.  

filehdr(4) 

unistd: file 

header for symbolic constants. 

unistd(4) 

modify and obtain disk volume 

header information. dvhtool: 

dvhtool(M) 

image files.. istat: print the 

header information of a list of 

istat(6D) 

prtvtoc: print volume 

header information. 

prtvtoc(1M) 

ldthread: read the file 

header of a common object file.  

ldthread(3X) 

lddseek: seek to the optional file 

header of a common object file. 

lddseek(3X) 

/read an indexed/named section 

header of a common object file. 

lddread(3X) 

file. ldahread: read the archive 

header of a member of an archive 

lddread(3X) 

vh: disk volume 

header.  

vh(7M) 

commands. help: ask for 

help about SCCS error messages and 

help(1) 

help: ask for help about SCCS error 

help(1) 

/gethostent, sethostent, endhostent, 

error: get network host entry. 

gethostbyname(3N) 

gtop: display processes having 

highest CPU usage in a window. 

gtop(1) 

top: display processes having 

highest CPU usage. 

top(1) 

sfe: routines that provide a 

high-level interface to basic/ 

sfe(3X) 

hist: compute and display the 

hist(6D) 

histogram of an image file.. 

histogram of an image file. 

hist(6D) 

April 1990  

Version 5.0
history: print
    history event list. ....... csh(1)
    history: print history event list. ....... csh(1)
hl: hardware spinlocks driver. ....... hl(7M)
ntho1, ntho<: convert values between
    host and network byte order. /hton1, byteorder(3N)
endhostent, error: get network
    host entry. /gethostent, s ethostent, gethostbyname(3N)
unregisterhost: remove the existing
    host entry in yp hosts data base. unregisterhost(3N)
get/set unique identifier of current
    host. gethostid, s ethostid: gethostid(2)
sethostname: get/set name of current
    host. gethostname, gethostname(2)
gethostsex: get the byte sex of the
    host machine swap - *0 - swap the sex/
    host name data base. host name(4)
    hosts: host name resolution description. hostn(5)
runtime: show
    host status of local machines. runtime(1C)
set or print identifier of current
    host system. hostid: hostid(1)
set or print name of current
    host system. hostname: hostname(1)
sethostresorder: specify order of
    file. resolver: sethostresorder(3N)
current host system.
    hostid: set or print identifier of
    hostname: host name resolution
    description. hostname(5)
renamehost: rename the existing
    host name in yp hosts data base. renamehost(3N)
current host system.
    hostname: set or print name of
    hostname(1)
rhosts: list of trusted
    hosts and users. rhosts(4)
rename the existing hostname in yp
    hosts data base. renamehost(3N)
remove the existing host entry in yp
    hosts data base. unregisterhost(3N)
hosts: host name data base. hosts(4)
hosts.equiv: list of trusted
    hosts. hosts.equiv: list of trusted hosts. hosts.equiv(4)
    hosts.equiv: list of trusted hosts. hosts.equiv(4)
    hosts: ping: send ping(1M)
    uptime: show how long system has been up. uptime(1)
    hash search tables.
    hsearch, hcreate, hdestroy: manage hsearch(3C)
    values between host and network/
    between host and network/ hton1,
    hton1, hton2, nton1, nton2: convert
    hy: HyperNet interface. hy(7)
    asinh, acosh, atan3: inverse
    hyperbolic functions. asinh(3M)
cosh, tanh, fsinh, fcosh, ftanh:
    hyperbolic functions. sinh, sinh(3M)
    hy: HyperNet interface. hy(7)
hyroute: set the
    HyperNet routing tables. hyroute(1M)
complex absolute value.
    hypot, cabs: Euclidean distance, hypot(3M)
tables. hyroute: set the HyperNet routing
    /tablet reader daemon for Bitpad
    I compatible tablet/digitizers. %tablet(1M)
    ical: calendar. ical(1G)
hosts. ping: send
    ICMP ECHO REQUEST packets to network. ping(1M)
    Protocol.
    icmp: Internet Control Message. icmp(7P)
polyhedron.
    ico: animate an icosahedron or other ico(1)
ico: animate an
    icosahedron or other polyhedron. ico(1)
an image file.
    icut: save a part of the screen in icut(6D)
disk accounting data by user
    id. diskusg: generate. diskusg(1M)
semaphore set or shared memory
    id. ipcm: remove a message queue, ipcm(1)
    names. id: print user and group IDs and id(1)
setpgid: set process group
    ID. setpgid(2)
set real and effective group
    ID. setregid(2)
BSDsetpgp: set process group
whoami: print effective current user
c newave: real-time simulation of an
simulation of the surface of an
sys_id: system
descriptor. sysinfo: print system
descriptor. fs typ: determine file system
descriptor. gethostid, sethostid: get/set unique
descriptor. hostid: set or print
descriptor. shmget: get shared memory segment
descriptor. sysid: return system
descriptor. ident: identify files.
descriptor. file structure. fuser:
descriptor. what:
descriptor. uname:
descriptor. executable or shell script with an
autologin: set autologin user
identity. ident: identity of user
process group, and parent process
real group, and effective group
setegid, setrgid: set user and group
set real and effective user
create session and set process group
setuid, setegid: set user and group
fp_class: classes of IEEE floating-point values.
driver for National Instruments VME
IEEE-488 controller. gpib:
if: conditional statement.
to acquire a semaphore, and fails
interface parameters.
unifdef: strip or reduce
controller.
controller. ik: 
closeup: zoom in on an
xwd: 
imged: small
core: format of core
and display the histogram of an
save a part of the screen in an
scr_dump: format of curses screen
save a part of the screen in an
save a portion of the screen in an
the header information of a list of
the range of pixel values in an
mapimg: translates a screen
ipaste: display an
iset: set the type of an
izoom: magnify or shrink an
da screen image into an RGB
ID (System V and 4.3BSD). setpgp,
setpgp(2)
whoami()
n newave(1D)
idealized waterbed. wave: real-time
wave(6D)
ident(1)
sys_id(4)
sysinfo(1)
fs typ(1M)
g ethostid(2)
h ostitd(1)
shmget(2)
syssid(3C)
ident(1)
fuser(1M)
what(1)
uname(1)
tag(1)
a utologin(4)
ID s and names.
id(1)
getpgid(2)
getuid(2)
seteguid, setegrid(3C)
seteguid(2)
setsid(2)
setsid(2)
fp_class(3C)
gpib(7M)
ifconfig: configure network
unifdef(1)
Ik on 10088 hardcopy interface
Ik on 10088 hardcopy interface
image.
image display for X.
image editor.
image file.
image file. hist: compute
image file. icut:
sr c_dump(4)
scrsave(6D)
image file. snapshot:
image file. istat: print
image.. imgexp: expand
image into an RGB image.
ipaste(1G)
image.
imaged(6D)
imaged(1G)
mapimg(1G)
mapimg(1G)
Permutated Index

winicons: stowed window................. winicons(SW)
  xwd: dump an..................... xwd(1)
  roting: maps an.................. roting(6D)
  encoding, rle: force an......... encoding(6D)
  length encoding, verbatim: force an
  tobw: convert a color............(1G)
  ipaint: Paint using bitmap....... ipaint(6D)
  pnch: file format for card....... pnch(4)
  scanner: scan color.............. scanner(1)
    the make utility.

values in an image...
  nohup: run a command............. nohup(1)
  nohup: run command.............. nohup(1)

extract strings from C programs to
  limits: file header for
  sysmips: MIPS Computer Systems
  syssgi: Silicon Graphics
  xnt: Xylogics 1/2
  /get/set terminal state primitives
  foreground process group primitives
  /posix line control primitives
  termios / /posix baud rate primitives
    only), which: locate a program file
  fsync: synchronize a file's
    dirent: file system
    entries and put in a file system
    an auxiliary entry, given an
    given a procedure descriptor
    common / libdindex: compute the
    /strnmp, strcspn, strtok, strstr,
    common object / ldthread: read an
    common / ldsread, ldthread: read an
    ldsseek, ldsnseek: seek to an
    terminals. last:
    inet_makeaddr, inet_ifaof,/
    address / inet_ifaof, inet_makeaddr,
    inet_addr, inet_network, inet_ntoa,
    inet_makeaddr, inet_ifaof,/
    inet_makeaddr, inet_ifaof,
    inet_makeaddr, inet_addr,
    inet_addr, inet_network,
    terminfo descriptions.
    window.
  USDUMPSEMA(3P), dump out
  semaphore / dump out
  fstab: static
  lvtab: information about logical volumes.
  rlog: print log messages and other

image mechanism.
  image of an X window.
  image onto a surface.
  image to be stored using run length
  image to be stored without run
  image to black and white.
  images as brushes.
  images.
  images.
  images.
  images.
  make: C preprocessor interface to
  imaged: small image editor.
  imgexp: expand the range of pixel
  immune to hangups and quits.
  immune to hangups.
  imon: inode monitor device.
  implement shared strings. xstr:
  implementation-specific constants.
  Inc. system call.
  Inc. system call.
  inch magnetic tape controller.
    #include int tegetattr (int fildes,/
    #include int tegetpgrp (int / get/set
    #include int tcsendbreak (int /
    #include speed_t cfgetospeed (struct
    including aliases and path (csh
    in-core state with that on disk.
    independent directory entry.
    independent format /read directory
    index. ldgetaux: retrieve
    index. /procedure descriptor
    index of a symbol table entry of a
    index, rindex: string operations.
    indexed symbol table entry of a
    indexeded named section header of a
    indexed/named section of a common/
    indicate last logins of users and
    inet: Internet protocol family.
    inet_addr, inet_network, inet_ntoa,
    inet_makeaddr, inet_ifaof,/
    inet_ifaof, inet_netof: Internet
    inet_makeaddr, inet_ifaof,/
    inet_makeaddr, inet_ifaof,
    inet_makeaddr, inet_addr,
    inet_addr, inet_network,
    terminfo descriptions.
    window.
  information about a specific lock
  information about a specific
  information about filesystems.
  information about logical volumes.
  information about RCS files.

Version 5.0

April 1990
<table>
<thead>
<tr>
<th>Silicon Graphics</th>
<th>Permutated Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>utilization RANDOM(3B). get modify and obtain disk volume header</td>
<td>information about resource getrunspace: dvhtool: dvhtool(1M)</td>
</tr>
<tr>
<td>file. reloc: relocation</td>
<td>information for a common object reloc(4)</td>
</tr>
<tr>
<td>finger: user</td>
<td>information lookup program. finger(1)</td>
</tr>
<tr>
<td>lpsstat: print LP status</td>
<td>information. lpsstat(1)</td>
</tr>
<tr>
<td>files.. istat: print the header</td>
<td>information. istat(6D)</td>
</tr>
<tr>
<td>pac: printer/plotter accounting</td>
<td>information. pac(1M)</td>
</tr>
<tr>
<td>prtvtoc: print volume header</td>
<td>information. prtvtoc(1M)</td>
</tr>
<tr>
<td>rpcinfo: report RPC</td>
<td>information. rpcinfo(1M)</td>
</tr>
<tr>
<td>fingerd: remote user</td>
<td>information server. fingerd(1M)</td>
</tr>
<tr>
<td>statfs, fstatfs: get file system</td>
<td>information. statfs(2)</td>
</tr>
<tr>
<td>a file of intermediate-code symbolic</td>
<td>information. stdump: dump stdump(1)</td>
</tr>
<tr>
<td>sysfs: get file system type</td>
<td>information. sysfs(2)</td>
</tr>
<tr>
<td>system: system configuration</td>
<td>information table. system(4)</td>
</tr>
<tr>
<td>USDUMPLOCK(3P). usputinfo: exchange</td>
<td>information though an arena usputinfo, xdpynfo(1)</td>
</tr>
<tr>
<td>xdpynfo: display</td>
<td>information utility for X. xdpynfo(1)</td>
</tr>
<tr>
<td>xwininfo: window</td>
<td>information utility for X. xwininfo(1)</td>
</tr>
<tr>
<td>initat: script for the initialization. init process. init(4)</td>
<td></td>
</tr>
<tr>
<td>initialization. init, telinit: process control init(1M)</td>
<td></td>
</tr>
<tr>
<td>functions. barrier, new_barrier, routines for changing generators</td>
<td>init_barrier, free_barrier: barrier barrier(3P)</td>
</tr>
<tr>
<td>get descriptor table size INTGROUPS(3). /number generator;</td>
<td>random, INTGROUPS(3X). getdtablesize:</td>
</tr>
<tr>
<td>access list (berkeley 4.3 version) INTGROUPS_BSD(3B). set group</td>
<td>setgroups:</td>
</tr>
<tr>
<td>network: network initialization and shutdown script. network(1M)</td>
<td></td>
</tr>
<tr>
<td>cshrc: system-wide csh</td>
<td>initialization command file. cshrc(4)</td>
</tr>
<tr>
<td>init, telinit: process control initialization. init(1M)</td>
<td></td>
</tr>
<tr>
<td>brc, bcheckrc: system initialization procedures. brc(1M)</td>
<td></td>
</tr>
<tr>
<td>usinit, _utrace: semaphore and lock</td>
<td>initialization routine. usinit(3P)</td>
</tr>
<tr>
<td>tset: terminal dependent initialization. tset(1)</td>
<td></td>
</tr>
<tr>
<td>terminfo database. tpt: initialize a terminal or query tpt(1)</td>
<td></td>
</tr>
<tr>
<td>4.3 version) READV(3C). initialize group access list (bsd</td>
<td>initgroups:</td>
</tr>
<tr>
<td>CFGETOSPEED(3T).</td>
<td>initialize group access list initgroups:</td>
</tr>
<tr>
<td>GETGROUPS(3B).</td>
<td>initialize group access list initgroups:</td>
</tr>
<tr>
<td>lvinit:</td>
<td>initialize logical volume devices. lvinit(1M)</td>
</tr>
<tr>
<td>setup:</td>
<td>initialize system for first user. setup(1)</td>
</tr>
<tr>
<td>xinit: X Window System</td>
<td>initialver. xinit(1)</td>
</tr>
<tr>
<td>usinitlock:</td>
<td>initializes a lock. usinitlock(3P)</td>
</tr>
<tr>
<td>usnewlock: allocates and initializes a lock. usnewlock(3P)</td>
<td></td>
</tr>
<tr>
<td>usinitsem: initializes a semaphore. usinitsem(3P)</td>
<td></td>
</tr>
<tr>
<td>usnewsmsa: allocates and initializes a semaphore. usnewsmsa(3P)</td>
<td></td>
</tr>
<tr>
<td>connect:</td>
<td>connect a connection on a socket. connect(2)</td>
</tr>
<tr>
<td>popen, pclose:</td>
<td>pipe to/from a process. popen(3S)</td>
</tr>
<tr>
<td>number generator; routines/ random, process.</td>
<td>initstate, setstate: better random random,</td>
</tr>
<tr>
<td>ctri: clear i-node. ctri(1M)</td>
<td></td>
</tr>
<tr>
<td>System inode. inode: format of an Extent File inode(4)</td>
<td></td>
</tr>
<tr>
<td>format of an Extent File System inode. inode: inode(4)</td>
<td></td>
</tr>
<tr>
<td>imon: inode monitor device. imon(7M)</td>
<td></td>
</tr>
<tr>
<td>read: accept input from the standard input. sh(1)</td>
<td></td>
</tr>
<tr>
<td>a man page and then prompt for input. mansh: display mansh(6D)</td>
<td></td>
</tr>
<tr>
<td>fscanf, sscanf: convert formatted input. scanf, scanf(3S)</td>
<td></td>
</tr>
</tbody>
</table>

April 1990 Version 5.0
<table>
<thead>
<tr>
<th>Silicon Graphics</th>
<th>Permutated Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>muncher: draw</td>
<td>muncher(1)</td>
</tr>
<tr>
<td>audio: bi-directional audio channel</td>
<td>audio(7)</td>
</tr>
<tr>
<td>ftc:</td>
<td>ftc(1)</td>
</tr>
<tr>
<td>Silicon Graphics 5080 workstation</td>
<td>gse(7)</td>
</tr>
<tr>
<td>t3270: Silicon Graphics 3270</td>
<td>t3270(7)</td>
</tr>
<tr>
<td>console: console</td>
<td>console(7)</td>
</tr>
<tr>
<td>ik: Ikon 10088 hardcopy</td>
<td>ik(7)</td>
</tr>
<tr>
<td>cps: construct C to PostScript</td>
<td>cps(1)</td>
</tr>
<tr>
<td>transfermanager: provide a visible names. xfontsel: point &amp; click</td>
<td>transfermanager(1G)</td>
</tr>
<tr>
<td>demos: buttonfly: a pretty user</td>
<td>xfontsel(1)</td>
</tr>
<tr>
<td>hy: HyperNet</td>
<td>buttonfly(6D)</td>
</tr>
<tr>
<td>klog: kernel error logging</td>
<td>hy(7)</td>
</tr>
<tr>
<td>mtio: magnetic tape</td>
<td>klog(7)</td>
</tr>
<tr>
<td>ifconfig: configure network</td>
<td>mtio(7)</td>
</tr>
<tr>
<td>plp: parallel line printer</td>
<td>ifconfig(1M)</td>
</tr>
<tr>
<td>mkf2c: generate FORTRAN-C</td>
<td>plp(7)</td>
</tr>
<tr>
<td>dks: Small Computer Systems</td>
<td>mkf2c(1)</td>
</tr>
<tr>
<td>a compilation unit symbol table</td>
<td>Interface (SCSI) disk driver.</td>
</tr>
<tr>
<td>swap: swap administrative</td>
<td>dks(7M)</td>
</tr>
<tr>
<td>/routines that provide a high-level administration. sysadm: menu</td>
<td>Interface (SCSI) disk driver.</td>
</tr>
<tr>
<td>dirview: graphical WorkSpace: graphical</td>
<td>/routines that provide</td>
</tr>
<tr>
<td>imake: C preprocessor</td>
<td>interfaces. termio, termios:</td>
</tr>
<tr>
<td>system. xmh: X</td>
<td>interfaces to file system.</td>
</tr>
<tr>
<td>/that provide a binary read/write</td>
<td>interfaces to file system.</td>
</tr>
<tr>
<td>telnet: User</td>
<td>interfaces to the make utility.</td>
</tr>
<tr>
<td>tps: SCSI 1/4-inch Cartridge tape</td>
<td>xmh(1)</td>
</tr>
<tr>
<td>tty: controlling terminal general System V and POSIX terminal</td>
<td>interface to the MH message handling</td>
</tr>
<tr>
<td>staux: routines that provide scalar information. stdump: dump a file of</td>
<td>interface to the MIPS symbol table.</td>
</tr>
<tr>
<td>xslatoms: list registerinmethost: allocate</td>
<td>stio(3X)</td>
</tr>
<tr>
<td>routines. /inet_inaof, inet_netof: bootp: server for</td>
<td>telnet(1C)</td>
</tr>
<tr>
<td>bootp: server for</td>
<td>interface.</td>
</tr>
<tr>
<td>icmp:</td>
<td>tps(7M)</td>
</tr>
<tr>
<td>named:</td>
<td>interface.</td>
</tr>
<tr>
<td>ftp:</td>
<td>tty(7)</td>
</tr>
<tr>
<td>server. ftpd:</td>
<td>interfaces. termio, termios:</td>
</tr>
<tr>
<td>inet:</td>
<td>interfaces to auxiliaries.</td>
</tr>
<tr>
<td>ip:</td>
<td>staux(3X)</td>
</tr>
<tr>
<td>mailq: send mail over the</td>
<td>intermediate-code symbolic.</td>
</tr>
<tr>
<td>inetd:</td>
<td>stdump(1)</td>
</tr>
<tr>
<td>telnetd:</td>
<td>xslatoms(1)</td>
</tr>
<tr>
<td>Protocol. tcp:</td>
<td>intemed atoms defined on server.</td>
</tr>
<tr>
<td>Protocol server. tfpd:</td>
<td>registerinmethost(3N)</td>
</tr>
<tr>
<td>udp:</td>
<td>Internet address manipulation</td>
</tr>
<tr>
<td>generator.</td>
<td>inet(3N)</td>
</tr>
<tr>
<td>driver.. ips, dkip:</td>
<td>Internet Bootstrap Protocol.</td>
</tr>
<tr>
<td></td>
<td>bootp(1M)</td>
</tr>
<tr>
<td></td>
<td>Internet Control Message Protocol.</td>
</tr>
<tr>
<td></td>
<td>named(1M)</td>
</tr>
<tr>
<td></td>
<td>Internet file transfer program.</td>
</tr>
<tr>
<td></td>
<td>ftp(1C)</td>
</tr>
<tr>
<td></td>
<td>Internet File Transfer Protocol.</td>
</tr>
<tr>
<td></td>
<td>ftdp(1M)</td>
</tr>
<tr>
<td></td>
<td>Internet protocol family.</td>
</tr>
<tr>
<td></td>
<td>inet(7f)</td>
</tr>
<tr>
<td></td>
<td>Internet Protocol.</td>
</tr>
<tr>
<td></td>
<td>ip(7P)</td>
</tr>
<tr>
<td></td>
<td>sendmail, newaliases.</td>
</tr>
<tr>
<td></td>
<td>sendmail(1M)</td>
</tr>
<tr>
<td></td>
<td>Internet &quot;super-server&quot;.</td>
</tr>
<tr>
<td></td>
<td>inetd(1M)</td>
</tr>
<tr>
<td></td>
<td>Internet TELNET protocol server.</td>
</tr>
<tr>
<td></td>
<td>telnetd(1M)</td>
</tr>
<tr>
<td></td>
<td>Internet Transmission Control</td>
</tr>
<tr>
<td></td>
<td>tcp(7P)</td>
</tr>
<tr>
<td></td>
<td>Internet Trivial File Transfer</td>
</tr>
<tr>
<td></td>
<td>tfpd(1M)</td>
</tr>
<tr>
<td></td>
<td>Internet User Datagram Protocol.</td>
</tr>
<tr>
<td></td>
<td>udp(7P)</td>
</tr>
<tr>
<td></td>
<td>interp: gamma-corrected color ramp</td>
</tr>
<tr>
<td></td>
<td>interp(6D)</td>
</tr>
<tr>
<td></td>
<td>Interphase disk controllers and</td>
</tr>
<tr>
<td></td>
<td>ips(7M)</td>
</tr>
</tbody>
</table>
Permuted Index

esh: a shell (command interpreter) with C-like syntax. ... csh(1)

pipe: create an interprocess channel. ... pipe(2)

facilities status. ipcs: report inter-process communication ... ipcs(1)

fio: standard interprocess communication package. ... stdio(3C)

release blocked signals and wait for interrupt (4.3BSD). /atomically ... sigpause(3B)

release blocked signals and wait for interrupt (POSIX). /atomically ... sigsuspend(2)

onintr: process interrupts in command scripts. ... csh(1)

trap: process interrupts in command scripts. ... sh(1)

sleep: suspend execution for an interval. ... sleep(1)

sleep: suspend execution for interval. ... sleep(3C)

setitimer: get/set value of interval timer. getitimer, ... getitimer(2)

application programs, and/into: introduction to commands, ... intro(1)

intro: introduction to file formats. ... intro(4)

intro: introduction to games and demos. ... intro(6)

and application programs. intro: introduction to maintenance commands ... intro(1M)

functions. math: introduction to mathematical library ... math(3M)

intro: introduction to miscellany. ... intro(5)

facilities. networking: introduction to networking ... netintro(7)

rsintro: introduction to RCS commands. ... rsintro(1)

intro: introduction to special files. ... intro(7)

libraries. intro: introduction to subroutines and ... intro(3)

error numbers. intro: introduction to system calls and ... intro(2)

nccheck: generate path names from i-numbers. ... nccheck(1M)

hinv: hardware inventory command. ... hinv(1M)

endinvent, scaninvent: get hardware inventory entry. /setinvent, ... getinvent(3)

asinh, acosh, atanh: inverse hyperbolic functions. ... asinh(3M)

trigonometric functions and their inverses. /facos, fatan, fatan2: ... trig(3M)

utility to enter arguments and invoke commands. launch: graphical ... launch(1)

cdio: 6-port serial I/O. ... cdio(7)

select: synchronous I/O multiplexing. ... select(2)

streamio: STREAMS ioctl commands. ... streamio(7)

ioctl: control device. ... ioctl(2)

ip: Internet Protocol. ... ip(7P)

mrdout: IP multicast routing daemon. ... mrdout(1M)

slip: Serial Line IP. ... slip(1M)

brushes. ipaint: Paint using bitmap images as ... ipaint(6D)

ipaste: display an image. ... ipaste(1G)

semaphore set or shared memory id. ipcrm: remove a message queue, ... ipcrm(1)

communication facilities status. ipcs: report inter-process ... ipcs(1)

ipi, xylip: Xlogics controllers and driver. ... ipi(7M)

controller and driver. ... ipi, xylip: Xlogics IP disk ... ipi(7M)

collectors and driver. ... ixmousemove, dpmove: Interphase disk ... ixmousemove(7)

gclear: clear IRIS graphics screen. ... gclear(1G)

xsgi: SGI Iris server for the X Window System. ... xsgi(1)

eternet: IRIS-4D Series ethernet controllers. ... ethernet(7)

event recording and playback from an IRIX shell. /journalal: 4Sight ... journalplay(1)

unname: identify the current IRIX system. ... unname(1)

unname: get identity of current IRIX system. ... unname(2)

/sxsdigit, islower, isupper, isalpha, isdigit, isxsdigit, islower, isupper, /iscntrl, ispunct, isprint, isgraph, ... ctype(3C)

isalpha, issalnum, isspace, iscntrl, ispunct, ... ctype(3C)

isalnum, isspace, iscntrl, ispunct, ... ctype(3C)

iscntrl, ispunct, isprint, isgraph, ... ctype(3C)

Version 5.0 April 1990
ttname, isatty: find name of a terminal
isncrl, ispunct, isprint, isgraph/
isdigit, isxdigit, islower, isupper,
isset: set the type of an image.
isgraph, isascii, tolower, toupper/
isspace, iscntrl, ispunct, isprint,
controller. ts:
isspace, isdigit, isxdigit,
isalnum, isspace, iscntrl, ispunct,
isalpha, isspace, iscntrl, islower,
islower, isupper, isalpha, issalnum,
system:
for use with file type rules.
of a list of image files..
isdigit, isxdigit, islower,
isalnum, isspace, iscntrl, isdigit,
news: print news
ftimer: control clock and
functions.
j0, j1, jn, y0, y1, yn: bessel
functions. j0,
j1, jn, y0, y1, yn: bessel
dynamics.
jello: simulates nonrigid body
j0, j1,
ustat: uucp status inquiry and
proto: prototype
kill: kill
queue. lprm: remove
jobs from the line printer spooling
jobs and processes.
join: relational database operator.
jot: a simple mouse-based text
viewer.
and playback toolchest.
and/ journalplay, journalrecord,
journalend: 4Sight event recording/
event recording and/ journalplay,
/erand48, srand48, mrand48, mrand48,
dragon: generates Mandelbrot and
Julia sets.
klog: kernel error logging interface.
setsym: set up a debug
lboot: configure bootable
kernl. lboot
sgikopt: retrieve
bindkey: function
makekey: generate encryption
makekey(1)
keyboard: keyboard specifications.
exmodmap: utility for modifying
keymaps in X.
parameters. mousewarpl, dialwarpl,
keywarp: set input warping
apropos: locate commands by
keyword lookup.
manuals; find manual entries by
xkill: kill a client by its X resource.
xkill
kill: kill jobs and processes.
kill: kill jobs and processes.
killall: kill named processes.

April 1990
Version 5.0
Permutated Index

awk: pattern scanning and processing
    language.                      awk(1)
    bc: arbitrary-precision arithmetic
    language.                      bc(1)
    cpp: the C
    command programming
    cftime: /the standard/restricted
    mkPS: register a
    chargeee, ckapct, dodisk,
    terminal/ wintem: utility to
    arguments and invoke commands.
    launch: graphical utility to enter
    efs: layout of the Extent file system.
    lboot: configure bootable kernel.
"/mrand48, jrand48, srand48, seed48,
    link editor.
    file. lclose, a member of an archive file.
    for reading. lopen, object file.
    floating-point numbers. frexp,
    routines.
    common object file.
    entry, given an index.
    object file symbol table entry.
    descriptor given a procedure/
    number entries of a common/ ldread,
    entries of a/ ldread, lldlimit,
    manipulate line number entries of a/
    number entries of a section of a/
    entries of a section of a/ ldsseek,
    of a section of a common/ ldsread,
    section header of a/ ldsread,
    section of a common object/ ldsseek,
    header of a common object file.
    object file for reading.
    relocation entries of a section of
    indexed/named section header of a/
    indexed/named section of a common/
    symbol table entry of a common/
    table entry of a common object/
    of a common object file.
kill: send a signal to a process or
    kill(2)
kill: send signal to a process
    kill(3B)
kill: terminate a process.
    kill(1)
killall: kill named processes.
    killall(1M)
group (4.3BSD).
    killpg: send signal to a process
    killpg(3B)
    klog: kernel error logging
    klog(7)
    mem, memm, memm: core memory.
    mem(7)
    integers and long integers.
    l3i64, ltof: convert between 3-byte
    l3i64(3C)
    and base-64 ASCII string.
    a64i, a64l: convert between long integer
    a64i(3C)
systems.
    labelit: provide labels for file
    labelit(1M)
    labels for file systems.
    labelit(1M)

Version 5.0
April 1990
-52-
exit: leave shell. ........................................ csh(1)
exit: leave shell. ........................................ sh(1)
an image to be stored using run length encoding. rle: force ........................................ rle(6D)
an image to be stored without run length encoding. verbatim: force ........................................ verbatim(6D)
truncate a file to a specified length. truncate, truncatem. ........................................ truncate(2)
getopt: get option letter from argument vector. ........................................ getopt(3C)
lex: generate programs for simple lexical tasks. ........................................ lex(1)
lex: generate programs for simple lexical tasks. ........................................ lex(1)
lsearch, lfind: linear search and update. ........................................ lsearch(3C)
look for name collisions between libraries. collide: ........................................ collide(1)
introduction to subroutines and libraries. intro: ........................................ intro(3)
ar: archive (library) file format. ........................................ ar(4)
math: introduction to mathematical library functions. ........................................ math(3M)
find ordering relation for an object library. lorder: ........................................ lorder(1)
ar: archive and library maintainer. ........................................ ar(1)
mkshlib: create a shared library. ........................................ mkshlib(1)
external data representation (xdr) library routines M_FORK(3P). ........................................ xdr:
rpc: Remote Procedure Call (RPC) library routines. ........................................ rpc(3R)
dgld: Distributed Graphics Library server. ........................................ dgld(1M)
lighting and shadows. light: demonstrates real-time lighting and shadows. ........................................ light(6D)
lights bouncing around a scene. ........................................ lights(6D)
implementation-specific constants. limits: file header for ........................................ limits(4)
ulimit: change or display size limits. ........................................ sh(1)
ulimit: get and set user limits. ........................................ ulimit(2)
establish an out-going terminal line connection. dial: ........................................ dial(3C)
tcdata, tcflush, tcfldset: posix line control primitives #include int/ ........................................ tcgetattrbreak,
set terminal type, modes, speed, and line discipline. getty: ........................................ getty(1M)
slip: Serial line discipline. uugetty: ........................................ uugetty(1M)
line. ........................................ line(1)
line: read one line. ........................................ line(1)
linenum: line number entries in a MIPS object ........................................ linenum(4)
object/ Adlinit, Idliitem: manipulate line number entries of a common ........................................ Idlread(3X)
a common/ Idlseek, Idlnseek: seek to line number entries of a section of ........................................ Idlseek(3X)
ul: line numbering filter. ........................................ ul(1)
cut: cut out selected fields of each line of a file. ........................................ cut(1)
lpr: off line print. ........................................ lpr(1)
lpc: line printer control program. ........................................ lpc(1M)
lpd: line printer daemon. ........................................ lpd(1M)
plp: parallel line printer interface. ........................................ plp(7)
send/cancel requests to an LP line printer. lp, cancel: ........................................ lp(1)
lprm: remove jobs from the line printer spooling queue. ........................................ lprm(1)
lp: read one line. ........................................ lp(1)
lsearch, lfind: linear search and update. ........................................ lsearch(3C)
col: filter reverse line-feeds. ........................................ col(1)
MIPS object file. linenum: line number entries in a ........................................ linenum(4)
lptest: generate lineprinter ripple pattern. ........................................ lptest(1)
comm: select or reject lines common to two sorted files. ........................................ comm(1)
device. fold: fold long lines for finite width output ........................................ fold(1)
head: give first few lines. ........................................ head(1)
uniq: report repeated lines in a file. ........................................ uniq(1)
Permuted Index

lines of several files or subsequent lines of one file. /merge same pastes(1)
directories. link, unlink: link and unlink files and link(1M)
ld, uld: MIPS link editor and ucode link editor. ld(1)
link: link editor. ld(1)
a.out: assembler and link editor output. a.out(4)
link: link to a file. link(2)
link: link or move files. cp(1)
readlink: read value of a symbolic link. readlink(2)
symlink: make symbolic link to a file. link(2)
and directories. link, unlink: link and unlink files link(1M)
drain: capture unimplemented link-layer protocols. drain(7P)
clone a file tree using symbolic links. llink: tlink(1)
lint: a C program checker. lint(1)
multi-colored pool of liquid. liquid: A faucet dripping into a liquid(li) liquid(6D)
set group access list application resource database. appres(1)
SETGROUPS(3B). get group access list (berkeley 4.3 version)/ setgroups:
symlink: make symbolic list (bsd 4.3 version) READV(2C). getgroups:
and directories. list CFGETOSPEED(3T). initgroups:
a display. xlsclients list client applications running on xlsclients(1)
ls: list contents of directory. ls(1)
glob: filename expand argument list. csh(1)
history: print history event list. csh(1)
shift: manipulate argument list. csh(1)
xlsfonts: server font list display for X. xlsfonts(1)
xlswins: server window list display for X. xlswins(1)
nm: name list dump of MIPS object files. nm(1)
getgroups: get group access list. getgroups(2)
initialize group access list GETGROUPS(3B). initgroups:
symlink: make symbolic list interned atoms defined on xlsatoms(1)
server. xslatoms: list interned atoms defined on xlsatoms(1)
nlist: get entries from name list. nlist(3X)
print the header information of a list of image files.. istat: istat(6D)
for each: loop over list of names. csh(1)
rhosts: list of trusted hosts and users. rhosts(4)
hosts.equiv: list of trusted hosts. hosts.equiv(4)
do, for: loop over list of words. sh(1)
listres: list resources in widgets. listres(1)
setgroups: set group access list. setgroups(2)
shift: manipulate argument list. sh(1)
stdarg: variable argument list. stdarg(5)
tape. List_tape: list the contents of a given backup list_tape(1)
varargs: variable argument list. varargs(5)
output of a variable argument list. /vprintf: print formatted vprintf(3S)
listen: listen: listen for connections on a socket. listen(2)
socket. listen: listen for connections on a listen(2)
listres: list resources in widgets. listres(1)
xargs: construct argument list(s) and execute command. xargs(1)
given backup tape. List_tape: list the contents of a list_tape(1)
Silicon Graphics

Permutated Index

| cp: | ln, mv: copy, link or move files. | cp(1) |
| cpload: | load average display for X. | xload(1) |
| /_procedure_string_table, _gp: | loader defined symbols in a program. | end(3C) |
| file. | loadmap: loads the colormap from a file. | loadmap(1G) |
| loadmap: | loads the colormap from a file. | loadmap(1G) |
| asctime(), strftime(), tzset, convert/aliases and path (csh only). which: | locate a program file including which(1) |
| apropos: | locate commands by keyword lookup. apropos(1) |
| for program. whereis: | locate source, binary, and or manual whereis(1) |
| usconfig: semaphore and configuration operations. | usconfig(3P) |
| usconfig: | lock control operations. | usconfiglock(3P) |
| usinit, _utrace: semaphore and lock initialization routine. | usinit(3P) |
| flock: apply or remove an advisory lock on an open file. | flock(3B) |
| _mpin, _munpin: | lock pages in memory. | _mpin(2) |
| memory. plock: | lock process, text, or data in memory. | plock(2) |
| out information about a specific process. | lock USDUMPSEMA(3P). dump | usdumplock: |
| usfrolock: free a lock. | usfrolock(3P) |
| usinitlock: initializes a lock. | usinitlock(3P) |
| allocates and initializes a lock. usnewlock: | usnewlock(3P) |
| lockf: record locking on files. | lockf(3C) |
| lockf: record locking on files. | lockf(3C) |
| ecc: dump memory ecc log. | ecc(1) |
| gamma: | log gamma function. | gamma(3M) |
| newgrp: log in to a new group. | newgrp(1) |
| newgrp: log in to a new group. | newgrp(1) |
| fexpm1, flog, flog10/_exp, expm1, about RCS files. rlog: print log messages and other information. | flog(1) |
| _flog: log of failed login attempts. | _flog(1) |
| setlogmask, vsyslog: control system log. syslog, openlog, closelog, syslog(3B) |
| syslogd: log systems messages. | syslogd(1M) |
| flog, flog10/_exp, expm1, log. | log10, log1p, pow, fexp. | exp(3M) |
| flog10/_exp, expm1, log, log10, | log10, log1p, pow, fexp, expm1, | exp(3M) |
| flog, flog10, flog1p: exponential, | log1p, pow, fexp, expm1, flog, | exp(3M) |
| exponent/ copy sign, drem, finite, | logb, scalb: copy sign, remainder, | copysign(3M) |
| exponent/ copy sign, drem, finite, | logb, scalb: copy sign, remainder, | copysign(3M) |
| rwho: who's logged in on local machines. | rwho(1C) |
| klog: kernel error logging interface. | klog(7) |
| lvinit: initialize logical volume devices. | lvinit(1M) |
| lv: logical volume Disk driver. | lv(7M) |
| mklv: construct or extend a logical volume. | mklv(1M) |
| check and restore consistency of logical volumes. lvck: | lvck(1M) |
| lvtab: information about logical volumes. lvtab(4) |
| loginlog: log of failed login attempts. | loginlog(4) |
| login: login configuration file. | login(4) |
| login: login configuration file. | login(4) |
| login: login new user. | login(4) |
| getlogin: get login name. | getlogin(3C) |
| logname: get login name. | logname(1) |
| cuserid: get character login name of the user. | cuserid(3S) |
| logname: return login name of user. | logname(3X) |
| login: login new user. | login(4) |
| login: login new user. | login(4) |

April 1990

-55-

Version 5.0
pandora: login on the graphics console.   pandora(1)
attributes. passwd: change login password and password passwd(1)
noidclogin: login process control. noiconlogin(5)
visaulogin: login process control. visuallogin(5)
select and control console login program. /noiconlogin: visuallogin(4)
rlogin: remote login. rlogin(1C)
rlogind: remote login server. rlogind(1M)
NeWS server. exporttonews: Pass a login shell's environment to the exporttonews(1)
setting up an environment at login: sign on. login(1)
last: indicate last login time. profile. profile(4)
logins of users and terminals. loginlog: log of failed login loginlog(4)
llogname: get login name. logname(1)
llogname: return login name of user. logname(3X)
olwm: OPEN LOOK window manager. olwm(1)
apropos: locate commands by keyword lookup. apropos(1)
finger: user information lookup program. finger(1)
break: exit while/foreach loop. sh(1)
continue: cycle in loop. sh(1)
end: terminate loop. sh(1)
for: loop over list of names. sh(1)
do, for: loop over list of words. sh(1)
break: exit while/for loop. sh(1)
continue: cycle in loop. sh(1)
done: terminate loop. sh(1)
an object library. lorder: find ordering relation for lorder(1)
an LP line printer. lp, cancel: send/cancel requests to lp(1)
cancel: send/cancel requests to an LP line printer. lp(1)
Centronics-interface printer with an LP line printer. mkcentpr: register a color mkcentpr(1M)
register a LaserWriter printer with LP: mkPS: mkPS(1M)
enable, disable: enable/disable enable(1)
by deleting/ preset: reset the LP queue system to a pristine state preset(1M)
accept, reject: allow or prevent LP requests. accept(1M)
/lpshut, lpmove: start/stop the LP scheduler and move requests. lpsched(1M)
ladmin: configure the LP spooling system. lpsched(1M)
rmprinter: remove a printer from the LP spooling system. rmprinter(1M)
lpsched, lpmove: start/stop the LP scheduler and move requests. lpsched(1M)
and move requests. lpsched, lpmove: start/stop the LP scheduler and move requests. lpsched(1M)
lpstat: print LP status information. lpstat(1)
lpadmin: configure the LP spooling system. lpadmin(1M)
lpc: line printer control program. lpc(1M)
lpd: line printer daemon. lpd(1M)
lp: print queue examination. lp(1)
lpr: off line print. lpr(1)
lpadmin: configure the LP spooling system. lprn(1)
the LP scheduler and move requests. lpsched, lpmove: start/stop the LP scheduler and move requests. lpsched(1M)

Version 5.0 -56- April 1990
Silicon Graphics

lpstat: print LP status information.
lpstat(1)

pattern.
lptest: generate lineprinter ripple
lpstat(1)

rand48, seed48, srand48, irand48, erand48,
rand48, nrand48, mrand48, jrand48,
rand48(3C)

ls: list contents of directory.
ls(1)

update.
isearch, lfind: linear search and
isearch(3C)

(System V and 4.3BSD).
iseek: move read/write file pointer
iseek(2)

stat,
lsat: get file status.
stat(2)

to integers and long integers. il32,
il03: convert between 3-byte
il32(3C)

lv: logical volume Disk driver.
1v(7M)

of logical volumes.
1vck: check and restore consistency
1vck(1M)

devices.
ivinit: initialize logical volume
ivinit(1M)

freetent: get
lvtab file entry REMOVE(3C).
getfent,
lvtab: information about logical
lvtab(4)

volumes.
m4: macro processor.
m4(1)

u3b, u3b2, u3b5, u3b15, vax, m68k,
m68000, mips, 4d, 4d60: get/ pdp11,
m68k, m68000, mips, 4d, 4d60: get/
machid(1)

the/ /get the byte sex of the host
machine swap_*() - swap the sex of
sex(3X)

values: machine-dependent values.
values(5)

sgetl: access long integer data in a
machine-independent fashion. sputl,
sputl(3X)
runtime: show host status of local
machines. runtime(1C)
rwho: who's logged in on local
machines. rwho(1C)
m4: macro processor.
m4(1)
alias: shell
macros. csh(1)
memory.
madvise: give advise about handling
madvise(2)
magnetic tape controller.
xmt(7M)
mag: pixel replication and
mag(6D)
mtio: magnetic tape interface.
mtio(7)
mt: magnetic tape manipulating program.
mt(1)
magnification in a window.
magnification in a window.
magnify and report on the screen
snoop(6D)
magnify or shrink an image.
zoom(6D)
xmag: magnify parts of the screen.
xmag(1)
xmt: remote
magtape protocol module.
xmt(1M)
rebuild the database file for
mail aliases file. newaliases:
newaliases(1M)
mail: send mail to users or read
mail.
Mail: send and receive
mail.
mailbox: mail notification.
mailbox(1)
sendmail, newaliases, mailq: send
mail over the internet.
sendmail(1M)
Mail: send and receive mail.
mailbox(1)
mail.
mail: send mail to users or read
mail.
mail: send
to users or read mail.
mail(1)
a binary file for transmission via
mail. /udecode: encode/decode
uencode(1C)
rm: send, receive
mail via UUCP.
rm(1M)
xbiff:
mailbox flag for X.
xbiff(1)
mailbox: mail notification.
mailbox(1)
sendmail, newaliases,
mailq: send mail over the internet.
sendmail(1M)
admin, amaninfo: arbitrary arena
main memory allocator. /acalloc,
amalloc(3P)
malloc, free, realloc, calloc:
main memory allocator.
malloc(3C)
calloc, malloc, maninfo: fast
main memory allocator.
malloc(3X)
groups of programs. make:
maintainer, update, and regenerate
make(1)
ar: archive and library
ar(1)
### Permutated Index

<table>
<thead>
<tr>
<th>Programs</th>
<th>Maintenance commands and application</th>
<th>intro(1M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mkbootape:</td>
<td>make a boot tape.</td>
<td>mkbootape(1M)</td>
</tr>
<tr>
<td>File, Delta</td>
<td>make a delta (change) to an SCCS</td>
<td>delta(1)</td>
</tr>
<tr>
<td>Mkdir</td>
<td>make a directory.</td>
<td>mkdir(2)</td>
</tr>
<tr>
<td>Ordinary File, Mknod</td>
<td>make a directory, or a special or</td>
<td>mknod(2)</td>
</tr>
<tr>
<td>Mkfifo</td>
<td>make a FIFO special file.</td>
<td>mkfifo(2)</td>
</tr>
<tr>
<td>Mktemp, Mkstemp</td>
<td>make a unique file name.</td>
<td>mktemp(3C)</td>
</tr>
<tr>
<td>Greyscale</td>
<td>make different patterns.</td>
<td>greyscale(6D)</td>
</tr>
<tr>
<td>Mkdir</td>
<td>make directories.</td>
<td>mkdir(1)</td>
</tr>
<tr>
<td>Regenerate Groups of Programs</td>
<td>make: maintain, update, and</td>
<td>make(1)</td>
</tr>
<tr>
<td>For Use with Apropos, Makewhatis</td>
<td>make manual page &quot;whatis&quot; database</td>
<td>makewhatis(1M)</td>
</tr>
<tr>
<td>Banner</td>
<td>make posters.</td>
<td>banner(1)</td>
</tr>
<tr>
<td>Readonly</td>
<td>make shell variables read-only.</td>
<td>sh(1)</td>
</tr>
<tr>
<td>Symlink</td>
<td>make symbolic link to a file.</td>
<td>symlink(2)</td>
</tr>
<tr>
<td>Makemap</td>
<td>make the default color map.</td>
<td>makemap(1G)</td>
</tr>
<tr>
<td>Script</td>
<td>make typescript of terminal session.</td>
<td>script(1)</td>
</tr>
<tr>
<td>C Preprocessor Interface to the Makefiles</td>
<td>make utility.</td>
<td>make(1)</td>
</tr>
<tr>
<td>Files</td>
<td>make depend: create dependencies in</td>
<td>make depend(1)</td>
</tr>
<tr>
<td>Makedepend: Create Dependencies in</td>
<td>makefiles.</td>
<td>makefiles(1M)</td>
</tr>
<tr>
<td>Makefiles</td>
<td>makekey: generate encryption key.</td>
<td>makekey(1)</td>
</tr>
<tr>
<td>&quot;whatis&quot; Database for Use with/</td>
<td>makewhatis: make manual page</td>
<td>makewhatis(1M)</td>
</tr>
<tr>
<td>Free, Realloc, Calloc, Malloc,</td>
<td>mallocinfo: fast main memory / malloc,</td>
<td>malloc(3X)</td>
</tr>
<tr>
<td>Memory Allocator</td>
<td>malloc, free, realloc, calloc: main</td>
<td>malloc(3C)</td>
</tr>
<tr>
<td>Malloc, Mallocinfo, Fast Main Memory/</td>
<td>malloc, free, realloc, calloc</td>
<td>malloc(3X)</td>
</tr>
<tr>
<td>Malloc, Free, Realloc, Calloc</td>
<td>malloc, calllopt, mallocinfo: fast main memory/</td>
<td>malloc(3C)</td>
</tr>
<tr>
<td>Manwh: Display a Man Page and then Prompt for Input</td>
<td>manpage(6D)</td>
<td></td>
</tr>
<tr>
<td>Reference Manuals; Find Manual/</td>
<td>man: print entries from the on-line</td>
<td>man(1)</td>
</tr>
<tr>
<td>Tsearch, Tfind, Tdelete, Twalk</td>
<td>manage binary search trees.</td>
<td>tsearch(3C)</td>
</tr>
<tr>
<td>Hsearch, Hcreate, Hdestroy</td>
<td>manage hash search tables.</td>
<td>hsearch(3C)</td>
</tr>
<tr>
<td>Passmgmt: Password File</td>
<td>management.</td>
<td>passmgmt(1M)</td>
</tr>
<tr>
<td>Sigignore, Sigpause: Signal Management (System V), /sysrellse, /sigset</td>
<td>sigset(2)</td>
<td></td>
</tr>
<tr>
<td>Twm: Tab Window</td>
<td>Manager for the X Window System.</td>
<td>twm(1)</td>
</tr>
<tr>
<td>Sm: A Session</td>
<td>manager for x.</td>
<td>sm(1)</td>
</tr>
<tr>
<td>Uwm: A Window</td>
<td>manager for x.</td>
<td>uwm(1)</td>
</tr>
<tr>
<td>Olwm: OPEN LOOK Window</td>
<td>manager.</td>
<td>olwm(1)</td>
</tr>
<tr>
<td>Xdm: X Display</td>
<td>Manager.</td>
<td>xdm(1)</td>
</tr>
<tr>
<td>Dragon: Generates Mandelbrot and Julia Sets</td>
<td>dragon(6D)</td>
<td></td>
</tr>
<tr>
<td>Shift</td>
<td>manipulate argument list.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>Shift</td>
<td>manipulate argument list.</td>
<td>sh(1)</td>
</tr>
<tr>
<td>Records, Fwtmp, Wtmpfix</td>
<td>manipulate connect accounting</td>
<td>fwtmp(1M)</td>
</tr>
<tr>
<td>Common, Idread, Idlimit, Idlitem</td>
<td>manage line number entries of a</td>
<td>Idread(3X)</td>
</tr>
<tr>
<td>Numbers, Freqx, Freqx, Modif</td>
<td>manage parts of floating-point</td>
<td>freqx(3C)</td>
</tr>
<tr>
<td>Route: Manually</td>
<td>manipulate the routing tables.</td>
<td>route(1M)</td>
</tr>
<tr>
<td>Mt: Magnetic Tape</td>
<td>manipulating program.</td>
<td>mt(1)</td>
</tr>
<tr>
<td>Routines, /Sgi_Dumpset: Signal Set Manipulation and Examination</td>
<td>sigset(3)</td>
<td></td>
</tr>
<tr>
<td>Inet_netof: Internet Address</td>
<td>manipulation routines. /inet_Inaof, /inet(3N)</td>
<td></td>
</tr>
<tr>
<td>Scalb: Copysign, Remainder, Exponent</td>
<td>manipulations. /drem, finite, logb, ieee(3M)</td>
<td></td>
</tr>
<tr>
<td>Scalb: Copysign, Remainder, Exponent</td>
<td>manipulations. /drem, finite, logb, ieee(3M)</td>
<td></td>
</tr>
<tr>
<td>On-line Reference Manuals; Find Manual Entries by Keyword. /From</td>
<td>man(1)</td>
<td></td>
</tr>
</tbody>
</table>

**Version 5.0**

---

April 1990
<table>
<thead>
<tr>
<th>Silicon Graphics</th>
<th>Permutated Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>locate source, binary, and or X Window System... xman:</td>
<td>manual for program. whereas: ... whereas(1)</td>
</tr>
<tr>
<td>use with apropos. makewhatis: make tables. route:</td>
<td>Manual page display program for the manual page &quot;whatis&quot; database for manually manipulate the routing</td>
</tr>
<tr>
<td>/entries from the on-line reference prompt for input. bstream:</td>
<td>manuals; find manual entries by/ manwsh: display a man page and then many buffered filter.</td>
</tr>
<tr>
<td>makemap: make the default color mmap:</td>
<td>map. ... map pages of memory.</td>
</tr>
<tr>
<td>display the contents of the color into an RGB image.</td>
<td>map. showmap. ... showmap(6D)</td>
</tr>
<tr>
<td>TCP, UDP port to RPC program number to facilitate better cache rotimg:</td>
<td>mapping.. /in an executable ... maps an image onto a surface.</td>
</tr>
<tr>
<td>cachectl:</td>
<td>mark pages cacheable or uncacheable. cachectl(2)</td>
</tr>
<tr>
<td>sigsetmask: set current signal change or display file creation</td>
<td>mask (4.3BSD). ... sigsetmask(3B)</td>
</tr>
<tr>
<td>change or display file creation umask: set file-creation mode</td>
<td>mask. ... mask. umask. ... sh(1)</td>
</tr>
<tr>
<td>umask: set and get file creation create an error message file by</td>
<td>mask. ... mask. ... mkstr(1)</td>
</tr>
<tr>
<td>master:</td>
<td>master configuration database. ... master(4)</td>
</tr>
<tr>
<td>database.</td>
<td>master: master configuration ... master(4)</td>
</tr>
<tr>
<td>regular expression compile and math:</td>
<td>match routines. regexp: ... math(5)</td>
</tr>
<tr>
<td>math: introduction to library functions.</td>
<td>math: introduction to mathematical</td>
</tr>
<tr>
<td>math: introduction to getrlimit, setrlimit: control demo [ ] X11 ].</td>
<td>math: math functions and constants.</td>
</tr>
<tr>
<td>maze: an automated maze program... [ demo ] X11 ].</td>
<td>mathematical library functions.</td>
</tr>
<tr>
<td>winicons: stowed window image</td>
<td>maximum system resource consumption. getrlimit(2)</td>
</tr>
<tr>
<td>/read the archive header of a group: group</td>
<td>maze: an automated maze program...</td>
</tr>
<tr>
<td>multgrps: spawn a shell with</td>
<td>membcmp, memcmy, memset, memccpy: memory operations.</td>
</tr>
<tr>
<td>groups: show group</td>
<td>memchr, membcmp, memcmy, memset, memccpy: memory operations.</td>
</tr>
<tr>
<td>memchr, memcmy, memcmy, memset, memccpy: memory operations.</td>
<td>memccpy: memory operations.</td>
</tr>
<tr>
<td>memory operations. memchr, operations. membcmp, memcmy, amallinfo: arbitrary arena main</td>
<td>memchr, membcmp, memcmy, memset, memccpy: memory operations.</td>
</tr>
<tr>
<td>malloc, free, realloc, calloc: main</td>
<td>membcmp, memcmy, memset, memccpy:</td>
</tr>
<tr>
<td>calloc, malloc, mallocinfo: fast main</td>
<td>memory allocator. /malloc(), ... malloc(3P)</td>
</tr>
<tr>
<td>usmalloc, usmallocinfo: user shared</td>
<td>memory allocator. /malloc(), ... malloc(3P)</td>
</tr>
<tr>
<td>shmctl: shared</td>
<td>memory control operations. ... shmctl(2)</td>
</tr>
<tr>
<td>ecc: dump</td>
<td>memory ecc log. ... ecc(1)</td>
</tr>
<tr>
<td>spawn new process in a virtual queue, semaphore set or shared madvise: give advise about handling</td>
<td>memory efficient way. vfork: ... memory id. ipcrm: remove a message</td>
</tr>
<tr>
<td>mem, kmem, mmem: core</td>
<td>memory. ...</td>
</tr>
</tbody>
</table>

April 1990 —59— Version 5.0
Permutated Index

mmap: map pages of memory. mmap(2)
m pin, munpin: lock pages in memory. m pin(2)
munmap: unmap pages of memory. munmap(2)
memcmp, memcmp, memset, memcpy:
mem operations. memchr, memory(3C)
shm allocate, shmdt: shared memory operations. shmemop(2)
lock process, text, or data in memory. plock(2)
shmget: get shared memory segment identifier. shmget(2)
msync: synchronize memory with physical storage. msync(2)
memchr, memcmp, memcpy, for extending the Workspace menu functions. / specification transferdevice(4)
administration. sys adm: menu interface to do system sysadm(1)
entries in the workspace transfer menu. / interface for selecting transfermanager(1G)
sort: sort and/or merge files. sort(1)
merge: three-way file merge. merge(1)
acctmrg: merge or add total accounting files. acctmrg(1M)
rcsmmerge: merge RCS revisions. rcsmmerge(1)
subsequent lines of one/ paste: merge same lines of several files or paste(1)
merge: three-way file merge. merge(1)
msg: permit or deny messages. msg(1)
msgctl: message control operations. msgctl(2)
xmessage: X window system message display program. xmessage(1)
mkstr: create an error message file by massaging C source. mkstr(1)
recv, recvfrom, recvmsg: receive a message from a socket. recv(2)
send, sendto, sendmsg: send a message to a socket. send(2)
xmh: X interface to the MH message handling system. xmh(1)
response. confirm: display a message in a window and request a confirm(1G)
inform: display a message in a window. inform(1G)
modt: message of the day. modt(4)
getmsg: get next message off a stream. getmsg(2)
putmsg: send a message on a stream. putmsg(2)
msgsnd, msgrcv: message operations. msgop(2)
icmp: Internet Control Message Protocol. icmp(7P)
msg get: message queue. msgget(2)
shared memory id. ipcrm: remove a message queue, semaphore set or ipcrm(1)
help: ask for help about SCCS error messages and commands. help(1)
RCS files. rlog: print log messages and other information about rlog(1)
msg: permit or deny messages. msg(1)
sys_errno, sys_nerr: system error messages. perror, sterror, erro, perror(3C)
psignal, sys_ siglist: system signal messages. psignal(3C)
syslogd: log systems messages. syslogd(1M)
values. sysmeter: meter display of system performance sysmeter(1)
(xdr) library routines M_FORK(3P). / data representation xdr:

/m_set_procs, m_get_numprocs, m_set_myid, m_next, m_lock, /m_get_numprocs, m_set_procs, m_get_numprocs, m_get_myid, m_next, m_lock,
xmh: X interface to the minor device on a STREAMS driver. xmh(1)
clone: open any minor device on a STREAMS driver. clone(7)
/u3b5, u3b15, vac, m68k, m68000, mips, 4d, 4d60: get processor type/ machid(1)
as: MIPS assembler. as(1)
emulate_branch: MIPS branch emulation. emulate_branch(3X)
cc: MIPS C compiler. cc(1)
call. sysmips: MIPS Computer Systems Inc. system sysmips(2)
<table>
<thead>
<tr>
<th>Silicon Graphics</th>
<th>Permutated Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>an identifying number. tag: tag a</td>
<td>MIPS executable or shell script with .tag(1)</td>
</tr>
<tr>
<td>disassembler: disassemble a</td>
<td>MIPS instruction and print the/ .disassembler(3X)</td>
</tr>
<tr>
<td>editor. ld, uld:</td>
<td>MIPS link editor and ucode link .ld(1)</td>
</tr>
<tr>
<td>linenum: line number entries in</td>
<td>MIPS object file. .linenum(4)</td>
</tr>
<tr>
<td>scnhdr: section header for a</td>
<td>MIPS object file. .scnhdr(4)</td>
</tr>
<tr>
<td>filehdr: file header for</td>
<td>MIPS object files. .filehdr(4)</td>
</tr>
<tr>
<td>nm: name list dump of</td>
<td>MIPS object files. .nm(1)</td>
</tr>
<tr>
<td>a binary read/write interface to the</td>
<td>MIPS symbol table. /hat provide .stco(3X)</td>
</tr>
<tr>
<td>/acctwtmp: overview of accounting and</td>
<td>miscellaneous accounting commands. .acct(1M)</td>
</tr>
<tr>
<td>intro: introduction to</td>
<td>miscellany. .intro(5)</td>
</tr>
</tbody>
</table>

<p>| Centronics interface printer with/ | MIT X Consortium. .xconsortium(1) |
| dependencies. | mkboottape: make a boot tape. .mkboottape(1M) |
| mkcentpr: register a color | mkcentpr(1M) |
| mkdepend: compute header file | mkdepend(1) |
| mkdir: make a directory. | mkdir(2) |
| mkdir: make directories. | mkdir(1) |
| routines. | mkf2c: generate FORTRAN-C interface .mkf2c(1) |
| mkfifo: make a FIFO special file. | mkfifo(2) |
| mkfile: create a file. | mkfile(1M) |
| mkfontdir: create fonts.dir file | mkfontdir(1) |
| mkfs: construct a file system. | mkfs(1M) |
| from directory of font files. | mkf2c(1) |
| m_get_numprocs, m_get_myid, m_next/ | m_kill_procs, m_set_procs, .m_kill(1M) |
| volume. | m_fork, .m_fork(1M) |
| mklv: construct or extend a logical | mklv(1M) |
| mknetpr: provide access to a remote | mknetpr(1M) |
| pipe (FIFO). | mknod: build special file or named .mknod(1M) |
| special or ordinary file. | mknod: make a directory, or a .mknod(2) |
| with LP. | mkPS: register a LaserWriter printer .mkps(1M) |
| mkshlib: create a shared library. | mkshlib(1) |
| mktemp, | mktemp: make a unique file name. .mktemp(3C) |
| by massaging C source. | mkstr: create an error message file .mkstr(1) |
| name. | mktemp(3C) |
| /m_get_numprocs, m_get_myid, m_next, | m_lock, m_unlock, m_park_procs, .m_lock(2) |
| m_get_numprocs, m_get_myid, | m_fork, .m_fork(1M) |
| m_lock, m_unlock, m_park_procs,/ | mmap: map pages of memory. .mmap(2) |
| mem, kmem, | mmem: core memory. .mmem(7) |
| /m_get_numprocs, m_get_myid, | m_next, m_lock, m_unlock,/ .m_next(1M) |
| umask: set file-creation | umask(1) |
| switch the system to multi-user | mode. multi:. .multi(1M) |
| chmod: change the permissions | mode of a file or directory. .chmod(1) |
| chmod, fchmod: change | mode of file. .chmod(2) |
| ds: generic (user | mode) SCSI driver. .ds(7M) |
| switch the system to single-user | mode. single:. .single(1M) |
| newton: a physical | modeling demo. .newton(6D) |
| network. dglnewton: a physical | modeling demo running across a .dglnewton(6D) |
| getty: set terminal type, | modes, speed, and line discipline. .getty(1M) |
| uugetty: set terminal type, | modes, speed, and line discipline. .uugetty(1M) |
| floating-point/ frexp, ldexp, | modf: manipulate parts of .frexp(3C) |
| touch: update access and | modification times of a file. .touch(1) |
| utime: set file access and | modification times. .utime(2) |
| information. dvhtool: | modify and obtain disk volume header .dvhtool(1M) |
| process. npri: | modify the scheduling priority of a .npri(1) |
| xmodmap: utility for | modifying keymaps in X. .xmodmap(1) |</p>
<table>
<thead>
<tr>
<th>Permutated Index</th>
<th>Silicon Graphics</th>
</tr>
</thead>
<tbody>
<tr>
<td>rmt: remote magtape protocol</td>
<td>module.</td>
</tr>
<tr>
<td>/ckpact, dodisk, lastlogin,</td>
<td>monacct, nulldm, prctmp, prdaily/</td>
</tr>
<tr>
<td>profile. monitor, monstartup,</td>
<td>moncontrol: prepare execution</td>
</tr>
<tr>
<td>imon: inode</td>
<td>monitor device.</td>
</tr>
<tr>
<td>gr_osview: graphical system</td>
<td>monitor.</td>
</tr>
<tr>
<td>prepare execution profile.</td>
<td>monitor, monstartup, moncontrol:</td>
</tr>
<tr>
<td>data. osview:</td>
<td>monitor operating system activity</td>
</tr>
<tr>
<td>setmon: set the default</td>
<td>monitor video output format.</td>
</tr>
<tr>
<td>snoop: network</td>
<td>monitoring protocol.</td>
</tr>
<tr>
<td>execution profile. monitor,</td>
<td>monstartup, moncontrol: prepare</td>
</tr>
<tr>
<td>flip: spin one or</td>
<td>more objects.</td>
</tr>
<tr>
<td>crt viewing.</td>
<td>more, page: file perusal filter for</td>
</tr>
<tr>
<td>mount:</td>
<td>mod: message of the day.</td>
</tr>
<tr>
<td>mount, umount:</td>
<td>mount a file system.</td>
</tr>
<tr>
<td>mountable filesystems.</td>
<td>mount and dismount filesystems.</td>
</tr>
<tr>
<td>setmnt: establish</td>
<td>mount: mount a file system.</td>
</tr>
<tr>
<td>filesystems.</td>
<td>mount table.</td>
</tr>
<tr>
<td>systems. mountall, umountall:</td>
<td>mount, umount: mount and dismount</td>
</tr>
<tr>
<td>multiple file systems.</td>
<td>mount, umount multiply file</td>
</tr>
<tr>
<td>mtab:</td>
<td>mounted file system table.</td>
</tr>
<tr>
<td>interactive local and network disk</td>
<td>mountd tool. disks:</td>
</tr>
<tr>
<td>and report on the screen under the</td>
<td>mouse: optical mouse specifications.</td>
</tr>
<tr>
<td>mouse: optical</td>
<td>mouse pointer. snoop: magnify</td>
</tr>
<tr>
<td>jut: a simple</td>
<td>mouse specifications.</td>
</tr>
<tr>
<td>mouse-based text editor.</td>
<td>mouse-based text viewer.</td>
</tr>
<tr>
<td>jutview: a simple</td>
<td>input warping parameters.</td>
</tr>
<tr>
<td>mvdif:</td>
<td>move a directory.</td>
</tr>
<tr>
<td>cp, ln, mv: copy, link or</td>
<td>move files.</td>
</tr>
<tr>
<td>V and 4.3BSD. lseek:</td>
<td>move requests. /lpshut, lpmove:</td>
</tr>
<tr>
<td>start/stop the L.P scheduler and</td>
<td>madmin: control and report</td>
</tr>
<tr>
<td>processor status.</td>
<td></td>
</tr>
<tr>
<td>parallel/ /m_next, m_lock, m_unlock,</td>
<td>m_park_procs, m_rele_procs, m_sync:</td>
</tr>
<tr>
<td>m_get_myid, m_next, m_kill_procs,</td>
<td>m_park_procs, m_rele_procs, m_sync:</td>
</tr>
<tr>
<td>m_set_procs, m_get_numprocs,</td>
<td>msgctl: message control operations.</td>
</tr>
<tr>
<td>m_park_procs, m_rele_procs, m_sync: parallel/</td>
<td>msgget: get message queue.</td>
</tr>
<tr>
<td>m_sync: parallel programming/</td>
<td>msgsnd, msgrcv: message operations.</td>
</tr>
<tr>
<td>physical storage.</td>
<td>msync: synchronize memory with</td>
</tr>
<tr>
<td>program.</td>
<td>mt: magnetic tape manipulating</td>
</tr>
<tr>
<td>mtio: magnetic tape interface.</td>
<td>mtab: mounted file system table.</td>
</tr>
<tr>
<td>membership in multiple groups.</td>
<td>multigrp: spawn a shell with</td>
</tr>
<tr>
<td>multi-user mode.</td>
<td>multi: switch the system to</td>
</tr>
<tr>
<td>mrouterd: IP multicast routing daemon.</td>
<td>mrouter(1M)</td>
</tr>
<tr>
<td>liquid: A faucet dripping into a multi-colored pool of liquid.</td>
<td>liquid(6D)</td>
</tr>
<tr>
<td>mountall, umountall: mount, umount multiple file systems.</td>
<td>mountall(1M)</td>
</tr>
<tr>
<td>spawn a shell with membership in multiple groups.</td>
<td>multgrps(1)</td>
</tr>
<tr>
<td>poll: input/output multiplexing.</td>
<td>poll(2)</td>
</tr>
<tr>
<td>select: synchronous I/O multiplexing.</td>
<td>select(2)</td>
</tr>
<tr>
<td>sympp: multiprocessor control.</td>
<td>sympp(2)</td>
</tr>
<tr>
<td>rc2: run commands performed for multi-user environment.</td>
<td>rc2(1M)</td>
</tr>
<tr>
<td>multi: switch the system to multi-way command branch.</td>
<td>multi(1M)</td>
</tr>
<tr>
<td>switch: multi-way command branch.</td>
<td>sh(1)</td>
</tr>
<tr>
<td>case: muncher: draw interesting patterns muncher(1)</td>
<td></td>
</tr>
<tr>
<td>/m_get_myid, m_next, m_lock, m_unlock, m_park_procs/ m_fork, munmap: unmap pages of memory. munmap(2)</td>
<td></td>
</tr>
<tr>
<td>mpin, munmap: lock pages in memory. mpin(2)</td>
<td></td>
</tr>
<tr>
<td>cp, ln, mv: copy, link or move files. cp(1)</td>
<td></td>
</tr>
<tr>
<td>mdir: move a directory. mdir(1M)</td>
<td></td>
</tr>
<tr>
<td>collide: look for name collisions between libraries. collide(1)</td>
<td></td>
</tr>
<tr>
<td>hosts: host name data base. hosts(4)</td>
<td></td>
</tr>
<tr>
<td>networks: network name data base. networks(4)</td>
<td></td>
</tr>
<tr>
<td>protocols: protocol name data base. protocols(4)</td>
<td></td>
</tr>
<tr>
<td>services: service name data base. services(4)</td>
<td></td>
</tr>
<tr>
<td>devnm: device name. devnm(1M)</td>
<td></td>
</tr>
<tr>
<td>tmpnam, tempnam: create a name for a temporary file. tmpnam(3S)</td>
<td></td>
</tr>
<tr>
<td>entry. ldgetname: retrieve symbol name for object file symbol table. ldgetname(3X)</td>
<td></td>
</tr>
<tr>
<td>ctermid: generate file name for terminal. ctermid(3S)</td>
<td></td>
</tr>
<tr>
<td>getpw: get name from UID. getpw(3C)</td>
<td></td>
</tr>
<tr>
<td>getenv: return value for environment name. getenv(3C)</td>
<td></td>
</tr>
<tr>
<td>getlogin: get login name. getlogin(3C)</td>
<td></td>
</tr>
<tr>
<td>getsockname: get socket name. getsockname(2)</td>
<td></td>
</tr>
<tr>
<td>nm: name list dump of MIPS object files. nm(1)</td>
<td></td>
</tr>
<tr>
<td>nlist: get entries from name list. nlist(3X)</td>
<td></td>
</tr>
<tr>
<td>logname: get login name. logname(1)</td>
<td></td>
</tr>
<tr>
<td>mktemp, mkstemp: make a unique file name. mktemp(3C)</td>
<td></td>
</tr>
<tr>
<td>rename: change the name of a file. rename(2)</td>
<td></td>
</tr>
<tr>
<td>ttyname, istatty: find name of a terminal. ttyname(3C)</td>
<td></td>
</tr>
<tr>
<td>getpeername: get name of connected peer. getpeername(2)</td>
<td></td>
</tr>
<tr>
<td>/setdomainname: get/set name of current domain. setdomainname(2)</td>
<td></td>
</tr>
<tr>
<td>gethostname, sethostname: get/set name of current host. gethostname(2)</td>
<td></td>
</tr>
<tr>
<td>hostname: set or print name of current host system. hostname(1)</td>
<td></td>
</tr>
<tr>
<td>tty: get the name of the terminal. tty(1)</td>
<td></td>
</tr>
<tr>
<td>cuserid: get character login name of the user. cuserid(3S)</td>
<td></td>
</tr>
<tr>
<td>logname: return login name of user. logname(3X)</td>
<td></td>
</tr>
<tr>
<td>pwd: working directory name. pwd(1)</td>
<td></td>
</tr>
<tr>
<td>hostname: host name resolution description. hostname(5)</td>
<td></td>
</tr>
<tr>
<td>named: Internet domain name server. named(1M)</td>
<td></td>
</tr>
<tr>
<td>nslookup: query name servers interactively. nslookup(1C)</td>
<td></td>
</tr>
<tr>
<td>bind: bind to a name to a socket. bind(2)</td>
<td></td>
</tr>
<tr>
<td>named: Internet domain name server. named(1M)</td>
<td></td>
</tr>
<tr>
<td>mknod: build special file or named pipe (FIFO). mknod(1M)</td>
<td></td>
</tr>
<tr>
<td>killall: kill named processes. killall(1M)</td>
<td></td>
</tr>
<tr>
<td>dimname: deliver portions of path names. basename, basename(1)</td>
<td></td>
</tr>
<tr>
<td>Permutated Index</td>
<td>Silicon Graphics</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>foreach: loop over list of names.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>term: conventional names for terminals.</td>
<td>term(5)</td>
</tr>
<tr>
<td>ncheck: generate path names from i-numbers.</td>
<td>ncheck(1M)</td>
</tr>
<tr>
<td>id: print user and group IDs and names.</td>
<td>id(1)</td>
</tr>
<tr>
<td>usr, rusr, swap, rswap: Partition names.. root, root,.</td>
<td>root(7M)</td>
</tr>
<tr>
<td>interface for selecting X11 font names.</td>
<td>xfontsel: point &amp; click xfontsel(1)</td>
</tr>
<tr>
<td>controller. gpib: driver for National Instruments VME IEEE-488</td>
<td>gpib(7M)</td>
</tr>
<tr>
<td>processing language. nawk: pattern scanning and</td>
<td>nawk(1)</td>
</tr>
<tr>
<td>i-numbers. ncheck: generate path names from</td>
<td>ncheck(1M)</td>
</tr>
<tr>
<td>/ceiling, remainder, absolute value,</td>
<td>floor(3M)</td>
</tr>
<tr>
<td>symbol/ /interface to basic functions</td>
<td>stfc(3X)</td>
</tr>
<tr>
<td>needed to access and add to the</td>
<td>netstat: show network status. netstat(1)</td>
</tr>
<tr>
<td>convert values between host and network byte order. ntohl, ntohs: byteorder(3N)</td>
<td></td>
</tr>
<tr>
<td>modeling demo running across a network. dglnet: a physical</td>
<td>dglnet(6D)</td>
</tr>
<tr>
<td>raytracer running across a network. dglray: a visualized</td>
<td>dglray(6D)</td>
</tr>
<tr>
<td>disks: interactive local and network disk mounts tool. disks(1G)</td>
<td></td>
</tr>
<tr>
<td>setnetent, endnetent: get network entry. getnetbyname.</td>
<td>getnetent(3N)</td>
</tr>
<tr>
<td>sethostent, endhostent, error: get network host entry. gethostbyname.</td>
<td>gethostbyname(3N)</td>
</tr>
<tr>
<td>send ICMP ECHO_REQUEST packets to network hosts. ping:</td>
<td>ping(1M)</td>
</tr>
<tr>
<td>script. network: network initialization and shutdown network(1M)</td>
<td></td>
</tr>
<tr>
<td>ifconfig: configure network interface parameters.</td>
<td>ifconfig(1M)</td>
</tr>
<tr>
<td>snoop: network monitoring protocol.</td>
<td>snoop(7P)</td>
</tr>
<tr>
<td>networks: network name data base.</td>
<td>networks(4)</td>
</tr>
<tr>
<td>shutdown script. network: network initialization and</td>
<td>network(1M)</td>
</tr>
<tr>
<td>raw: raw network protocol family.</td>
<td>raw(7P)</td>
</tr>
<tr>
<td>routed: network routing daemon.</td>
<td>routed(1M)</td>
</tr>
<tr>
<td>newshost: NeWS network security control.</td>
<td>newshost(1)</td>
</tr>
<tr>
<td>netstat: show network status.</td>
<td>netstat(1)</td>
</tr>
<tr>
<td>networking: introduction to networking facilities.</td>
<td>netintro(7)</td>
</tr>
<tr>
<td>networking facilities. networking: introduction to</td>
<td>netintro(7)</td>
</tr>
<tr>
<td>networks: network name data base.</td>
<td>networks(4)</td>
</tr>
<tr>
<td>X: a portable, network-transparent window system.</td>
<td>x(1)</td>
</tr>
<tr>
<td>creat: create a new file or rewrite an existing one.</td>
<td>creat(2)</td>
</tr>
<tr>
<td>newgrp: log in to a new group.</td>
<td>newgrp(1)</td>
</tr>
<tr>
<td>newgrp: log in to a new group.</td>
<td>sh(1)</td>
</tr>
<tr>
<td>fork: create a new process.</td>
<td>fork(2)</td>
</tr>
<tr>
<td>efficient way. vfork: spawn new process in a virtual memory</td>
<td>vfork(2)</td>
</tr>
<tr>
<td>sproc: create a new share group process.</td>
<td>sproc(2)</td>
</tr>
<tr>
<td>taskcreate: create a new task.</td>
<td>taskcreate(3P)</td>
</tr>
<tr>
<td>login: new user.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>the internet. sendmail, newaliases, mailq: send mail over</td>
<td>sendmail(1M)</td>
</tr>
<tr>
<td>for the mail aliases file. newaliases: rebuild the data base</td>
<td>newaliases(1M)</td>
</tr>
<tr>
<td>idealized surface. newwave: real-time simulation of an</td>
<td>newwave(1D)</td>
</tr>
<tr>
<td>free_barrier: barrier/ barrier, file. newform: change the format of a text</td>
<td>newform(1)</td>
</tr>
<tr>
<td>newgrp: log in to a new group.</td>
<td>newgrp(1)</td>
</tr>
<tr>
<td>newgrp: log in to a new group.</td>
<td>sh(1)</td>
</tr>
<tr>
<td>psio: NeWS buffered input/output package.</td>
<td>psio(3)</td>
</tr>
<tr>
<td>start up the sgi X server as a NeWS client. xstart:</td>
<td>xstart(1)</td>
</tr>
<tr>
<td>news: print news items.</td>
<td>news(1)</td>
</tr>
<tr>
<td>newshost: NeWS network security control.</td>
<td>newshost(1)</td>
</tr>
</tbody>
</table>
silicon graphics

psh: NeWS PostScript shell.
news: print news items.

psview: PostScript previewer for
a login shell's environment to the
NeWS. NeWS server. exportnews: Pass
exportnews(1)

psterm: NeWS terminal emulator.
pstern(1)

control...
newshost: NeWS network security
newshost(1)

/sns: generate a string for the
NEWSSERVER environment variable.
setnewshost(1)

getmsg: get
next message off a stream.
getmsg(2)

/fetch, store, delete, firstkey,
nexkey: data base subroutines.
nextkey(2)
nice: change priority of a process.
nice(2)
nice: run a command at low priority.
nice(1)
nice: run low priority process.
csh(1)

nl: line numbering filter.
nl(1)
nlist: get entries from name list.
nlist(3x)

files.
nm: name list dump of MIPS object
nm(1)

hangup and quits.
nohup: run a command immune to
nokrnl(1)

hangups.
nohup: run command immune to
nokrnl(1)

noiconlogin: login process control.
noiconlogin(5)

console login program. visuallogin

noiconlogin: select and control

siglongjmp, _setjmp, _longjmp:
non-local gotos. /sigsetjmp,

jello: simulates
nonrigid body dynamics.
jello(6D)

nvram, sgkopt: get or set
non-volatile RAM variable(s).
nvram(1M)

to acquire a semaphore, and fails if
not possible. uscpsem: attempts

xcalendar: calendar with a
notebook for X11.

relnotes: on-line release
notes viewer.
relnotes(1)

mailbox: mail
notification.
mailbox(1)
of a process.
npri: modify the scheduling priority

seed48, /drand48, erand48, jrand48,

constructs. deroff: remove
interactively.
deroff/deroff, tbl, and eqn

nslookup: query name servers
nslookup(1C)

host and network byte/ htonl, htonl,

and network/ htonl, htons, ntohl,

:: null command.

nul: the
null file.

/ckpacct, dodisk, lastlogin, monacct,
nulldrn, prctmp, prdayl, prctact,/ acctsh(1M)
rpc: RPC program

file. linenum: line
number entries in a MIPS object

/ldlimit, ldlitem: manipulate line
number entries of a common object/

ldlseek, ldlseeks: seek to line
number entries of a section of a/

obtain the prime factors of a
number. factor:

/initstate, setstate: better random
number generator; routines for/

portmap: TCP,UDP port to RPC program
number mapper.

df: report
number of free disk blocks.
df(1)

convert string to double-precision
number. strtod, atof:
strtol(3C)
or shell script with an identifying
number. tag: tag a MIPS executable
tag(1)

f cvt, gcvt: convert floating-point
number to string. e cvt,

get rpc port
number XDR(3R).

getrpcport:

nl: line
numbering filter.
nl(1)

uniformly distributed pseudo-random
numbers. /seed48, lcong48: generate

drand48(3C)
manipulate parts of floating-point numbers. \texttt{frexpl, ldexp, modf}: \texttt{frexpl(3C)}
to system calls and error numbers. \texttt{intro}: \texttt{intro(2)}
non-volatile RAM variable(s). \texttt{nvram, sgikopt: get or set}
\texttt{ldfcn: common object file access routines.}
\texttt{ldfcn(4)}
dis: disassemble an object file. \texttt{dis(1)}
ldopen, ldaopen: open a common object file for reading. \texttt{ldopen(3X)}
line number entries of a common object file function. \texttt{ldiread(3X)}
ldclose, ldaclose: close a common object file. \texttt{ldclose(3X)}
read the file header of a common object file. \texttt{ldfread(3X)}
entries of a section of a common object file /seek to line number
the optional file header of a common object file. \texttt{ldseek(3X)}
entries of a section of a common object file /seek to relocation
section header of a common object file /read an indexed/named
an indexed/named section of a common object file. \texttt{ldseek(3X)}
of a symbol table entry of a common object file. \texttt{ldtbin(3X)}
symbol table entry of a common object file. \texttt{ldtread(3X)}
seek to the symbol table of a common object file. \texttt{ldtseek(3X)}
line number entries in a MIPS object file. \texttt{linenum(4)}
odump: dump selected parts of an object file. \texttt{odump(1)}
relocation information for a common object file: \texttt{reloc(4)}
scnhdr: section header for a MIPS object file. \texttt{scnhdr(4)}
size: print the section sizes of an object file. \texttt{size(1)}
ldgetname: retrieve symbol name for object file symbol table entry. \texttt{ldgetname(3X)}
filehdr: file header for MIPS object files. \texttt{filehdr(4)}
nm: name list dump of MIPS object files. \texttt{nm(1)}
find ordering relation for an object library. \texttt{order(1)}
/find the printable strings in an object, or other binary file. \texttt{strings(1)}
flip: spin one or more objects. \texttt{flip(6D)}
information. \texttt{dvhtool: modify and obtain disk volume header}
number. \texttt{factor:}
\texttt{dvhtool(1M)}
o钟: display time of day. \texttt{oocl(1)}
octal dump. \texttt{od(1)}
octal dump. \texttt{od(1)}
octal dump. \texttt{od(1)}
object file. \texttt{odump: dump selected parts of an}
object file. \texttt{odump(1)}
olwm: \texttt{OPEN LOOK window manager. olwm(1)}
duart: on-board serial ports. \texttt{duart(7)}
a new file or rewrite an existing
one. \texttt{create:}
several files or subsequent lines of
one file. \texttt{merge same lines of}
line: read
one line. \texttt{line(1)}
flip: spin
one or more objects. \texttt{flip(6D)}
'slave' local clock to a better
one. \texttt{timeslave:}
command scripts.
onintr: process interrupts in
\texttt{timeslave(1M)}
\texttt{csh(1)}
manual/ man: print entries from the on-line reference manuals; find
relnotes: on-line release notes viewer.
\texttt{relnotes(1)}
file including aliases and path (csh only). which: locate a program
roting: maps an image
onto a surface. \texttt{roting(6D)}
reading. \texttt{ldopen, ldaopen: open a common object file for}
\texttt{ldopen(3X)}
fopen, freopen, fdopen: open a stream.
\texttt{fopen(3S)}
driver. clone:
open any minor device on a STREAMS
\texttt{clone(7)}
dup: duplicate an open file descriptor. \texttt{dup(2)}
dup2: duplicate an open file descriptor. \texttt{dup2(3C)}

Version 5.0

-66-

April 1990
change working directory, given an open file descriptor. fchdir: fchdir(2)
or remove an advisory lock on an open file. flock: apply flock(3B)
open: open for reading or writing. open(2)
olwm: OPEN LOOK window manager. olwm(1)
oopen: open for reading or writing. open(2)
rewinddir, closedir: directory/ opendir, readdir, telldir, seekdir, directory(3C)
vsyslog: control system/ syslog, openlog, closelog, setlogmask, syslog(3B)
osview: monitor operating system activity data. osview(1)
prf: operating system profiler. prf(7)
run commands performed to stop the operating system. rc0: rc0(1M)
/savecore: save a core dump of the operating system. savecore(1M)
/bcmsg, blkclr, hzero: byte string operations. bcopy, bstring(3C)
memcpy, memstr, memccpy: memory operations. memchr, memcmp, memory(3C)
msgctl: message control operations. msgctl(2)
msgsnd, msgrcv: message operations. msgsop(2)
prctl: operations on a process. prctl(2)
taskctl: operations on a task. taskctl(3P)
semctl: semaphore control operations. semctl(2)
semop: semaphore operations. semop(2)
shmctl: shared memory control operations. shmat, shmdt: shared memory shmemctl(2) operations. shmp(2)
strl, index, rindex: string operations. /strcsnp, strto, string(3C)
rewinddir, closedir: directory operations (System V), /seekdir, directory(3C)
and lock arena configuration operations. usconfig: semaphore usconfig(3P)
usctllock: lock control operations. usctllock(3P)
usctlsema: semaphore control operations. usctlsema(3P)
join: relational database operator. join(1)
mouse: optical mouse specifications. mouse(7)
curses: terminal screen handling and optimization package. curses(3X)
getopt: get option letter from argument vector. getopt(3C)
sgikopt: retrieve kernel option strings. sgikopt(2)
object file. ldohseck: seek to the optional file header of a common ldohseck(3X)
fcntl: file control options. fcntl(5)
statt: set the options for a terminal. stty(1)
getopt: parse command options. getopt(1)
getopts, getoptcvt: parse command options. getopt(1)
getsockopt, setsockopt: get and set options on sockets. getssockopt(2)
values between host and network byte order. /hnos, ntohl, ntohs: convert byteorder(3N)
services: sethostresorder: specify order of host-address resolution sethostresorder(3N)
library. lorder: find ordering relation for an object lorder(1)
make a directory, or a special ordinary file. mknod: mknod(2)
error. oserror, setoserror: get/set system oserror(3C)
activity data. osview: monitor operating system osview(1)
dial: establish an out-going terminal line connection. dial(3C)
a.out: assembler and link editor output. a.out(4)
fold long lines for finite width output device. fold: fold(1)
redirect: run a demo with error output directed to /dev/console. redirect(6D)
set the default monitor video output format. setmon: setmon(1)
GETRUSAGE(3), write output gathered from buffers writev:
/vfprintf, vprintf: print formatted output of a variable argument list. vfprintf(3S)
Permutated Index

cprintf, sprintf: print formatted output. printf, .............. printf(3S)
foreach: loop over list of names. .............. csh(1)
do, for: loop over list of words. .............. sh(1)
newaliases, mailq: send mail over the internet. sendmail, ......... sendmail(1M)
graphs demographic data in 3D over time. demograph: ............. demograph(6D)
exeyes: watch over your shoulder. .............. exeyes(1)
command, exec: overlay shell with specified csh(1)
command, exec: overlay shell with specified sh(1)
/acctdusg, accton, acctwtmpl: overview of accounting and/ ......... acct(1M)
and 4.3BSD). chown, fchown: change owner and group of a file (System V chown(2)
chown, chgrp: change owner or group. .............. chown(1)
information. pac: printer/plotter accounting ......... pack(1M)
expand files. pack, pcat, unpack: compress and ......... pack(1)
screen handling and optimization package. curses: terminal ......... curses(3X)
pvio: NeWS buffered input/output package. .............. psio(3)
sa2, sadc: system activity report package. sa1, .............. sar(1M)
standard input/output package. stdio .................. stdio(3S)
standard interprocess communication package. ftk: .............. stdipc(3C)
floating-point exception handler package TCSETATTR(3T). ......... handle_sigfpe:
ping: send ICMP ECHO_REQUEST packets to network hosts. ........ ping(1M)
manwhx: display a man page and then prompt for input. ......... manwhx(6D)
Window System.. xman: Manual page display program for the X xman(1)
viewing. more, page: file perusal filter for crt more(1)
getpagesize: get system page size. .............. getpagesize(2)
with/ makewhatis: make manual page "whatis" database for use makewhatis(1M)
cachetc: mark pages cacheable or uncachable. cachetc(2)
mpin, munpin: lock pages in memory. .............. mpin(2)
mmap: map pages of memory. .............. mmap(2)
munmap: unmap pages of memory. .............. munmap(2)
X window. plaid: paint some plaid-like patterns in an plaid(1)
brushes. ipaint: Paint using bitmap images as ipaint(6D)
socketpair: create a pair of connected sockets. socketpair(2)
console. pandora: login on the graphics pandora(1)
plp: parallel line printer interface. ......... plp(7)
pmake, make: create programs in parallel. .............. pmake(1)
/m Park_procs, m_rele_procs, m_wait: parallel programming primitives/ m_fork,
xsetroot: root window parameter setting utility for X. ........ xsetroot(1)
configure network interface parameters. ifconfig: ........ ifconfig(1M)
dialwarp, keywarp: set input warping parameters. mousewarp, ......... mousewarp(6D)
set: set shell flags or positional parameters. .............. sh(1)
get process, process group, and parent process IDs. /getppid: ......... getppid(2)
getopt: parse command options. .............. getopt(1)
getopts, getoptopt: parse command options. .............. getopt(1)
runcn: run a command on a particular cpu. .............. runcn(1)
root, root, usr, rusr, swap, rswap: Partition names. .......... root(7M)
odump: dump selected parts of an object file. .............. odump(1)
frexp, ldexp, modf: manipulate parts of floating-point numbers. frexp(3C)
xmag: magnify parts of the screen. .............. xmag(1)
the NeWS server. exporttonews: Pass a login shell's environment to exporttonews(1)
password attributes. passmgmt: password file management. passmgmt(1M)
password attributes. passwd: change login password and ......... passwd(1)

Version 5.0 — 68 — April 1990
functions. crypt:
  passwd: password file ............................................ passwd(4)
  passwd: change login
  passwd: change login password and
setpwnent, endpwnent, fgetpwnent: get
putpwent: write
  passwd file entry
  passwd file entry
password file management
password
getppass: read a
  password
  passwd/group file checkers
  pwck, grpck
files or subsequent lines of one/
a program file including aliases and
dirname: deliver portions of
  path (csh only). which: locate
  path names. basename
  path names from i-numbers
nccheck: generate
  ncheck(1M)
configurable pathname variables.
getwd: get current working directory
directory. getcwd: get
  pathconf, fpathconf: get
  pathconf(2)
  pathname
  getwd(3C)
  path-name of current working
  getcwd(3C)
fpathconf: get configurable
grep: search a file for a
  pattern
  grep(1)
lptest: generate lineprinter ripple
  lptest(1)
language. awk:
  awk
language. awk:
  awk(1)
egrep: search a file for a
greyscale: make different
munchner: draw interesting
plaid: paint some plaid-like
  plaid(1)
  pack, compress and expand
files. pack,
  pack(1)
pclose: initiate pipe to/from a
  pclose(3S)
pcreatelp, pcreatelvp: create a/
  pcreatel, pcreatelvp
  pcreatelv
  create a/ /pcreatel, pcreatev
  create a/ create a/
  pcreatel, pcreatev, pcreatel,
  pcreatelv, pcreatev
  pcreatel, pcreatev, pcreatelv,
  pcreatelv
  create a/ pcreatel, pcreatev,
pcreatelv, pcreatev, pcreatel,
pcreatev, pcreatel,
pcreatev
getpeername: get name of connected
  getpeername(2)
sigpending: return set of signals
  sigpending(2)
/systems. psec: extended system
  permissions file. ucheck:
  xcheck(1M)
check the uucp directories and
directory. chmod: change the
  permissions mode of a file or
  chmod(1)
msg:
  permit or deny messages
  msg(1)
acct:
  per-process accounting file format
  acct(4)
acctcms: command summary from
sys_errlist, sys_nerr: system error/
  per-process accounting records
  acctcms(1M)
more, page: file
  perror(3C)
  perusal filter for crt viewing
  more(1)
Permutated Index

pg: file  perusal filter for CRTs. .. . . . . . pg(1)
pg: file  perusal filter for CRTs. .. . . . . . pg(1)
tcsetgrp (int fildes, pid_t grpid); TCSENDBREAK(3T).  .. . . . . . int
newton: a physical modeling demo. .. . . . . . newton(6D)
across a network. dlgnewton: a physical modeling demo running .. . . dlgnewton(6D)
msync: synchronize memory with physical storage. .. . . . . . msync(2)
tcsetgrp (int fildes, pid_t pgid_id); TCSENDBREAK(3T).  .. . . . . . int
shadow: full-screen armchair pilot's view of the dogfight. .. . . shadow(6D)
to network hosts. ping: send ICMP ECHO_REQUEST packets ping(1M)
channel. pipe: create an interprocess .. . . . . . pipe(2)
mknod: build special file or named pipe (FIFO). .. . . . . . mknod(1M)
tee: pipe fitting. .. . . . . . tee(1)
popen, pclose: initiate pipe to/from a process. .. . . . . . open(3S)
in a window. mag: pixel replication and magnification .. . . mag(6D)
imgexp: expand the range of pixel values in an image. .. . . . . . imgexp(6D)
program. pixie: add profiling code to a .. . . . . . pixie(1)
pixstats: analyze program execution. .. . . . . . pixstats(1)
patterns in an X window. plaid: paint some plaid-like .. . . plaid(1)
plaid: paint some plaid-like .. . . . . . plaid(1)
/pl4Sight event recording and playback from an IRIX shell. .. . . . journalplay(1)
4/3Sight event record and playback toolchest. journalchst: .. . . journalchst(1W)
in memory. plock: lock process, text, or data .. . . . . . plock(2)
interface. plp: parallel line printer .. . . . . . plp(7)
parallel. pmake, smake: create programs in .. . . . . . pmake(1)
pnhc: file format for card images. .. . . . . . pnhc(4)
selecting X11 font names. xfontsel: point & click interface for .. . . xfontsel(1)
rewind, ftell: reposition a file pointer in a stream. fseek, .. . . fseek(3S)
report on the screen under the mouse pointer. snoop: magnify and .. . . snoop(6D)
lsseek: move read/write file pointer (System V and 4.3BSD). .. . . lsseek(2)
extract FORTRAN-callable entry points from a C file. extcentry: .. . . extcentry(1)
poll: input/output multiplexing. .. . . . . . poll(2)
ico: animate an icoshedron or other polyhedron. .. . . . . . ico(1)
fauccet dripping into a multi-colored pool of liquid. liquid: A .. . . . . . liquid(6D)
popl: pop shell directory stack .. . . . . . pl(1)
for a process. popd: pop shell directory stack .. . . . . . pl(1)
get rpc popen, pclose: initiate pipe to/from .. . . . . . open(3S)
portmap: TCP,UDP port number XDR(3R). .. . . . . . getrpcport:
data base of terminal types by portable, network-transparent window .. . . x(1)
system: X: a portable network-transparent window .. . . x(1)
file. snapshot: save a portion of the screen in an image .. . . snapshot(6D)
basename, dirname: deliver portions of path names. .. . . . . . basename(1)
number mapper. portmap: TCP,UDP port to RPC program .. . . portmap(1M)
duart: on-board serial port to RPC program .. . . portmap(1M)
set: shell flags or positional parameters. .. . . . . . sh(1)
speed_t/ cfsetospeed, cfsetispeed: posix baud rate primitives #include .. . . cfgetispeed,
group primitives/ tcsetgrp: posix get/set foreground process .. . . tcgetispeed,
primitives #include int/ tegetattr: posix get/set terminal state .. . . tcgetATTR,
#include/ tcdrain, tcflush, tcflo: posix line control primitives .. . . tcsetendbreak,
software signal facilities (POSIX). sigaction: .. . . . . . sigaction(2)
set of signals pending for process (POSIX). sigpending: return .. . . sigpending(2)
state of the set of blocked signals
signals and wait for interrupt
get configurable system variables
termios: general System V and
and examination routines
a semaphore, and fails if not
banner: make
cps: construct C to
psview: PostScript interface.
say: execute
ps: NeWS
exp, expm1, log, log10, log1p,
flog1p: exponential, logarithm,
halt the system.

/lastlogin, monacct, nulladm, prctmp,
xset: user
vmsprep: VMS tape
monitor, monstartup, moncontrol:
cpp: the C language
utility. make: C
a pristine state by deleting/
Graphics demos. buttonify: a
accept, reject: allow or
psview: PostScript
unget: undo a
sigprocmask: alter and return
previously.

(POSIX). /alter and return previous
(POSIX). atomically release blocked
(POSIX). sysconf:
POSIX terminal interfaces. termios:
(POSIX), with SGIs. /set manipulation
possible. /attempts to acquire
posters. 
(POSIX) interface.
PostScript previewer for NeWS.
PostScript.
PostScript shell.
pow, fexp, fexpm1, flog, flog10/
power. /fexp, fexpm1, flog, flog10,
powerdown: stop all processes and
pr: print files.
prctl: operations on a process.
prompt, prdaily, prtacct, runacct/
prdaily, prtacct, runacct, shutacct/
preference utility for X.
preparation aid.
prepare execution profile.
monitor (3X)
preprocessor interface to the make
preset: reset the lp queue system to
pretty user interface for Silicon
prevent LP requests.
prefix point of an SCCS file.
previous state of the set of blocked/
prof: operating system profiler.
profddc, prfssnap, prfrp: UNIX system
profld, prfstat, prfd, prfssnap,
prfrp: UNIX system profiler.
prfrp: UNIX system profiler.
prfrp: UNIX system profiler.
prfrp: UNIX system profiler.
profile factors of a number.
primitive data types.
/int/ /posix get/set terminal state
/get/get foreground process group
/tcflush, tcflow: posix line control
/cfsetspeed: posix baud rate
times:
prs: print all SCCS files.
xpr: print an X window dump.
date: print and set the date.
cal: print calendar.
file. sum: print checksum and block count of a
shownf: print contents of an SNF file.
xev: print contents of X events.
activity. sact: print current SCCS file editing

April 1990
-71-
Version 5.0
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>whoami</td>
<td>print effective current user id.</td>
</tr>
<tr>
<td>cat</td>
<td>concatenate and print files.</td>
</tr>
<tr>
<td>pr</td>
<td>print files.</td>
</tr>
<tr>
<td>vprintf, vpprintf, vsprintf</td>
<td>print formatted output of a variable</td>
</tr>
<tr>
<td>printf, fprintf, sprintf</td>
<td>print formatted output.</td>
</tr>
<tr>
<td>history</td>
<td>print history event list.</td>
</tr>
<tr>
<td>system, hostid</td>
<td>set or print identifier of current host</td>
</tr>
<tr>
<td>log</td>
<td>print log messages and other information about RCS files.</td>
</tr>
<tr>
<td>lpsstat</td>
<td>print LP status information.</td>
</tr>
<tr>
<td>lpr</td>
<td>print.</td>
</tr>
<tr>
<td>hostname</td>
<td>set or print name of current host system.</td>
</tr>
<tr>
<td>news</td>
<td>print news items.</td>
</tr>
<tr>
<td>infoconf: compare or print out terminfo descriptions.</td>
<td>infoconf(1M)</td>
</tr>
<tr>
<td>printenv</td>
<td>print out the environment.</td>
</tr>
<tr>
<td>acctcom</td>
<td>search and print process accounting file(s).</td>
</tr>
<tr>
<td>sysex</td>
<td>print system identification.</td>
</tr>
<tr>
<td>list of image files..</td>
<td>istat: print the header information of a disassemble a MIPS instruction and file.</td>
</tr>
<tr>
<td>disasm</td>
<td>print the results. disassembler: print the section sizes of an object</td>
</tr>
<tr>
<td>spray</td>
<td>print the symbol table.</td>
</tr>
<tr>
<td>id</td>
<td>print user and group IDs and names.</td>
</tr>
<tr>
<td>prtvoc</td>
<td>print volume header information.</td>
</tr>
<tr>
<td>other binary/</td>
<td>printable strings in an object, or printcap: printer capability data</td>
</tr>
<tr>
<td>strings: find the base.</td>
<td>printenv: print out the environment.</td>
</tr>
<tr>
<td>lpc</td>
<td>printer capability data base.</td>
</tr>
<tr>
<td>lpc: line</td>
<td>printer control program.</td>
</tr>
<tr>
<td>lpd: line</td>
<td>printer daemon.</td>
</tr>
<tr>
<td>rmprinter: remove a printer from the LP spooling system.</td>
<td>rmprinter(1M)</td>
</tr>
<tr>
<td>plp: parallel line</td>
<td>printer interface.</td>
</tr>
<tr>
<td>send/cancel requests to an LP line</td>
<td>printer. lp, cancel:</td>
</tr>
<tr>
<td>mknetpr: provide access to a remote printer.</td>
<td>mknetpr(1M)</td>
</tr>
<tr>
<td>routeprint: route file to printer.</td>
<td>routeprint(1)</td>
</tr>
<tr>
<td>lpm: remove jobs from the line printer spooling queue.</td>
<td>lpm(1)</td>
</tr>
<tr>
<td>mkPS: register a LaserWriter printer with LP. mkcentr: register</td>
<td>mkcentr(1M)</td>
</tr>
<tr>
<td>xdpr: dump an X window directly to a xprint, printer accounting</td>
<td>xprint(1M)</td>
</tr>
<tr>
<td>enable, disable</td>
<td>enable/disable LP to a pristine state by deleting formatted output.</td>
</tr>
<tr>
<td>to a pristine state by deleting</td>
<td>printers. /reset the lp queue system</td>
</tr>
<tr>
<td>addclient: allow remote printing clients to connect.</td>
<td>xprint(3S)</td>
</tr>
<tr>
<td>statistics/ handle_unaligned_traps, priority, getpriority, setpriority:</td>
<td>gather_unaligned_summary: gather</td>
</tr>
<tr>
<td>nice: run a command at low priority.</td>
<td>nice(1)</td>
</tr>
<tr>
<td>nice: change priority of a process.</td>
<td>nice(2)</td>
</tr>
<tr>
<td>npri: modify the scheduling priority of a process.</td>
<td>npri(1)</td>
</tr>
<tr>
<td>renice: alter priority of running processes.</td>
<td>renice(1M)</td>
</tr>
<tr>
<td>nice: run low priority process.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>/reset the lp queue system to a pristine state by deleting printers.</td>
<td>preset(1M)</td>
</tr>
</tbody>
</table>
rutines. rpc: Remote
   procedure/ ldgetpd: retrieve
   /procedure descriptor given a
broc, bcheckr: system initialization
shutacct, startup, turnacct: shell
facilitate better/ cord: rearranges
defined/ _procedure_table_size,
edate, eprom, _fext, _data, _fbss,
_fdata, _fbss, _procedure_table,
   kill: send signal to a
block signals from delivery to
acct: enable or disable
acctprc1, acctprc2:
acctcom: search and print
   alarm: set a
times: get
   init, telinit:
noiconlogin: login
visuallogin: login
   nice: run low priority
timex: time a command; report
   exit, _exit: terminate
fork: create a new
killpg: send signal to a
ID's. /getgpgr, getppid: get process,
setpgid: set
4.3BSD), setgpgr, BSDsetgpgr: set
setsid: create session and set
tcsetgpgr: posix get/set foreground
process, process group, and parent
efficient way. vfork: spawn new
   initab: script for the init
scripts. onintr:
   scripts. trap:
k ill: terminate a
   nice: change priority of a
modify the scheduling priority of a
   kill: send a signal to a
pcreatelp, pcreatev: create a
pclose: initiate pipe to/from a
return set of signals pending for
   prctl: operations on a
getpid, getgpgr, getppid: get
   sproc: create a new share group
   p: report
   plock: lock
times: get process and child
   ptrace: process trace.
pause: suspend
wait: await completion of
   abort: terminate current

Procedure Call (RPC) library
   procedure descriptor given a
   procedure descriptor index.
   procedures.
   procedures for accounting. /runacct,
   procedures in an executable to
   _procedure_string_table, _gp: loader,
   _procedure_table, end, etext,
   _procedure_table_size, /_fext,
   process (4.3BSD).
   process (4.3BSD). sigblock:
   process accounting.
   process accounting.
   process accounting file(s).
   process alarm clock.
   process and child process times.
   process control initialization.
   process.
   process data and system activity.
   process.
   process.
   process group (4.3BSD).
   process group, and parent process
   process group ID.
   process group ID (System V and
   process group IDs.
   process group primitives #include/
   process IDs. /getgpgr, getppid: get
   process in a virtual memory
   process.
   process interrupts in command
   process interrupts in command
   process.
   process.
   process.
   process.
   process.
   process or a group of processes.
   process.
   process.
   process.
   process.
   process.
   process.
   process.
   process.
   process.
   process.
   process.
   process.
   process.
   process.
   process.
   process.
powerdown: stop all processes and halt the system. powerdown(1M)

routines to block/unblock processes. /setblockprocentall: blockproc(2)

kill: kill jobs and processes. csh(1)
in a window. gr_top: display processes having highest CPU usage. gr_top(1)
top: display processes having highest CPU usage. top(1)
a signal to a process or a group of processes. kill: send kill(2)
killall: kill named processes. killall(2)
enice: alter priority of running processes. csh(1)
wait: wait for background processes to complete. sh(1)
wait: wait for background processes to complete. renice(1M)
wait, waitpid, wait3: wait for child processes to stop or terminate. wait(2)
structure. fuser: identify processes using a file or file. fuser(1M)
awk: pattern scanning and awk(1)
awk: pattern scanning and processing language. awk(1)
nawk: pattern scanning and processing language. nawk(1)
m4: macro processor. m4(1)

mpadmin: control and report processor status. mpadmin(1)
m68k, m68000, mips, 4d, 4d60: get processor type truth value. m68k(1)
sginap: timed sleep and processor yield function. sginap(2)

ftoc: interface between prof: analyze profile data. prof(1)
prof: analyze prof and cord. prof(1)

moncontrol: prepare execution profile. monitor, monstartup, monitor(3X)
profile. execution time profile. profile. prof(2)
profile. execute time. profil(2)
at login time. profil(2)
profile: setting up an environment profil(2)
prf: operating system profil(2)
prf(7)

prfch, prfsnap, prfpr: LINUX system profil(1M)
profiler. prld, prfstat, profiler(1M)

pixie: add profiling code to a program. pixie(1)
maze: an automated maze maze(1)

program... [ demo ][ X11 ]. maze(1)

program beautifier. cb(1)

lint: a C program checker. lint(1)

cxref: generate C program cross-reference. cxref(1)
ctrace: C program debugger. ctrace(1)
dgfax: electronic fax program. dglfax(1)

send signal to executing program. DIFFTIME(3C).

_arg: loader defined symbols in a program. /_procedure_string_table, end(3C)
pixstats: analyze program execution. pixstats(1)
path (csh only). which: locate a program file including aliases and
which(1)
finger: user information lookup program. finger(1)
uucico: file transport program for the uucp system. uucico(1M)
xman: Manual page display program for the X Window System. xman(1)
xhost: server access control program for X. xhost(1)

ftp: Internet file transfer program. ftp(1C)

lpc: line printer control program. lpc(1M)

lpq: spool queue examination program. lpq(1)

mt: magnetic tape manipulating program. mt(1)

e: program number data base. rpc(4)

e: program number mapper. portmap(1M)

pixie: add profiling code to a program. pixie(1)

rdist: remote file distribution program. rdist(1C)

getpriority, setpriority: get/set

program scheduling priority. getpriority(2)
<table>
<thead>
<tr>
<th>Silicon Graphics</th>
<th>Permutated Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdiff: side-by-side difference</td>
<td>program.</td>
</tr>
<tr>
<td>slides: slide display</td>
<td>program.</td>
</tr>
<tr>
<td>results of a finite element analysis</td>
<td>program. solidview: display the</td>
</tr>
<tr>
<td>tftp: trivial file transfer</td>
<td>program. tftp(1C)</td>
</tr>
<tr>
<td>timedc: timed control</td>
<td>program. timedc(1M)</td>
</tr>
<tr>
<td>unis: conversion for the uucp file transport</td>
<td>program. unisched: the scheduler</td>
</tr>
<tr>
<td>assert:</td>
<td>program verification.</td>
</tr>
<tr>
<td>select and control console login</td>
<td>program. visuallogon, noiconlogin:</td>
</tr>
<tr>
<td>source, binary, and or manual for</td>
<td>program. wherever: locate</td>
</tr>
<tr>
<td>X window system message display</td>
<td>program. xmessage:</td>
</tr>
<tr>
<td>commands, application programs, and the</td>
<td>programming commands., /to</td>
</tr>
<tr>
<td>standard/restricted command</td>
<td>programming language. /rsh: shell,</td>
</tr>
<tr>
<td>/m_rele_procs, m sync: parallel</td>
<td>programming primitives/</td>
</tr>
<tr>
<td>/to commands, application</td>
<td>m_fork,</td>
</tr>
<tr>
<td>lex: generate</td>
<td>programs, and programming commands.. intro(1)</td>
</tr>
<tr>
<td>pmake, smake: create maintenance commands and</td>
<td>programs in parallel.</td>
</tr>
<tr>
<td>application</td>
<td>pmake(1)</td>
</tr>
<tr>
<td>update, and regenerate groups of xstr: extract strings from C</td>
<td>programs. make: maintain,</td>
</tr>
<tr>
<td>manwsh: display a man page and then</td>
<td>programs to implement shared/</td>
</tr>
<tr>
<td>xprop: property display for X.</td>
<td>xprop(1)</td>
</tr>
<tr>
<td>arp: Address Resolution</td>
<td>proto: prototype job file for at.</td>
</tr>
<tr>
<td>bootp: server for Internet Bootstrap</td>
<td>proto(4)</td>
</tr>
<tr>
<td>rpcgen: an RPC</td>
<td>protocol compiler.</td>
</tr>
<tr>
<td>x10to11: X version 10 to version 11</td>
<td>protocol compiler.</td>
</tr>
<tr>
<td>setprotoent, endprotoent: get</td>
<td>protocol converter.</td>
</tr>
<tr>
<td>inet: Internet</td>
<td>protocol entry. /getprotobyname, getprotoent(3N)</td>
</tr>
<tr>
<td>raw: raw network</td>
<td>protocol family.</td>
</tr>
<tr>
<td>icmp: Internet Control Message</td>
<td>raw(7P)</td>
</tr>
<tr>
<td>ip: Internet</td>
<td>icmp(1P)</td>
</tr>
<tr>
<td>rmt: remote magtape</td>
<td>Protocol.</td>
</tr>
<tr>
<td>protocols:</td>
<td>ip(7P)</td>
</tr>
<tr>
<td>ftpd: Internet File Transfer</td>
<td>protocol family.</td>
</tr>
<tr>
<td>telnetd: Internet TELNET</td>
<td>protocols(1)</td>
</tr>
<tr>
<td>Internet Trivial File Transfer</td>
<td>protocol server. ftpd:</td>
</tr>
<tr>
<td>snoop: network monitoring</td>
<td>protocol server. telnetd(1M)</td>
</tr>
<tr>
<td>tcp: Internet Transmission Control</td>
<td>protocol. snoop(1P)</td>
</tr>
<tr>
<td>telnet: User interface to the TELNET</td>
<td>tcp(7P)</td>
</tr>
<tr>
<td>udp: Internet User Datagram</td>
<td>Protocol.</td>
</tr>
<tr>
<td>xscope: X Window</td>
<td>Protocol Viewer.</td>
</tr>
<tr>
<td>capture unimplemented link-layer</td>
<td>Protocol.</td>
</tr>
<tr>
<td>proto: prototype job file for at.</td>
<td>xscope(1)</td>
</tr>
<tr>
<td>interface to/ stio: routines that provide a binary read/write</td>
<td>protocols. drain:</td>
</tr>
<tr>
<td>table/ stcu: routines that provide a compilation unit symbol</td>
<td>drain(7P)</td>
</tr>
<tr>
<td>basic functions/ stfe: routines that provide a high-level interface to</td>
<td>protocols. protocols(1)</td>
</tr>
<tr>
<td>selecting entries/ transfermanager:</td>
<td>provide a visual interface for</td>
</tr>
<tr>
<td>mknethdr: provide access to a remote printer.</td>
<td>transfermanager(1G)</td>
</tr>
<tr>
<td>descriptor/ stdf: routines that provide access to per file</td>
<td>mknethdr(1M)</td>
</tr>
</tbody>
</table>

April 1990 –75– Version 5.0
| labelit: provide labels for file systems. | labelit(1M) |
| auxiliaries. staux: routines that provide scalar interfaces to staux(3X) |
| true, false: provide truth values. | true(1) |
| prs: print an SCCS file. | prs(1) |
| /monacct, nulladm, prctmp, prdaily, prtacct, runacct, shutacct, startup/ acctsh(1M) |
| information.. prvtoc: print volume header prvtoc(1M) |
| pty: | pty(7M) |
| generate uniformly distributed pseudo-random numbers. /icon48: drand48(3C) |
| messages. psh: NeWS PostScript shell. | psh(1) |
| package. psignal, sys_siglist: system signal psignal(3C) |
| psio: NeWS buffered input/output psio(3) |
| pterm: NeWS terminal emulator. | pterm(1) |
| NeWS. pview: PostScript previewer for | pview(1) |
| ptrace: process trace. | ptrace(2) |
| pty: pseudo terminal driver. | pty(7M) |
| copy. uuto, uupick: public UNIX-to-UNIX system file | uuto(1C) |
| stream. ungetc: push character back into ungetc(3S) |
| pushd: push shell directory stack. | pushd(1) |
| puts, fputs: put a string on a stream. | puts(3S) |
| putc, putchar, fpout, putw: put character or word on a stream. | putc(3S) |
| getdents: read directory entries and getdents(2) |
| character or word on a stream. put in a file system independent/ putc(3S) |
| or word on a stream. putc, putchar, fpout, putw: put character putc(3S) |
| environment. putenv: change or add value to | putenv(3C) |
| putmsg: send a message on a stream. putmsg(2) |
| stream. putpwntr: write password file entry. | putpwent(3C) |
| puts, fputs: put a string on a | puts(3S) |
| pututline, setutent, endutent, | getut(3C) |
| real-time display of famous cube puzzle. cube: cube(6D) |
| puzzle: puzzle game for X. | puzzle(1) |
| puzzle: puzzle game for X. | puzzle(1) |
| checkers. pwck, grpck: password/group file | pwck(1M) |
| pwd: working directory name. | pwd(1) |
| qsort: quicker sort. | qsort(3C) |
| nslookup: query name servers interactively. | nslookup(1C) |
| tput: initialize a terminal or tput(1) |
| queuedefs: at/batch/cron queuedefs(4) |
| spool: spool spool(1) |
| queue examination program. | spool(1) |
| jobs from the line printer spooling queue. | spool(1) |
| msgget: get message msgget(2) |
| memory id. ipcm: remove a message ipcm(1) |
| deleting/ preset: reset the lp queue system to a pristine state by queue(1M) |
| queue USGGETINFO(3P) queue(1M) |
| description file. queuedefs: at/batch/cron queue queue(4) |
| queuing: insert/remove element from a qsort: quicker sort. qsort(3C) |
| int tcflush (int fildes, int queu-select): ./int fildes); int |
| qsort: quicker sort. | qsort(3C) |
| run a command immune to hangups and quits. nohup: nohup(1) |
| gview: viewer for radiosity data. gview(6D) |
remove a file
sgikopt: get or set non-volatile
interp: gamma-corrected color
generator.
random, initstate, setstate: better
about resource utilization
rand, srand: simple
imgexp: expand the
for the symbol table/ranhashinit,
access routine for the symbol table/
symbol table/ranhashinit, ranhash,
/cfsetospeed, cfsetispeed: posix baud
raw:
flyray: a visualized
dggray: a visualized
the operating system.
multi-user environment.
for returning a stream to a remote/
/transferrdevice for performing rcp/
/rcsintro: introduction to RCS
rcs: change RCS
rcsfile: format of
messages and other information about
cli: check in
co: check out
rcsdiff: compare
rcsmerge: merge
check: check
commands.
program.
getpass:
input.
of a common object file. ldthread:
of a common/ ldshread, ldnshread:
source:
file system independent/ getdents:
read:
WRITEV(3C).
mail: send mail to users or
line:
of an archive file. ldthread:
object file. ldthread:

RAISE(3C),........remove:
RAM variable(s), nvram,........nvram(1M)
interp generator........interp(6D)
rand, srand: simple random-number........rand(3C)
random number generator; routines........random,
RANDOM(3B). get information........getrusage:
random-number generator........rand(3C)
range of pixel values in an image........imgexp(6D)
ranchash, ranlookup: access routine........ranhash(3X)
ranchashinit, ranhash, ranlookup........ranhash(3X)
ranchashinit: access routine for the........ranhash(3X)
rate primitives #include speed_t........cfgetospeed,
raw network protocol family........raw(7P)
raw: raw network protocol family........raw(7P)
raytracer........flyray(6D)
raytracer running across a network........dggray(6D)
rc0: run commands performed to stop........rc0(1M)
rc2: run commands performed for........rc2(1M)
rcmd, tressvport, ruserok: routines........rcmd(3N)
rcp: remote file copy........rcp(1C)
rcp within the Workspace........rcpdevice()
rcpDevice: an interactive........rcpdevice()
rcs: change RCS file attributes........rcs(1)
rcsintro: introduction to RCS........rcsintro(1)
rcs file attributes........rcs(1)
rcsfile(4)
RCS files. rlog: print log........rlog(1)
RCS revisions........ci(1)
RCS revisions........co(1)
RCS revisions........rcsdiff(1)
RCS revisions........rcsmerge(1)
RCS status of a file........check(1)
rcsdiff: compare RCS revisions........rcsdiff(1)
rcsfile(4)
rcsintro: introduction to RCS........rcsintro(1)
rcsmerge: merge RCS revisions........rcsmerge(1)
rdist: remote file distribution........rdist(1C)
read a password........getpass(3C)
read: accept input from the standard........sh(1)
read an indexed symbol table entry........ldtbread(3X)
read an indexed/named section header........ldthread(3X)
read commands from file........csh(1)
read commands from file........sh(1)
read directory entries and put in a........getdents(2)
read from file........read(2)
read input to scattered buffers........readv:
read mail........mail(1)
read one line........line(1)
read from file........read(2)
read the archive header of a member........ldthread(3X)
read the file header of a common........ldthread(3X)
Permutied Index

readlink: read value of a symbolic link. ... readlink(2)
readir, closedir, opendir, ... directory(3C)
rewindir, closedir, dirfd:/ ... opendir, seekdir,
compatible/ tabled: tabled reader daemon for Bitpad I ... tabled(1M)
open a common object file for reading. Idopen, Idopen: ... Idopen(3X)
open: open for reading or writing. ... open(2)
read-only: readonly: make shell variables ... sh(1)
readonly: make shell variables ... sh(1)
group access list (bsd 4.3 version) ... initgroups:
READV(3C). initialize ... lseek(2)
and 4.3BSD). lseek: move ... lseek(2)
sto: routines that provide a binary ... stio(3X)
setregid: set real and effective group ID. ... setregid(2)
setreuid: set real and effective user ID's. ... setreuid(2)
/get real user, effective user, ... getuid(2)
/geteuid, getgid, getegid: get real user, effective user, real/
allocator. malloc, free, ... malloc(3C)
fast main memory/ malloc, free, ... malloc(3X)
/puzzle. cube: real-time display of famous cube ... cube(6D)
l... light(6D)
surface. newave: real-time simulation of an idealized ... newave(1D)
of an idealized waterbed. ... wave(6D)
execute to facilitate/ cord: rearranges procedures in an ... cord(1)
reboot: reboot the system. ... sh(1)
reboot: reboot the system. ... sh(1)
aliases file. newaliases: rebuild the data base for the mail ... newaliases(1M)
recv, recvfrom, recvmsg: receive a message from a socket. ... recv(2)
Mail: send and receive mail. ... mail(3B)
r... receive mail via UUCP. ... mail(3B)
handler. re_comp, re_exec: regular expression ... regex(3B)
rehash: recompute command hash table. ... csh(1)
journallow: 4Sigh event record and playback toolchest. ... journallow(1W)
lockf: record locking on files. ... lockf(3C)
shell. /journalendar: 4Sigh event recording and playback from an IRIX ... journalplay(1)
summary from per-process accounting records. acctcmsg: command ... acctcmsg(1M)
manipulate connect accounting records. ftmp, wtmpfix: ... ftmp(1M)
message from a socket. recv, recvfrom, recvmsg: receive a ... recv(2)
from a socket. recv, recvfrom, ... recv(2)
recvmsg: receive a message from a ... recv(2)
ed: ... ed(1)
output directed to /dev/console. redirect: run a demo with error ... redirect(6D)
unifddf: strip or reduce ifdef in C code. ... unifydf(1)
eval: re-evaluate shell data. ... csh(1)
eval: re-evaluate shell data. ... sh(1)
re_exec: regular expression handler. ... regex(3B)
man: print entries from the on-line ... man(1)
gather statistics on unaligned reference manuals; find manual/ ... unaligned(3X)
xrefresh: refresh all or part of an X screen. ... xrefresh(1)
regular expression. ... regcmp(3X)
regcmp, regex: compile and execute ... regcmp(3X)
regcmp: regular expression compile. ... regcmp(1)
make: maintain, update, and regenerate groups of programs. ... make(1)

Version 5.0
April 1990
expression. regexp, and match routines.
Centronics-interface/mkcentpr
LP. mkPS:
address for workstation.
swapINX: floating-point control routines. regexp:
re_comp, re_exec:
regcmp, regexp: compile and execute regular
registenhost: allocate internet
servers. /set_fpc_led, swapRM,
regexp: regular expression compile and match
regexp:
regexp: regular expression compile.
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
regexp:
Permutated Index

unalias: remove aliases.
file. flock: apply or remove an advisory lock on an open file
unlink: remove directory entry.
unsetenv: remove environment variables.
rm, mdir: remove files or directories.
spooling queue. lprm: remove jobs from the line printer
constructs. deroff: remove nroff/troff, tbl, and eqn
strip: remove symbols and relocation bits.
hosts data base. unregisterhost: remove the existing host entry in yp
freelist: get lvtab file entry REMOVE(3C).
queue USGETINFO(3P).
rename: change the name of a file.
hosts data base. renamehost: rename the existing hostname in yp
hostname in yp hosts data base. renamehost: rename the existing hostname
processes.
fsck, fsck: check and repair file systems.
while: repeat commands conditionally.
until: repeat commands conditionally.
repeat: execute command repeatedly.
uniq: report repeated lines in a file.
repeat: execute command repeatedly.
window. mag: pixel replication and magnification in a
clock: report CPU time used.
fsstat: report file system status.
facilities status. ipc: report inter-process communication
 df: report number of free disk blocks.
pointer. snoop: magnify and report on the screen under the mouse
sa1, sa2, sadc: system activity report package.
activity. timex: time a command; report process data and system
       ps: report process status.
       mpadmin: control and report processor status.
       uniq: report repeated lines in a file.
sar: system activity reporter.
stream. fseck, rewind, fill: reposition a file pointer in a
routines M_FORK(3P). external data representation (xdr) library
display a message in a window and request a response. confirm:
accept, reject: allow or prevent LP requests.
start/stop the LP scheduler and move requests. lpsched, lpsht, lpmove: lpsched(1M)
lp, cancel: send/cancel requests to an LP line printer.
uuxqt: execute remote command requests.
utility to launch applications that require a terminal emulator.
pristine state by deleting/ preset:
/res_search, res_mkquery, res_send, reset the lp queue system to a
terminal settings to current window/
  res_init, dn_comp, dn_expanding/
  resize: utility to set TERMCAP and
  res_mkquery, res_send, res_init,
  res_query, res_search,
  hostname: host name resolution description.
  arps: address resolution display and control.
ftime: control clock and timer resolution.
arp: Address Resolution Protocol.
/specify order of host-address resolution services.

Version 5.0

-80-

April 1990
resolver: host-address configuration file
resolver: host-address resolver
resInit, dn_comp, dn_expansion: resolver routines. /res_send
setrlimit: control maximum system resource. getrlimit
appres: list application resource database.
appres(1)
xrdb: X server resource database utility.
usvsema: frees a resource to a semaphore.
getinformation about resource utilization RANDOM(3B).
getrusage:
xkill: kill a client by its X resource.
listres: list resources in widgets.
xkill(1)
listres(1)
a message in a window and request a response. confirm: display
confirm(1G)
res_send, res_init, dn_comp:
res_query, res_search, res_mkquery,
res_init, dn_comp/, res_query,
res_query, res_search, res_mkquery,
volumes. lvck: check and restore consistency of logical
or directory from tape. Restore: restore the specified file
directory from tape. Restore: restore the specified file or
bru: backup and restore utility.
bru(1)
a MIPS instruction and print the results. disassembler: disassemble
program. solidview: display the results of a finite element analysis
an index. ldgetaux: retrieve an auxiliary entry, given
sgikopt: retrieve kernel option strings.
sigkopt(2)
a procedure descriptor/ ldgetpd: retrieve procedure descriptor given
symbol table entry. ldgetname: retrieve symbol name for object file
abs: return integer absolute value.
abs(3C)
logname: return login name of user.
logname(3X)
blocked/ sigprocmask: alter and process (POSIX). sigpending
reexec: return stream to a remote command.
sysid: return system identifier.
sysid(3C)
ustestsema: return the value of a semaphore.
ustestsema(3P)
getenv: return value for environment name.
getenv(3C)
stat: data returned by stat system call.
stat(5)
/resvport, ruserok: routines for returning a stream to a remote/
col: filter reverse line-feeds.
col(1)
ci: check in RCS revisions.
ci(1)
co: check out RCS revisions.
co(1)
rcsdiff: compare RCS revisions.
rcsdiff(1)
rcsmerge: merge RCS revisions.
rcsmerge(1)
revolve: surface of revolution demonstration.
revolve: surface of revolution
revolve(6D)
pointer in a stream. fseek.
rewind, ftell: reposition a file
rewind, closedir: directory/
rewind, closedir, dirfd:/
openDir,
readdir, telldir, seekdir,
rewinddir, closedir, dirfd:(/
openDir,
creat: create a new file or
rewrite an existing one.
creat(2)
reexec: return stream to a remote
reexec(3N)
reexecd: remote execution server.
reexecd(1M)
translates a screen image into an
RGB image. mapping.
hosts: list of trusted hosts and
rhstos(4)
strcspn, strtok, strstr, index,
rindex: string operations. /strspn,
rindex: string(3C)
rint, trunc, fnrnc: floor, ceiling/.
rfloor(3M)
April 1990 –81– Version 5.0
Permuted Index

lptest: generate lineprinter ripple pattern ................. lptest(1)
using run length encoding. rle: force an image to be stored rle(6D)
information about RCS files. rlog: print log messages and other rlog(1)
rolin: remote login. rolgin(1C)
rolind: remote login server. rolgin(1M)
directories. rm, mdir: remove files or rm(1)
mail: receive mail via UUCP. mail(1M)
file. mdel: remove a delta from an SCCS mdel(1)
mdir: remove a directory. mdir(2)
rm, mdir: remove files or directories. mdir(1)
LP spooling system. mprt: remove a printer from the
prt: remote magtape protocol module. mprt(1M)
chroot: change root directory. chroot(2)
chroot: change root directory for a command. chroot(1M)
Partition names... root, root, usr, swap, rswap, root(7M)
sqrt, fsqrt, cbrt: cube root, square
sqrt, fsqrt, cbrt: cube root, square root.
sqrt(3M)
sqrt(3M)
sqrt, fsqrt, cbrt: cube root, square root.
sqrt, fsqrt, cbrt: cube root, square root.
sqrt, fsqrt, cbrt: cube root, square root.
sqrt(3M)
sqrt(3M)
utility for X. xsedtroot: root window parameter setting xsedtroot(1)
surface. roting: maps an image onto a roting(6D)
surface.
routeprint: route file to printer. routeprint(1)
routing tables. route: manually manipulate the
route: manually manipulate the
route: manually manipulate the
route: manually manipulate the
route: manually manipulate the
route: manually manipulate the
route(1M)
routed: network routing daemon. routed(1M)
routed: network routing daemon. routed(1M)
routed: network routing daemon. routed(1M)
routed: network routing daemon. routed(1M)
routed: network routing daemon. routed(1M)
routed: network routing daemon. routed(1M)
routed: network routing daemon. routed(1M)
/ranhash, ranlookup: access
semaphore and lock initialization
/better random number generator,
remote/ rcmd, rproxy, ruserok:
Internet address manipulation
ldfcn: common object file access
data representation (xdr) library
mkf2c: generate FORTRAN-C interface
set manipulation and examination
regular expression compile and match
dn_comp, dn_expand: resolver
Remote Procedure Call (RPC) library
read/write interface to the/ stio:
unit symbol table interface. stcu:
interface to basic functions/ stfe:
file descriptor section of/ stfd:
interfaces to auxiliaries. staux:
/unblockprocall, setblockprocall:
/taskunblock, tasksetblockcnt:
strprint: routines to print the symbol table.
ustestlock, usunsetlock: spinlock
gated: gateway
mrouted: IP multicast
routed: network
hyroute: set the HyperNet
route: manually manipulate the
getrpcbyname, getrpcbyname: get
rpcinfo: report
RPC entry. getrpccent, rpcinfo(1M)
RPC information. rpcinfo(1M)
rpc: Remote Procedure Call (RPC) library routines.
get
rpc port number XDR(3R).
getrpcport:
rpc:
RPC program number data base.
portmap: TCP, UDP port to
RPC program number mapper.
RPC protocol compiler.
remote Procedure Call (RPC)
RPC program number data base.
RPC protocol compiler.
returning a stream to a/ rcmd,
report RPC information.
Partition names. root,
resvport, ruserok: routines for
rsh: remote shell.
root, usr, rusr, swap, rswap:
root(7M)
sh: shell, the standard/restricted
rshd: remote shell server.
shell(1)
utility for use with file type
rswap: Partition names.
isSuper: supertype checking
rules.
runasct:
run command at low priority.
nice:
rundasct: command immune to hangups and
runon:
run a command on a particular cpu.
nohup:
run length encoding.
run length encoding. verbatim:
nice:
rundasct: run daily accounting.
forced to be stored using
forced to be stored without
runasct: run daily accounting.
runtime: show host status of local
runasct, shutdown, startup/
rune run a command on a particular
runlow priority process.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
runtime: show host status of local
runasct: run daily accounting.
Permutated Index

of the colormap.
savemap: saves the current contents

colormap. savemap:
savemap: saves the current contents of the
say: execute PostScript.
say(1)
allocation. brk,
brk: change data segment space
sc: spreadsheet calculator.
sc(1)
status: routines that provide
scalar interfaces to auxiliaries.
status(3X)
copiesign, drem, finite, logb,
copiesign: copiesign, remainder, exponent /
copiesign(3M)
copiesign, drem, finite, logb,
copiesign: copiesign, remainder, exponent /
copiesign(3M)
scandir, alphasort:
scandir: scan a directory.
sindicdir(3C)
scanner:
scanner: scan color images.
scanner(1)
directory.
sindicdir, alphasort: scan a
scandir(3C)
formatted input.
scanf, fscanf, sscanf: convert
scanf: scanf(3S)
getinventory, setinventory, endinventory,
scandinvent: get hardware inventory/
getinventory(3)
bf: big file
scanner.
scanner: scan color images.
scanner(1)
awk:
awk: pattern
scanning and processing language.
awk(1)
hawk:
hawk: pattern
scanning and processing language.
hawk(1)
read input to
scattered buffers WRITELV(3C). readv:
change the delta commentary of an
SCCS delta. cdc:
cdc(1)
comb: combine
SCCS deltas.
comb(1)
help: ask for help about
SCCS error messages and commands.
help(1)
delta: make a delta (change) to an
SCCS file.
delta(1)
scant: print current
SCCS file editing activity.
sact(1)
get: get a version of an
SCCS file.
get(1)
print an
SCCS file.
print(1)
rmfile: remove a delta from an
SCCS file.
rmdel(1)
scsddiff: compare two versions of an
SCCS file.
scsddiff(1)
scsfile: format of
SCCS file.
scsfile(4)
unget: undo a previous get of an
SCCS file.
unget(1)
val: validate
SCCS file.
val(1)
admin: create and administer
SCCS files.
admin(1)
what: identify
SCCS files.
what(1)
colored lights bouncing around a
scene.
bounce: three
bounce(6D)
schedctl: scheduler control call.
schedctl(2)
ckbupsd: check file system backup
schedule.
ckbupsd(1M)
lshut, lpmov: start/stop the LP
scheduler and move requests.
lpmsched(1M)
schedctl: scheduler control call.
schedctl(2)
transport program. unsched: the
scheduler for the uucp file
unsched(1M)
setpriority: get/set program
setpriority: scheduling priority, getpriority,
getpriority(2)
npri: modify the
setpriority of a process.
npri(1)
xcalc: scientific calculator for x.
xcalc(1)
octet file.
schnhdr: section header for a MIPS
schnhdr(4)
image file.
scr_dump: format of curses screen
scr_dump(4)
blanktime: set the
screen blanking timeout.
blanktime(1G)
cedit: edit colors on the
screen.
cedit(6D)
clear: clear terminal
clear(1)
gclear: clear IRIS graphics
gclear(1G)
package. curses: terminal
curses handling and optimization.
curses(3X)
scr_dump: format of curses
screen image file.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mapimg:</td>
<td>translates a screen image into an RGB image.</td>
</tr>
<tr>
<td>icut:</td>
<td>save a part of the screen image into an image file.</td>
</tr>
<tr>
<td>scrsave:</td>
<td>save a part of the screen in an image file.</td>
</tr>
<tr>
<td>snapshot:</td>
<td>save a portion of the screen in an image file.</td>
</tr>
<tr>
<td>snoop:</td>
<td>magnify and report on the screen under the mouse pointer.</td>
</tr>
<tr>
<td>xmag:</td>
<td>magnify parts of the screen.</td>
</tr>
<tr>
<td>xrefresh:</td>
<td>refresh all or part of an X session.</td>
</tr>
<tr>
<td>editor based on/</td>
<td>vi, view, vedit:</td>
</tr>
<tr>
<td>network initialization and shut down</td>
<td>of a shell</td>
</tr>
<tr>
<td>tag:</td>
<td>tag a MIPS executable or shell script with an identifying number.</td>
</tr>
<tr>
<td>process interrupts in command</td>
<td>scripts. onintr:</td>
</tr>
<tr>
<td>trap:</td>
<td>process interrupts in command</td>
</tr>
<tr>
<td>in an image file.</td>
<td>scrsave: save a part of the screen</td>
</tr>
<tr>
<td>interface.</td>
<td>SCSI 1/4-inch Cartridge tape</td>
</tr>
<tr>
<td>rsclclosete:</td>
<td>communicate with generic SCSI devices.</td>
</tr>
<tr>
<td>Small Computer Systems Interface</td>
<td>(SCSI) disk driver. dks:</td>
</tr>
<tr>
<td>ds: generic (user mode)</td>
<td>SCSI driver.</td>
</tr>
<tr>
<td>smfd:</td>
<td>SCSI floppy disk driver.</td>
</tr>
<tr>
<td>program:</td>
<td>sdiff: side-by-side difference</td>
</tr>
<tr>
<td>string. fgrep:</td>
<td>search a file for a character</td>
</tr>
<tr>
<td>grep:</td>
<td>search a file for a pattern.</td>
</tr>
<tr>
<td>full regular expressions. egrep:</td>
<td>search a file for a pattern using</td>
</tr>
<tr>
<td>bsearch:</td>
<td>search a sorted table.</td>
</tr>
<tr>
<td>file(s). acctcom:</td>
<td>search and print process accounting</td>
</tr>
<tr>
<td>lssearch, lfind:</td>
<td>search and update.</td>
</tr>
<tr>
<td>hcreate, hdestroy:</td>
<td>manage hash files</td>
</tr>
<tr>
<td>tfind, tdelete, twalk:</td>
<td>manage binary files</td>
</tr>
<tr>
<td>Add_disk:</td>
<td>add a secondary disk to the system.</td>
</tr>
<tr>
<td>file. scnhrd:</td>
<td>section header for a MIPS object.</td>
</tr>
<tr>
<td>/ldnshread:</td>
<td>read an indexed/named section header of a common object/</td>
</tr>
<tr>
<td>/seek to line number entries of a</td>
<td>section of a common object file.</td>
</tr>
<tr>
<td>/ldnseek:</td>
<td>seek to a named indexed section of a common object file.</td>
</tr>
<tr>
<td>/access to per file descriptor</td>
<td>section of the symbol table.</td>
</tr>
<tr>
<td>size:</td>
<td>print the size of an object file.</td>
</tr>
<tr>
<td>newhost:</td>
<td>NeWS network</td>
</tr>
<tr>
<td>sed:</td>
<td>stream editor.</td>
</tr>
<tr>
<td>brk, sbk:</td>
<td>change data</td>
</tr>
<tr>
<td>seed48, lcong48:</td>
<td>generate uniformly/</td>
</tr>
<tr>
<td>rand48, mrand48, jrand48, srand48, a common object/</td>
<td>Ldseeks, Ldnseek: seek to an indexed/named section of</td>
</tr>
<tr>
<td>section of a/ Ldseeks, Ldnseek:</td>
<td>seek to line number entries of a</td>
</tr>
<tr>
<td>section of a/ Ldnseek, ldnseek:</td>
<td>seek to relocation entries of a</td>
</tr>
<tr>
<td>a common object file.</td>
<td>ldohseek: seek to the optional file header of</td>
</tr>
<tr>
<td>object file. ldohseek:</td>
<td>seek to the symbol table of a common</td>
</tr>
<tr>
<td>opendir, readdir, telldir, directory/ readdir, telldir, shmem:</td>
<td>get shared memory</td>
</tr>
<tr>
<td>brk, sbk:</td>
<td>change data</td>
</tr>
<tr>
<td>network:</td>
<td>specify for enabling</td>
</tr>
<tr>
<td>transferdevice:</td>
<td>of a shell</td>
</tr>
<tr>
<td>tag:</td>
<td>tag a MIPS executable or shell script with an identifying number.</td>
</tr>
<tr>
<td>process interrupts in command</td>
<td>scripts. onintr:</td>
</tr>
<tr>
<td>trap:</td>
<td>process interrupts in command</td>
</tr>
<tr>
<td>in an image file.</td>
<td>scrsave: save a part of the screen</td>
</tr>
<tr>
<td>interface.</td>
<td>SCSI 1/4-inch Cartridge tape</td>
</tr>
<tr>
<td>rsclclosete:</td>
<td>communicate with generic SCSI devices.</td>
</tr>
<tr>
<td>Small Computer Systems Interface</td>
<td>(SCSI) disk driver. dks:</td>
</tr>
<tr>
<td>ds: generic (user mode)</td>
<td>SCSI driver.</td>
</tr>
<tr>
<td>smfd:</td>
<td>SCSI floppy disk driver.</td>
</tr>
<tr>
<td>program:</td>
<td>sdiff: side-by-side difference</td>
</tr>
<tr>
<td>string. fgrep:</td>
<td>search a file for a character</td>
</tr>
<tr>
<td>grep:</td>
<td>search a file for a pattern.</td>
</tr>
<tr>
<td>full regular expressions. egrep:</td>
<td>search a file for a pattern using</td>
</tr>
<tr>
<td>bsearch:</td>
<td>search a sorted table.</td>
</tr>
<tr>
<td>file(s). acctcom:</td>
<td>search and print process accounting</td>
</tr>
<tr>
<td>lssearch, lfind:</td>
<td>search and update.</td>
</tr>
<tr>
<td>hcreate, hdestroy:</td>
<td>manage hash files</td>
</tr>
<tr>
<td>tfind, tdelete, twalk:</td>
<td>manage binary files</td>
</tr>
<tr>
<td>Add_disk:</td>
<td>add a secondary disk to the system.</td>
</tr>
<tr>
<td>file. scnhrd:</td>
<td>section header for a MIPS object.</td>
</tr>
<tr>
<td>/ldnshread:</td>
<td>read an indexed/named section header of a common object/</td>
</tr>
<tr>
<td>/seek to line number entries of a</td>
<td>section of a common object file.</td>
</tr>
<tr>
<td>/ldnseek:</td>
<td>seek to a named indexed section of a common object file.</td>
</tr>
<tr>
<td>/access to per file descriptor</td>
<td>section of the symbol table.</td>
</tr>
<tr>
<td>size:</td>
<td>print the size of an object file.</td>
</tr>
<tr>
<td>newhost:</td>
<td>NeWS network</td>
</tr>
<tr>
<td>sed:</td>
<td>stream editor.</td>
</tr>
<tr>
<td>brk, sbk:</td>
<td>change data</td>
</tr>
<tr>
<td>seed48, lcong48:</td>
<td>generate uniformly/</td>
</tr>
<tr>
<td>rand48, mrand48, jrand48, srand48, a common object/</td>
<td>Ldseeks, Ldnseek: seek to an indexed/named section of</td>
</tr>
<tr>
<td>section of a/ Ldseeks, Ldnseek:</td>
<td>seek to line number entries of a</td>
</tr>
<tr>
<td>section of a/ Ldnseek, ldnseek:</td>
<td>seek to relocation entries of a</td>
</tr>
<tr>
<td>a common object file.</td>
<td>ldohseek: seek to the optional file header of</td>
</tr>
<tr>
<td>object file. ldohseek:</td>
<td>seek to the symbol table of a common</td>
</tr>
<tr>
<td>opendir, readdir, telldir, directory/ readdir, telldir, shmem:</td>
<td>get shared memory</td>
</tr>
<tr>
<td>brk, sbk:</td>
<td>change data</td>
</tr>
</tbody>
</table>

April 1990

Version 5.0
program. visuallogin, noiconlogin: select and control console login ... visuallogin(4)
sorted files. comm: select or reject lines common to two ... comm(1)
multiplexing. select: synchronous I/O ... select(2)
file. cut: cut out ... cut(1)
odump: dump ... odump(1)
/providers a visual interface for selecting entries in the workspace/ ... transfermanager(1G)
point & click interface for selecting X11 font names. xfontsel: ... xfontsel(1)
interchange between cut buffer and selection. xcutsel: ... xcutsel(1)
case: selector in switch. ... csh(1)
uscspsema: attempts to acquire a semaphore, and fails if not/ ... uscspsema(3P)
configuration operations. usconfig: semaphore and lock arena ... usconfig(3P)
routine. usinit, _utrace: semaphore and lock initialization ... usinit(3P)
semct1: semaphore control operations. ... semct1(2)
uscsctlsema: semaphore control operations. ... uscsctlsema(3P)
out information about a specific semaphore DIRECTORY_BSD(3B). dump usdumpsema: ... semop(2)
semop: semaphore operations. ... semop(2)
ipcsm: remove a message queue, semaphore set or shared memory id. ... ipcsm(1)
usfresema: free a semaphore. ... usfresema(3P)
usinitsema: initializes a semaphore. ... usinitsema(3P)
allocates and initializes a semaphore. usnewsema: ... usnewsema(3P)
uspsema: attempt to acquire a semaphore. ... uspsema(3P)
ustestsema: return the value of a semaphore. ... ustestsema(3P)
usvsema: frees a resource to a semaphore. ... usvsema(3P)
semget: get set of semaphores. ... semget(2)
semct1: semaphore control ... semct1(2)
semget: get set of semaphores. ... semget(2)
semop: semaphore operations. ... semop(2)
send, sendto, sendmsg: send a message from a socket. ... send(2)
putmsg: send a message on a stream. ... putmsg(2)
group of processes. kill: send a signal to a process or a ... kill(2)
Mail: send and receive mail. ... mail_bsd(1)
network hosts. ping: send ICMP ECHO_REQUEST packets to ping(1M)
sendmail, newaliases, mail: send mail over the internet. ... sendmail(1M)
mail: send mail to users or read mail. ... mail_att(1)
message from a socket. send, sendto, sendmsg: send a ... send(2)
kill: send signal to a process (4.3BSD). ... kill(3B)
(4.3BSD). killpg: send signal to a process group ... killpg(3B)
DIFFTIME(3C). send signal to executing program ... raise:
printer. lp, cancel: send/cancel requests to an LP line ... lp(1)
aliases: aliases file for sendmail. ... aliases(4)
mail over the internet. sendmail, newaliases, mail: send ... sendmail(1M)
socket. send, sendto, sendmsg: send a message from a ... send(2)
a socket. send, sendto, sendmsg: send a message from ... send(2)
cdio: 6-port serial I/O. ... cdio(7)
slip: Serial Line IP. ... slip(1M)
duart: on-board serial ports. ... duart(7)
ethernet: IRIS-4D Series ethernet controllers. ... ethernet(7)
xhost: server access control program for X. ... xhost(1)
xstart: start up the sg1 X server as a NeWS client. ... xstart(1)
timed: time server daemon. ... timed(1M)
dgld: Distributed Graphics Library server. ... dgld(1M)
<table>
<thead>
<tr>
<th>Command/Service</th>
<th>Description</th>
<th>Permutation Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>shell’s environment to the NeWS</td>
<td>server. exportonews: Pass a login</td>
<td>exportonews(1)</td>
</tr>
<tr>
<td>fingerd: remote user information</td>
<td>server.</td>
<td>fingerd(1M)</td>
</tr>
<tr>
<td>xlsfonts:</td>
<td>server font list displayer for X.</td>
<td>xlsfonts(1)</td>
</tr>
<tr>
<td>Protocol. bootp:</td>
<td>server for Internet Bootstrap</td>
<td>bootp(1M)</td>
</tr>
<tr>
<td>Xsgi: SGI Iris</td>
<td>server for the X Window System.</td>
<td>xsgi(1)</td>
</tr>
<tr>
<td>Internet File Transfer Protocol</td>
<td>server. ftpd:</td>
<td>ftpd(1M)</td>
</tr>
<tr>
<td>named: Internet domain name</td>
<td>server.</td>
<td>named(1M)</td>
</tr>
<tr>
<td>xrdb: X</td>
<td>server resource database utility.</td>
<td>xrdb(1)</td>
</tr>
<tr>
<td>rexecd: remote execution</td>
<td>server.</td>
<td>rexecd(1M)</td>
</tr>
<tr>
<td>rlogind: remote login</td>
<td>server.</td>
<td>rlogind(1M)</td>
</tr>
<tr>
<td>rshd: remote shell</td>
<td>server.</td>
<td>rshd(1M)</td>
</tr>
<tr>
<td>rwho: system status</td>
<td>server.</td>
<td>rwho(1M)</td>
</tr>
<tr>
<td>talkd: remote user communication</td>
<td>server.</td>
<td>talkd(1M)</td>
</tr>
<tr>
<td>telnetd: Internet TELNET protocol</td>
<td>server.</td>
<td>telnetd(1M)</td>
</tr>
<tr>
<td>Trivial File Transfer Protocol</td>
<td>server. tftp: Internet</td>
<td>tftp(1M)</td>
</tr>
<tr>
<td>xlswins:</td>
<td>server window list displayer for X.</td>
<td>xlswins(1)</td>
</tr>
<tr>
<td>list interned atoms defined on</td>
<td>server. xlsatoms:</td>
<td>xlsatoms(1)</td>
</tr>
<tr>
<td>X: X Window System</td>
<td>server. xserver</td>
<td>xserver(1)</td>
</tr>
<tr>
<td>nsslookup: query name</td>
<td>servers interactively</td>
<td>nsslookup(1C)</td>
</tr>
<tr>
<td>order of host-address resolution</td>
<td>services. sethostsorder: specify</td>
<td>sethostsorder(3N)</td>
</tr>
<tr>
<td>setsid: create</td>
<td>session and set process group IDs.</td>
<td>setsid(2)</td>
</tr>
<tr>
<td>logout: end</td>
<td>session.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>sm: a</td>
<td>session manager for x.</td>
<td>sm(1)</td>
</tr>
<tr>
<td>script: make typescript of terminal</td>
<td>session.</td>
<td>script(1)</td>
</tr>
<tr>
<td>alarm: set a process alarm clock</td>
<td>set a process alarm clock.</td>
<td>alarm2</td>
</tr>
<tr>
<td>umask: set and get file creation mask</td>
<td>set and get file creation mask.</td>
<td>umask(2)</td>
</tr>
<tr>
<td>autologin: set autologin user identity.</td>
<td>set autologin user identity.</td>
<td>autologin(4)</td>
</tr>
<tr>
<td>of the standard supported character</td>
<td>set. charset: description</td>
<td>charset(5)</td>
</tr>
<tr>
<td>sigsetmask: set current signal mask</td>
<td>set current signal mask (4.3BSD).</td>
<td>sigsetmask(3B)</td>
</tr>
<tr>
<td>timezone: set default system time zone.</td>
<td>set default system time zone.</td>
<td>timezone(4)</td>
</tr>
<tr>
<td>execution. env: set environment for command</td>
<td>set environment for command.</td>
<td>env(1)</td>
</tr>
<tr>
<td>times. utime: set file access and modification</td>
<td>set file access and modification.</td>
<td>utime(2)</td>
</tr>
<tr>
<td>utimes: set file times.</td>
<td>set file times.</td>
<td>utimes(3B)</td>
</tr>
<tr>
<td>umask: set file-creation mode mask.</td>
<td>set file-creation mode mask.</td>
<td>umask(1)</td>
</tr>
<tr>
<td>version) INTGROUPS_BSD(3B).</td>
<td>set group access list (berkeley 4.3)</td>
<td>setgroups(2)</td>
</tr>
<tr>
<td>mousewarp, dialwarp, keywarp:</td>
<td>set input warping parameters.</td>
<td>mousewarp(6D)</td>
</tr>
<tr>
<td>/sgi_siganys, sgi_dumps: set signal</td>
<td>set manipulation and examination/</td>
<td>sigsetops(3)</td>
</tr>
<tr>
<td>nvram, sgikopt: get or</td>
<td>set non-volatile RAM variable(s).</td>
<td>nvram(1M)</td>
</tr>
<tr>
<td>apply: apply a command to a</td>
<td>set of arguments.</td>
<td>apply(1)</td>
</tr>
<tr>
<td>semget: get set of semaphores.</td>
<td>set of blocked signals (POSIX).</td>
<td>sigprocmask(2)</td>
</tr>
<tr>
<td>POSIX: sigpending: return set of signals pending for process</td>
<td>sigpending(2)</td>
<td></td>
</tr>
<tr>
<td>getsockopt, setsockopt: get and</td>
<td>set options on sockets.</td>
<td>getsockopt(2)</td>
</tr>
<tr>
<td>host system. hostid: set or print identifier of current</td>
<td>set or print identifier of current.</td>
<td>hostid(1)</td>
</tr>
<tr>
<td>system. hostname: set or print name of current host</td>
<td>set or print name of current host.</td>
<td>hostname(1)</td>
</tr>
<tr>
<td>remove a message queue, semaphore</td>
<td>set or shared memory id. ipcrm:</td>
<td>ipcrm(1)</td>
</tr>
<tr>
<td>setpgid: set process group ID.</td>
<td>set process group ID.</td>
<td>setpgid(2)</td>
</tr>
<tr>
<td>4.3BSD. setgrp, BSDsetgrp: set process group ID (System V and</td>
<td>setprocess(2)</td>
<td></td>
</tr>
</tbody>
</table>
setuid: create session and set process group IDs.         setuid(2)
setregid: set real and effective group ID.         setregid(2)
ssetreuid: set real and effective user ID's.         ssetreuid(2)
parameters: set: set shell flags or positional sh(1)
parameters.set: set: set shell flags or positional sh(1)
tabs: tabs: set tabs on a terminal.         tabs(1)
current window/ resize: utility to set TERMCAP and terminal settings to resize(1)
line discipline. getty: set terminal type, modes, speed, and getty(1M)
line discipline. ugetty: set terminal type, modes, speed, and ugetty(1M)
window. textcolors: set the colors used by a text textcolors(1G)
date: print and set: set the date.         date(1)
format. setmon: set the default monitor video output setmon(1)
/.gamma. gamma: get or set the gamma value stored in gamma(6D)
hyroute: set the HyperNet routing tables.         hyroute(1M)
sty: set the options for a terminal.         sty(1)
blanktime: set the screen blanking timeout.         blanktime(1G)
iset: set the type of an image.         iset(6D)
stime: set time.         stime(2)
debugging. setsysm: set up a debug kernel for symbolic setsysm(1)
seteuid, setreuid, setegid, setregid: set user and group IDs.         seteuid(3C)
setuid, setgid: set user and group IDs.         setuid(2)
ulimit: get and set user limits.         ulimit(2)
ssetenv: set variable in environment.         ssetenv(1)
blockproc, unblockproc, /blockprocall, /blockprocall,/blockprocall: routines to/ blockproc(2)
ssetlinebuf: assign buffering to a/ buffering to a/ setbuf, setvbuf, setbuffer, setbuffer, setlinebuf: assign setbuf(3S)
setbuffer, setlinebuf: assign setbuf(3S)
/setascii, _tolower, _toupper, setchrclass: character handling.         ctype(3C)
current domain. getdomainname: get/set name of getdomainname(2)
IDs. seteuid, setreuid, setegid, setregid: set user and group setegid, setregid: set user and group seteuid(3C)
ssetenv: set variable in environment.         ssetenv(1)
sseteuid, ssetid, ssetregid: sseteuid(3C)
get_fpc_eir:/fpc, get_fpc_esr, get_fpc_ecr, get_fpc_eir, get_fpc_eir, set_fpcLed, swapRM, swapINX:/fpc(3C)
/setpcLed, swapRM, swapINX:/fpc(3C)
/setuid: get/set unique identifier gethostid(2)
host. gethostname: get/set name of current gethostname(2)
host-address resolution services. gethostresorder: specify order of gethostresorder(3N)
get hardware inventory / getinvent, endinvent, scaninvent: getgetinvent(3)
timer. gettimer, /longjmp, sigsetjmp, siglongjmp, setitimer: get/set value of interval setitimer(2)
 siglongjmp, _longjmp: non-local gotos. setjmp(3C)
 siglongjmp, _longjmp: _longjmp: set jmp, long jmp, sigset jmp, set jmp(3C)
 encryption. crypt, setkey, encrypt: generate hashing crypt(3C)
stream. setbuf, setvbuf, setbuffer, setlinebuf: assign buffering to a setbuffer(3S)
log. syslog, openlog, closelog, setlogmask, vsyslog: control system syslog(3B)
video output format. setmon: set the default monitor setmon(1)

Version 5.0 –88– April 1990
entry. /netbyaddr, getnetbyname, 
for the NEWSSERVER environment/grepnetbyaddr, 
eterror,
serror,
group ID (System V and 4.3BSD). 
grepnetbyaddr, 
scheduler priority, grepblsfr, setpriority, 
getprotobynumber, getprotobyname, 
getprotobyname, 
getpwent, getpwnam, getpwnam, 
group ID. 
grepnetbyaddr, 
setuid, setreuid, setegid, 
resource consumption. getrlimit, 
and group IDs. setegid, 
generates Mandelbrot and Julia 
grepnetbyaddr, 
genera
gesets. getsockopt. 
generator/ 
symmetric debugging. 
gmtimeofday, 
time. profile. 
rxsetroot: root window parameter 
utility to set TERM window and terminal 
gretypedefs: speed and terminal 
group ID's. setegid, 
group ID's. 
/usr. 
install buffering to a/ setbuf, 
simulate the flight of any of 
one/ paste: merge same lines of 
swap the/ gethostsex: get the byte 
the host machine swap_*() - swap the 
machine-independent fashion. spult, 
sgetic: access long integer data in a 
sgigsc: SGI graphics system call. 
texturebind: SGI graphics system call. 
System. Xsgi: 
SGI Iris server for the X Window 
xssi: xstart up the 
/sgi_xserver, sigismember, 
and/ /sgi_xserver, sgi_siganys, 
variable(s). nvram, 
strings. 
yield function. 
set/ sgi_xserver, sgi_xserver, sgi_siganys, 
stderr/ sgi_xserver, sgi_siganys, 
standard/restricted command/ 
view of the dogfight. 
demonstrates real-time lighting and 
sproc: create a new 
mkshlib: create a 
setent, endtendent: get network 
setnetent, snewhost, snet: generate a string 
serror: set/get system error. 
setpgrp: set process group ID. 
serror(3C) 
setpgrp(2) 
grothpriority: get/set system program 
setpriority(2) 
grothpriority: get/set system program 
setpgrp: set process group ID. 
setpgrp(2) 
setgetpwnam, getpwnam, getpwnam: 
grothpriority: get/set system program 
setpriority(2) 
grothpriority: get/set system program 
setpriority(2) 
grothpriority: get/set system program 
setpriority(2) 
setgroup: set real and effective 
setreuid(2) 
setgroup: set real and effective 
setreuid(2) 
setgroup: set user and group IDs. 
setegid(3C) 
setgroup: set control maximum system 
setrlimit(2) 
setgroup: set real and effective 
setreuid(3C) 
setgetpwnam, getpwnam, getpwnam: 
grothpriority: get/set system program 
setpriority(2) 
setgetpwnam, getpwnam, getpwnam: 
grothpriority: get/set system program 
setpriority(2) 
setgetpwnam, getpwnam, getpwnam: 
grothpriority: get/set system program 
setpriority(2) 
setgetpwnam, getpwnam, getpwnam: 
grothpriority: get/set system program 
setpriority(2) 
setgetpwnam, getpwnam, getpwnam: 
grothpriority: get/set system program 
setpriority(2) 
setgetpwnam, getpwnam, getpwnam: 
grothpriority: get/set system program 
setpriority(2) 
sget: access long integer data in a 
sgetic(3C) 
texturebind: SGI graphics system call. 
sgetic(3C) 
texturebind: SGI graphics system call. 
sgetic(3C) 
SGI Iris server for the X Window 
xsplash(1) 
SGI- /signal set manipulation and 
SGI/ sxi_xserver, sgi_siganys, 
sget: get or set non-volatile RAM 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgi_xserver, sgi_siganys, 
sget: get or set non-volatile RAM 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
sgetic(2) 
shadow: full-screen armchair pilot's 
shadows: light: 
share group process. 
shared library.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>usmallopt, ussmallinfo: user</td>
<td>shared memory allocator. /usmalloc, shared memory control operations.</td>
<td>usmalloc(3P)</td>
</tr>
<tr>
<td>shmod:</td>
<td>shared memory control operations.</td>
<td>shmod(2)</td>
</tr>
<tr>
<td>a message queue, semaphore set or shmat, shmdt:</td>
<td>shared memory id. ipcrm: remove shared memory operations.</td>
<td>ipcrm(1)</td>
</tr>
<tr>
<td>shmdt:</td>
<td>shared memory segment identifier.</td>
<td>shmdt(2)</td>
</tr>
<tr>
<td>strings from C programs to implement sc: spread</td>
<td>shared strings. xstr: extract sheet calculator.</td>
<td>xstr(1)</td>
</tr>
<tr>
<td>C-like syntax. csh: a system: issue a exit: leave</td>
<td>shell (command interpreter) with shell command.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>eval: re-evaluate</td>
<td>shell data.</td>
<td>system(3S)</td>
</tr>
<tr>
<td>eval: re-evaluate</td>
<td>shell data.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>popd: pop</td>
<td>shell directory stack.</td>
<td>sh(1)</td>
</tr>
<tr>
<td>pushd: push</td>
<td>shell directory stack.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>parameters. set: set</td>
<td>shell flags or positional</td>
<td>sh(1)</td>
</tr>
<tr>
<td>recording and playback from an IRIX alias:</td>
<td>shell journal: 4Sight event shell. /journal: 4Sight event</td>
<td>journalplay(1)</td>
</tr>
<tr>
<td>/shutacct, startup, turnacct:</td>
<td>shell procedures for accounting.</td>
<td>acctsh(1M)</td>
</tr>
<tr>
<td>psh: NeWS PostScript</td>
<td>shell.</td>
<td>psh(1)</td>
</tr>
<tr>
<td>rsh: remote</td>
<td>shell.</td>
<td>rsh BSD(1C)</td>
</tr>
<tr>
<td>extending the/ transferdevice: a tag: tag a MIPS executable or rshd: remote</td>
<td>shell script specification for shell script with an identifying/</td>
<td>transferdevice(4)</td>
</tr>
<tr>
<td>exit: leave</td>
<td>shell.</td>
<td>rshd(1M)</td>
</tr>
<tr>
<td>command programming/ sh, rsh:</td>
<td>shell server.</td>
<td>sh(1)</td>
</tr>
<tr>
<td>set: change value of</td>
<td>shell variable.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>@: arithmetic on</td>
<td>shell variables.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>unset: discard</td>
<td>shell variables.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>readonly: make</td>
<td>shell variables read-only.</td>
<td>sh(1)</td>
</tr>
<tr>
<td>export: add</td>
<td>shell variables to the environment.</td>
<td>sh(1)</td>
</tr>
<tr>
<td>groups. multgrps: spawn a exec: overlay</td>
<td>shell with membership in multiple shell with specified command.</td>
<td>multgrps(1)</td>
</tr>
<tr>
<td>exec: overlay</td>
<td>shell with specified command.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>wsh: creates and specifies a window server. exporttonews: Pass a login shell’s environment to the NeWS</td>
<td>exporttonews(1)</td>
<td></td>
</tr>
<tr>
<td>operations.</td>
<td>shift: manipulate argument list.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>operations.</td>
<td>shift: manipulate argument list.</td>
<td>sh(1)</td>
</tr>
<tr>
<td>shmdt, shmat:</td>
<td>shared memory operations.</td>
<td>shmdt(2)</td>
</tr>
<tr>
<td>shmdctl: shared memory control</td>
<td></td>
<td>shmdctl(2)</td>
</tr>
<tr>
<td>shmat, shmdt:</td>
<td>get shared memory segment identifier.</td>
<td>shmdt(2)</td>
</tr>
<tr>
<td>xeyes: watch over your shoulder.</td>
<td></td>
<td>xeyes(1)</td>
</tr>
<tr>
<td>xshowmap:</td>
<td>show colormap.</td>
<td>xshowmap(1)</td>
</tr>
<tr>
<td>groups: show group memberships.</td>
<td></td>
<td>groups(1)</td>
</tr>
<tr>
<td>ruetime: show host status of local machines.</td>
<td></td>
<td>ruetime(1C)</td>
</tr>
<tr>
<td>uptime: show how long system has been up.</td>
<td></td>
<td>uptime(1)</td>
</tr>
<tr>
<td>netstat: show network status.</td>
<td></td>
<td>netstat(1)</td>
</tr>
<tr>
<td>color map. showmap: display the contents of the file. showman: print contents of an SNF</td>
<td>showman(6D)</td>
<td></td>
</tr>
<tr>
<td>izoom: magnify or shrink an image.</td>
<td></td>
<td>izoom(6D)</td>
</tr>
<tr>
<td>connection. shutdown: shut down part of a full-duplex</td>
<td></td>
<td>shutdown(2)</td>
</tr>
</tbody>
</table>

Version 5.0

April 1990
state. shutdown: shut down system, change system shutdown(1M)
/prctmp, prtdaily, prtacct, runacct, shutdown script. network(1M)
network: network initialization and shutdown: shut down part of a shutdown(2)
full-duplex connection. shutdown: shut down system, change shutdown(1M)
system state. systemdown: interactive script for shutting the system down...
systemdown(1G)

sdiff: side-by-side difference program. sdiff(1)
facilities (POSIX).
sigaction: software signal sigaction(2)
sigaction: software signal sigaction(2)
sigfilsset, sigismember, sigfilsset, sigset(3)
sigaddset, sigdelset, sigemptyset, sigset(3)
delivery to process (4.3BSD). sigdelset, sigemptyset, sigset(3)
sigismember, sigaddset, sigdelset, sigemptyset, sigset(3)
sigfilsset, sigismember, sigaddset, sigdelset, sigemptyset, sigset(3)
sigpause: signal management/ sigset sigpause: signal management/ sigset sigset(2)
sigset, sighold, sigrelse, sigset(), sigset(2)
/sigdelset, sigemptyset, sigfilsset, sigset(), sigset(3)
sigismember, sigset(), sigset(3)
setjmp, longjmp, sigsetjmp, sigsetjmp, setjmp(3C)
login: signal. signal on. login(1)
signal: simplified software signal facilities (4.3BSD). signal(3B)
4.3bsd software signal facilities GETDTABLES(3). sigvec:
sigaction: software signal facilities (POSIX). sigaction(2)
signal: software signal facilities (System V). signal(2)
/sigrelse, sigignore, sigpause: signal management (System V). sigset(2)
sigsetmask: set current signal mask (4.3BSD). sigsetmask(3B)
psignal, sys_siglist: system signal messages. psignal(3C)
pause: suspend process until signal. signal. pause(2)
/siganyset, sgi_dumpset: signal mask manipulation and/ signal mask manipulation and/ sigset(3)
facilities (4.3BSD). signal: simplified software signal signal(3B)
(System V). signal: software signal facilities signal(2)
kill: send signal to a process (4.3BSD). kill(3B)
killpg: send signal to a process group (4.3BSD). killpg(3B)
processes. kill: send a signal to a process or a group of kill(2)
DIFFTIME(3C). send signal to executing program raise:
sigpause: atomically release blocked signals and wait for interrupt/ sigpause(3B)
(POSIX). /atomically release blocked signals and wait for interrupt/ sigsuspend(3B)
(4.3BSD). sigblock: block signals from delivery to process sigblock(3B)
sigpending: return set of signals pending for process (POSIX) sigpending(2)
previous state of the set of blocked signals (POSIX) /alter and return sigprocmask(2)
ssignal, sgisignal: software signals. ssignal(3C)
signals and wait for interrupt/ sigpause: atomically release blocked sigpause(3B)
System V). sigpause: signal management (System sigset(2)
signal(2)
/sighold, sigrelse, sigignore, sigpause: signal management/ sigset sigset(2)
signal(2)
/signal and wait for interrupt/ sigprocmask: alter and return sigprocmask(2)
V). sigrelse, sigignore, sigpause: signal management/ sigset sigset(2)
pending for process (POSIX). sigrelse, sigignore, sigpause: sigset(2)
previous state of the set of signal management/ sigset, sighold, sigset, sighold, sigrelse, sigignore, sigpause: signal/ sigset(2)
V). sigrelse, sigignore, sigpause: signal/ sigset(2)
/signal and wait for interrupt/ sigset, longjmp, sigsetjmp, longjmp, sigsetjmp, siglongjmp, _setjmp, longjmp/
(4.3BSD). sigsetmask: set current signal mask sigsetmask(3B)
broadcast signals and wait for/ sigsuspend: atomically release sigsuspend(2)
additions) SIGVEC(3B). specific
card. t3270: Silicon Graphics 3270 interface t3270(7)
interface card. gse: Silicon Graphics 5080 workstation gse(7)

April 1990

Version 5.0
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Permutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>distcp: copy</td>
<td>software distribution</td>
<td>distcp(1M)</td>
</tr>
<tr>
<td>inst:</td>
<td>software installation tool</td>
<td>inst(1M)</td>
</tr>
<tr>
<td>signal:</td>
<td>simplified software signal facilities (4.3BSD)</td>
<td>signal(3B)</td>
</tr>
<tr>
<td>GETDTABLESZE(3)</td>
<td>4.3bsd software signal facilities</td>
<td>sigvec:</td>
</tr>
<tr>
<td>sigaction:</td>
<td>software signal facilities (POSIX)</td>
<td>sigaction(2)</td>
</tr>
<tr>
<td>V): signal:</td>
<td>software signal facilities (System)</td>
<td>signal(2)</td>
</tr>
<tr>
<td>ssignal, gsignal:</td>
<td>software signals</td>
<td>ssignal(3C)</td>
</tr>
<tr>
<td>versions:</td>
<td>software versions tool</td>
<td>versions(1M)</td>
</tr>
<tr>
<td>finite element analysis program.</td>
<td>solidview: display the results of a solidview(6D)</td>
<td></td>
</tr>
<tr>
<td>dumpfont:</td>
<td>dump font out in some other format.</td>
<td>dumpfont(1)</td>
</tr>
<tr>
<td>window. plaid: paint</td>
<td>some plaid-like patterns in an X plaid(1)</td>
<td></td>
</tr>
<tr>
<td>sort:</td>
<td>sort and/or merge files</td>
<td>sort(1)</td>
</tr>
<tr>
<td>qsort: quicker</td>
<td>sort</td>
<td>qsort(3C)</td>
</tr>
<tr>
<td>tsort: topological</td>
<td>sort</td>
<td>tsort(1)</td>
</tr>
<tr>
<td>select or reject lines common to two</td>
<td>sorted files. comm: comm(1)</td>
<td></td>
</tr>
<tr>
<td>bsearch:</td>
<td>binary search a sorted table.</td>
<td>bsearch(3C)</td>
</tr>
<tr>
<td>program. whereas: locate</td>
<td>source, binary, and or manual for whereas(1)</td>
<td></td>
</tr>
<tr>
<td>an error message file by massaging C</td>
<td>source. mkstr: create mkstr(1)</td>
<td></td>
</tr>
<tr>
<td>zero:</td>
<td>source of zeroes.</td>
<td>zero(7)</td>
</tr>
<tr>
<td>dbx: a</td>
<td>source-level debugger</td>
<td>dbx(1)</td>
</tr>
<tr>
<td>brk, sbk:</td>
<td>change data segment space allocation.</td>
<td>brk(2)</td>
</tr>
<tr>
<td>multiple groups. mulgrps:</td>
<td>spawn a shell with membership in</td>
<td>mulgrps(1)</td>
</tr>
<tr>
<td>ct:</td>
<td>spawn getty to a remote terminal.</td>
<td>ct(1C)</td>
</tr>
<tr>
<td>memory efficient way. vfork:</td>
<td>spawn new process in a virtual</td>
<td>vfork(2)</td>
</tr>
<tr>
<td>mkfifo:</td>
<td>make a FIFO special file</td>
<td>mkfifo(2)</td>
</tr>
<tr>
<td>mknod: build</td>
<td>special file or named pipe (FIFO).</td>
<td>mknod(1M)</td>
</tr>
<tr>
<td>dn_ll, dn_netman: 4DDN</td>
<td>special files.</td>
<td>dn_ll(7)</td>
</tr>
<tr>
<td>intro:</td>
<td>introduction to special files.</td>
<td>intro(7)</td>
</tr>
<tr>
<td>MAKEDEV:</td>
<td>Create device special files.</td>
<td>madev(1M)</td>
</tr>
<tr>
<td>mknod: make a directory, or a</td>
<td>special or ordinary file.</td>
<td>mknod(2)</td>
</tr>
<tr>
<td>dump out information about a</td>
<td>specific lock USDUMPSEMA(3P).</td>
<td>usdumpplock:</td>
</tr>
<tr>
<td>dump out information about a</td>
<td>specific semaphore/ usdumpsema:</td>
<td></td>
</tr>
<tr>
<td>cftime:</td>
<td>language specific strings.</td>
<td>cftime(4)</td>
</tr>
<tr>
<td>transferdevice: a shell script</td>
<td>specification for extending the/</td>
<td>transferdevice(4)</td>
</tr>
<tr>
<td>fspec: format</td>
<td>specification in text files.</td>
<td>fspec(4)</td>
</tr>
<tr>
<td>keyboard: keyboard</td>
<td>specifications.</td>
<td>keyboard(7)</td>
</tr>
<tr>
<td>mouse: optical mouse</td>
<td>specifications.</td>
<td>mouse(7)</td>
</tr>
<tr>
<td>exec: overlay shell with</td>
<td>specified command.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>exec: overlay shell with</td>
<td>specified command.</td>
<td>sh(1)</td>
</tr>
<tr>
<td>Backup: backup the</td>
<td>specified file or directory.</td>
<td>backup(1)</td>
</tr>
<tr>
<td>tape. Restore: restore the</td>
<td>specified file or directory from</td>
<td>restore(1)</td>
</tr>
<tr>
<td>truncate: truncate a file to a</td>
<td>specified length. truncate, truncate(2)</td>
<td></td>
</tr>
<tr>
<td>swap_*() - swap the sex of the</td>
<td>specified structure. /host machine sex(3X)</td>
<td></td>
</tr>
<tr>
<td>wsh: creates and</td>
<td>specifies a window shell.</td>
<td>wsh(1G)</td>
</tr>
<tr>
<td>resolution/ sethostresorder:</td>
<td>specify order of host-address</td>
<td>sethostresorder(3N)</td>
</tr>
<tr>
<td>getty: set terminal type, modes,</td>
<td>speed, and line discipline.</td>
<td>getty(1M)</td>
</tr>
<tr>
<td>uugetty: set terminal type, modes,</td>
<td>speed, and line discipline.</td>
<td>uugetty(1M)</td>
</tr>
<tr>
<td>getty. gettydefs:</td>
<td>speed and terminal settings used by</td>
<td>gettydefs(4)</td>
</tr>
<tr>
<td>(struct termios *termios_p, speed_t</td>
<td>speed); GETLVENT(3C). cfsetispeed int</td>
<td>int</td>
</tr>
</tbody>
</table>
Permuter Index

\(\text{[struct termios} *\text{termios}_p, \text{speed}_t \text{ speed})\) speed; \text{speed}_t \text{ cfgetispeed} (\text{struct/}) \quad \text{int}

\(\text{[struct termios} *\text{termios}_p, \text{speed}_t \text{ speed})\) \text{speed}_t \text{ cfgetispeed} (\text{struct termios/}) \quad \text{int}

\(\text{posix baud rate primitives} \#\text{include} \quad \text{speed}_t \text{ cfgetospeed} (\text{struct termios/}) \quad \text{cfgetospeed,}

\(\text{[struct termios} *\text{termios}_p, \text{speed}_t \text{ speed})\) \text{speed}_t \text{ cfgetispeed} \quad \text{int}

\(\text{spelling errors.} \quad \text{spell, spellin, spellout: find} \quad \text{spell(1)}

\(\text{spelling errors.} \quad \text{spell, spellin, spellout: find} \quad \text{spell(1)}

\(\text{spelling errors.} \quad \text{spell, spellin, spellout: find} \quad \text{spell(1)}

\(\text{spelling errors.} \quad \text{spell, spellin, spellout: find} \quad \text{spell(1)}

\(\text{flip: spin one or more objects.} \quad \text{flip(6D)}

\(\text{uswsetlock, usetlock, usunselock:} \quad \text{spinlock routines.} \quad \text{/uswsetlock,} \quad \text{uselock(3P)}

\(\text{hl: hardware} \quad \text{spinlocks driver.} \quad \text{hl(7M)}

\(\text{split: split a file into pieces.} \quad \text{split(1)}

\(\text{csplit: context} \quad \text{split.} \quad \text{csplit(1)}

\(\text{split: split a file into pieces.} \quad \text{split(1)}

\(\text{uucleanup: uucp} \quad \text{spool directory clean-up.} \quad \text{uucleanup(1M)}

\(\text{lpq: spool queue examination program.} \quad \text{lpq(1)}

\(\text{remove jobs from the line printer} \quad \text{spooling queue.} \quad \text{lprm:} \quad \text{lprm(1)}

\(\text{lpadmin: configure the LP} \quad \text{spooling system.} \quad \text{lpadmin(1M)}

\(\text{remove a printer from the LP} \quad \text{spooling system.} \quad \text{mprinter:} \quad \text{mprinter(1M)}

\(\text{arena: a future} \quad \text{sport.} \quad \text{arena(6D)}

\(\text{sc:} \quad \text{spread sheet calculator.} \quad \text{sc(1)}

\(\text{printf, fprintf,} \quad \text{printf: print formatted output.} \quad \text{printf(3S)}

\(\text{process.} \quad \text{sproc: create a new share group} \quad \text{sproc(2)}

\(\text{data in a machine-independent/} \quad \text{sputc, sgctl: access long integer} \quad \text{sputc(3X)}

\(\text{root.} \quad \text{sqrr, fsqrt, cbrt: cube root, square} \quad \text{sqrt(3M)}

\(\text{generator. rand,} \quad \text{rand: simple random-number} \quad \text{rand(3C)}

\(\text{/rand48, rand48, mrand48, jrand48,}/\text{random number generator; routines/} \quad \text{rand48, seed48, lcong48: generate/} \quad \text{drand48(3C)}

\(\text{random number generator; routines/} \quad \text{random, initstate, setstate: better} \quad \text{random,}

\(\text{scanf, fscanf,} \quad \text{sscanf: convert formatted input.} \quad \text{sscanf(3S)}

\(\text{ssignal, ssignal: software signals.} \quad \text{ssignal(3C)}

\(\text{popd: pop shell directory} \quad \text{stack.} \quad \text{csh(1)}

\(\text{pushd: push shell directory} \quad \text{stack.} \quad \text{csh(1)}

\(\text{package. stdio:} \quad \text{standard buffered input/output} \quad \text{stdio(3S)}

\(\text{read: accept input from the} \quad \text{standard colormap utility.} \quad \text{xormap(1)}

\(\text{read: accept input from the} \quad \text{standard input.} \quad \text{sh(1)}

\(\text{package. ftk:} \quad \text{standard interprocess communication} \quad \text{stdipc(3C)}

\(\text{charset: description of the} \quad \text{standard supported character set.} \quad \text{charset(5)}

\(\text{ansitape: ANSI} \quad \text{standard tape handler.} \quad \text{ansitape(1)}

\(\text{programming/} \quad \text{sh, rsh: shell, the} \quad \text{standard/restricted command} \quad \text{sh(1)}

\(\text{X} \quad \text{Standards.} \quad \text{xstandards(1)}

\(\text{client. xstart:} \quad \text{start up the sgi X server as a NeWS} \quad \text{xstart(1)}

\(\text{requests. lpsched, lpshut, lpmov/} \quad \text{start/stop the LP scheduler and move} \quad \text{lpsched(1M)}

\(\text{prdaily, prtcacct, runacct, shuctacct,} \quad \text{startup, turnacct: shell procedures/} \quad \text{acctsh(1M)}

\(\text{call.} \quad \text{stat: data returned by stat system call.} \quad \text{stat(5)}

\(\text{state by deleting printers.} \quad \text{state of the set of blocked signals} \quad \text{sigprocmask(2)}

Version 5.0

-94-

April 1990
tcgetattr: posix get/set terminal
state primitives #include int/ state primitives #include int/
tcgetattr, shutdown
shut down system, change system state. shutdown: shutdown(1M)
fsync: synchronize a file’s in-core state with that on disk fsync(2)
if: conditional if, then: conditional statement. sh(1)
information. statefs, statfs: get file system statfs(2)
file systems. fstab: static information about statfs(2)
fstab(4)/print_unaligned_summary: gather statistics on unaligned references.
ustat: get file system statistics. ustat(2)
fsstat: report file system status. fsstat(1M)
lpstat: print LP status information. lpstat(1)
ifeof, clearerr, fileno: stream status inquiries. err, ferr, ferr(3S)
ustat: uucp status inquiry and job control. ustat(1C)
communication facilities status. ipcs: report inter-process ipcs(1)
control and report processor status. mpadmin: mpadmin(1)
netstat: show network status. netstat(1)
check: check RCS status of a file. check(1)
runtime: show host status of local machines. runtime(1C)
ps: report process status. ps(1)
rwhoed: system status server. rwhoed(1M)
stat, liost, fstat: get file status. stat(2)
interfaces to auxiliaries. staux: routines that provide scalar staux(3X)
compilation unit symbol table/ stcu: routines that provide a stcu(3X)
input/output package/ stdarg: variable argument list. stdarg(5)
intermediate-code symbolic/ stdio: standard buffered stdio(3S)
to per file descriptor section of stdump: dump a file of stdump(1)
high-level interface to basic/ stfd: routines that provide access stfd(3X)
read/write interface to the MIPS/ stfe: routines that provide a stfe(3X)
system. powerdown: stop all processes and halt the powerdown(1M)
wait3: wait for child processes to stop or terminate. wait, waitpid, wait(2)
rc0: run commands performed to stop the operating system. rc0(1M)
synchronize memory with physical storage. msysnc msysnc(2)
data base/ dbminit, fetch, store, delete, firstkey, nextkey: dbm(3B)
gamma: get or set the gamma value stored in. /gamma. gamma(6D)
rl: force an image to be stored using run length encoding. rle(6D)
verbatim: force an image to be stored without run length encoding. verbatim(6D)
winicons: stowed window image mechanism. winicons(5W)
symbol table. sprite: routines to print the sprite(3X)
/strdup, strmcat, strcmp, strcasecmp, strncasecmp, string(3C)
strmcmp, strcasecmp, strncasecmp, strcspn, / strcat, strdup, strcat, strcmp, string(3C)
strchr, strchrm, strpbrk, strspn, string(3C)
strcat, strdup, strcat, strcmp, string(3C)
/strcmp, strcasecmp, strncasecmp, string(3C)
/strchr, strchrm, strpbrk, strspn, string(3C)
strcspn, strtok, strir, index, string(3C)
strcat, strdup, strcat, strcmp, string(3C)
sed: stream editor. sed(1)
fclose, fflush: close or flush a stream. fclose(3S)
fopen, fread, fopen: open a stream. fopen(3S)
reposition a file pointer in a stream. fseek, rewind, ftell. fseek(3S)
Permutted Index

getw: get character or word from a stream. getc, getchar, fgetc, ... getc(3S)
getmsg: get next message off a stream. ... getmsg(2)
gets, fgets: get a string from a stream. ... gets(3S)
putw: put character or word on a stream. putc, putwchar, fputc, ... putc(3S)
putmsg: send a message on a stream. ... putmsg(2)
puts, puts: put a string on a stream. ... puts(3S)
setlinebuf: assign buffering to a stream. setbuf, setvbuf, setbuffer, ... setbuf(3S)
ferror, feof, clearerr, fileno: stream status inquiries. ... feof(3S)
/ruserok: routines for returning a stream to a remote command. ... ruserok(3N)
rexec: return stream to a remote command. ... rexec(3N)

clone: open any minor device on a STREAMS ioctl commands. ... clone(7)
sys_errno: system error/ perror, strerror, errno, sys_errlist, ... perror(3C)
/gmtime, asctime, ctime, asctime, localtime ... convert between ... localtime(3C)
convert floating-point number to string. ... cvt(3C)
fgrep: search a file for a character string. ... fgrep(1)
setnewhost, ssn: generate a string for the NEWSSERVER/ setnewhost(1)
gets, fgets: get a string from a stream. ... getss(3S)
puts, fputs: put a string on a stream. ... puts(3S)

bcopy, bcmp, bllkclr, bzero: byte string operations. ... bstr(3C)
strtok, strptr, index, rindex: string operations. ... /strspn, strcspn, ... /strspn(3C)
tzset: convert date and time to string. ... /time(3C)
strotof, atof: convert string to double-precision number. ... strotof(3C)
strtol, atol, atoi: convert string to integer. ... strtol(3C)
cftime: language specific strings. ... /ctime(4)
in an object, or other binary file. strings: find the printable strings ... /strings(1)
shared strings. xstr: extract strings from C programs to implement xstr(1)
binary/ strings: find the printable strings in an object, or other strings(1)
sgikopt: retrieve kernel option sgikopt(2)

from C programs to implement shared

unifdef: strip or reduce ifdefs in C code. ... unifdef(1)

bits.

/strmcasecmp, strcmp, strncasecmp, strlen, strchr, strcchr, strpbrk, ... strlen(3C)
/strncpy, strncmp, strcasecmp, ... stncmp, strmpy, ... stncmp(3C)
strcasecmp/ strcat, strndup, ... stncat, strncmp, ... stncat(3C)
strcat, strndup, ... stncat, strcasecmp, ... stncasecmp, strcmp, stncmp, ... stncasecmp(3C)
/stmcasecmp, strcmp, strcasecmp, ... stmpy, strmpy, ... stmpy(3C)
/strncpy, strlcn, strchr, strcchr, ... strlcn, strchr, ... strlcn(3C)
/strncpy, strlcn, strchr, strcchr, ... strlcn, strchr, ... strlcn(3C)
/strncpy, strlcn, strchr, strcchr, ... strlcn, strchr, ... strlcn(3C)
/strncpy, strlcn, strchr, strcchr, ... strlcn, strchr, ... strlcn(3C)
double-precision number. ... strotof, atof: convert string to ... strotof(3C)
/strotof, strpbrk, strspn, strcspn, ... strotof, atof: convert string to ... strotof(3C)
/strotof, strpbrk, strspn, strcspn, ... strotof, atof: convert string to ... strotof(3C)

#include speed_t cfgetospeed
/speed_t speed); /speed_t cfgetispeed

speed); GETLVENTS(3C), cfsetispeed

(speed); speed_t/ cfsetispeed

(struct termios *termios_p); ... cfgetispeed,
(struct termios *termios_p); ... cfgetispeed,
(struct termios *termios_p, speed_t) ... cfgetispeed,
(struct termios *termios_p, speed_t) ... cfgetispeed,
(struct termios *termios_p, speed_t) ... cfgetispeed,
(struct termios *termios_p, speed_t) ... cfgetispeed,
Silicon Graphics


April 1990

-97- Version 5.0
needed to access and add to the symbol table. /to basics functions ... stfe(3X)
read/write interface to the MIPS symbol table. /that provide a binary ... stio(3X)
script: routines to print the symbol table. ... script(3X)
unistd: file header for symbolic constants. ... unistd(4)
setsys: set up a debug kernel for symbolic debugging. ... setsym(1)
dump a file of intermediate-code symbolic information. stdump: ... stdump(1)
readlink: read value of a symbolic link. ... readlink(2)
symlink: make symbolic link to a file. ... symlink(2)
tlink: clone a file tree using symbolic links. ... tlink(1)
strip: remove symbols and relocation bits. ... strip(1)
/gp: loader defined symbols in a program. ... end(3C)
file. symlink: make symbolic link to a ... symlink(2)
sync: update super block. ... sync(2)
sync: update the super block. ... sync(1M)
adjtime: correct the time to allow synchronization of the system clock. ... adjtime(2)
with that on disk. fsync: synchronize a file’s in-core state ... fsync(2)
storage. msync: synchronize memory with physical ... msync(2)
select: synchronous I/O multiplexing. ... select(2)
(command interpreter) with C-like administration. sysadm: menu interface to do system ... sysadm(1)
variables (POSIX). sysconf: get configurable system ... sysconf(2)
messages. perror, sterror, errno, sys_errnolist, sys_nerr: system error ... perror(3C)
information. sysex: get file system type ... sysexfs(2)
sysid: return system identifier. ... sysid(3C)
sys_id: system identification file. ... sys_id(4)
identification. sysinfo: print system ... sysinfo(1)
setlogmask, vsyslog: control system/ syslog, openlog, closelog, ... syslog(3B)
syslogd: log systems messages. ... syslogd(1M)
performance values. sysmeter: meter display of system ... sysmeter(1)
system call. symsips: MIPS Computer Systems Inc. symsips(2)
symp: multiprocessing control. ... symp(2)
/strerror, errno, sys_errnolist, sys_nerr: system error messages. ... perror(3C)
call. syssgi: Silicon Graphics Inc. system ... syssgi(2)
psignal, sys_siganlist: system signal messages. ... psignal(3C)
osview: monitor operating system activity data. ... osview(1)
sal, sa2, sacd: system activity report package. ... sar(1M)
sar: system activity reporter. ... sar(1)
a command; report process data and system activity. timed: time ... timed(1)
add a secondary disk to the system. Add_disk: ... add_disk(1)
SA: devices administered by System Administration. ... sa(7)
sysadm: menu interface to do system administration. ... sysadm(1)
vadmin: interactive system administration tool. ... vadmin(1G)
ckbups: check file system backup schedule. ... ckbups(1M)
syggc: SGI graphics system call. ... syggc(2)
stat: data returned by stat system call. ... stat(5)
symsips: MIPS Computer Systems Inc. system call. ... symsips(2)
syssgi: Silicon Graphics Inc. system call. ... syssgi(2)
texturebind: SGI graphics system call. ... texturebind(2)
intro: introduction to system calls and error numbers. ... intro(2)
shutdown: shut down system, change system state. ... shutdown(1M)
time to allow synchronization of the system clock. adjtime: correct the ... adjtime(2)

Version 5.0 — 98 — April 1990
Silicon Graphics

uux: UNIX-to-UNIX system command execution. uux(1C)
table: system configuration information system(4)
uucp, uilog, uname: UNIX-to-UNIX system copy. uucp(1C)
cu: call another UNIX system. cu(1C)
types: primitive system data types. types(5)
dbg, debug: the debug file system. dbg(4)
dirview: graphical interface to file system. dirview(1G)
interactive script for shutting the system. systemd(1G)
errno, sys_errlist, sys_status: system error messages. perror, perror(3C)
oserro, setsockopt: get/set system error. oserror(3C)
uuto, uupick: public UNIX-to-UNIX system file copy. uuto(1C)
setup: initialize system for first user. setup(1)
efs: layout of the Extent file system. fs(4)
halt: halt the system. halt(1M)
uptime: show how long system has been up. uptime(1)
or print identifier of current host system: hostid: set hostid(1)
set or print name of current host system: hostname: hostname(1)
sys_id: system identification file. sys_id(4)
sysinfo: print system identification. sysinfo(1)
fsync: determine file system identifier. fsync(1M)
sysid: return system identifier. sysid(3C)
dirent: file system independent directory entry. dirent(4)
directory entries and put in a file system independent format. readdir
statfs, fstatfs: get file system information. statfs(2)
bg, bcheckrc: system initialization procedures. brc(1M)
xinit: X Window System initializer. xinit(1)
inode: format of an Extent File System inode. inode(4)
setlogmask, syslog: control system log. /openlog, closelog, syslog(3B)
xlogo: X Window System logo. xlogo(1)
lpadmin: configure the LP spooling system. lpadmin(1M)
xmessage: X window system message display program. xmessage(1)
mkfs: construct a file system. mkfs(1M)
gr_osview: graphical system monitor. gr_osview(1)
mount: mount a file system. mount(2)
getpagesize: get system page size. getpagesize(2)
symeter: meter display of system performance values. symeter(1)
stop all processes and halt the system. powerdown: powerdown(1M)
prf: operating system profiler. prf(7)
prfstat, prfnd, prfstat, prfpr: UNIX system profiler. prf(1M)
performed to stop the operating system. proc: run commands proc(1M)
reboot: reboot the system. reboot(1M)
/retr limit: control maximum system resource consumption. retrlimit(2)
a printer from the LP spooling system. rmprinter: remove rmprinter(1M)
save a core dump of the operating system. savecore: savecore(1M)
X: X Window System server. xserver(1)
psignal, sys_signal: system signal messages. psignal(3C)
shutdown: shut down system, change system state. shutdown(1M)
ustat: get file system statistics. ustat(2)
fsstat: report file system status. fsstat(1M)
rwhod: system status server. rwhod(1M)

April 1990 —99— Version 5.0
<table>
<thead>
<tr>
<th>Permutated Index</th>
<th>Silicon Graphics</th>
</tr>
</thead>
<tbody>
<tr>
<td>information table.</td>
<td>system: system configuration</td>
</tr>
<tr>
<td>mtab: mounted file</td>
<td>system table.</td>
</tr>
<tr>
<td>timezone: set default</td>
<td>system time zone.</td>
</tr>
<tr>
<td>deleting/preset: reset the lp queue</td>
<td>system to a pristine state by</td>
</tr>
<tr>
<td>multi: switch the</td>
<td>system to multi-user mode.</td>
</tr>
<tr>
<td>single: switch the</td>
<td>system to single-user mode.</td>
</tr>
<tr>
<td>Tab Window Manager for the X Window</td>
<td>System. twm:</td>
</tr>
<tr>
<td>sysfs: get file</td>
<td>system type information.</td>
</tr>
<tr>
<td>umount: umount a file</td>
<td>system.</td>
</tr>
<tr>
<td>uname: identify the current IRIX</td>
<td>system.</td>
</tr>
<tr>
<td>uname: get identity of current IRIX</td>
<td>system.</td>
</tr>
<tr>
<td>file transport program for the uucp</td>
<td>system. uucico:</td>
</tr>
<tr>
<td>change owner and group of a file</td>
<td>(System V and 4.3BSD). /chown:</td>
</tr>
<tr>
<td>lseek: move read/write file pointer</td>
<td>(System V and 4.3BSD).</td>
</tr>
<tr>
<td>BSDsetgrp: set process group ID</td>
<td>(System V and 4.3BSD). setgrp,</td>
</tr>
<tr>
<td>termino, terminos: general</td>
<td>System V and POSIX terminal/</td>
</tr>
<tr>
<td>closedir: directory operations</td>
<td>(System V). /seekdir, rewinddir,</td>
</tr>
<tr>
<td>signal: software signal facilities</td>
<td>(System V). /sigrelse, sigignore,</td>
</tr>
<tr>
<td>sigpause: signal management</td>
<td>(System V). /sigset</td>
</tr>
<tr>
<td>sysconf: get configurable</td>
<td>system variables (POSIX).</td>
</tr>
<tr>
<td>who: who is on the</td>
<td>who(1)</td>
</tr>
<tr>
<td>Uutry: try to contact remote</td>
<td>system with debugging on.</td>
</tr>
<tr>
<td>graphical interface to file</td>
<td>system. WorkSpace:</td>
</tr>
<tr>
<td>portable, network-transparent window</td>
<td>system. X: a</td>
</tr>
<tr>
<td>display program for the X Window</td>
<td>System... xman: Manual page</td>
</tr>
<tr>
<td>interface to the MH message handling</td>
<td>system. xmh: X</td>
</tr>
<tr>
<td>SGI Irix server for the X Window</td>
<td>System. Xsgi:</td>
</tr>
<tr>
<td>shutting the system down..</td>
<td>systemdow: interactive script for</td>
</tr>
<tr>
<td>fsck, dfck: check and repair file</td>
<td>systems.</td>
</tr>
<tr>
<td>sysmips: MIPS Computer</td>
<td>Systems Inc. system call.</td>
</tr>
<tr>
<td>driver. dks: Small Computer</td>
<td>Systems Interface (SCSI) disk</td>
</tr>
<tr>
<td>labelit: provide labels for file</td>
<td>systems.</td>
</tr>
<tr>
<td>syslogd: log</td>
<td>systems messages.</td>
</tr>
<tr>
<td>mount, unmount multiple file</td>
<td>systems. mountall, mountall:</td>
</tr>
<tr>
<td>command file. cshrc:</td>
<td>system-wide csh initialization</td>
</tr>
<tr>
<td>interface card. t3270: Silicon Graphics 3270</td>
<td>t3270: Silicon Graphics 3270</td>
</tr>
<tr>
<td>System. twm: Tab Window Manager for the X Window</td>
<td>t3270(7)</td>
</tr>
<tr>
<td>bsearch: binary search a sorted</td>
<td>table.</td>
</tr>
<tr>
<td>rehash: recompute command hash</td>
<td>table.</td>
</tr>
<tr>
<td>unhash: discard command hash</td>
<td>table.</td>
</tr>
<tr>
<td>/access routine for the symbol</td>
<td>table definition file in archives.</td>
</tr>
<tr>
<td>symbol name for object file symbol</td>
<td>table entry. idgename: retrieve</td>
</tr>
<tr>
<td>/compute the index of a symbol</td>
<td>table entry of a common object file.</td>
</tr>
<tr>
<td>ldbread: read an indexed symbol</td>
<td>table entry of a common object file.</td>
</tr>
<tr>
<td>provide a compilation unit symbol</td>
<td>table interface. /routines that</td>
</tr>
<tr>
<td>mtab: mounted file system</td>
<td>table.</td>
</tr>
<tr>
<td>ldbseek: seek to the symbol</td>
<td>table of a common object file.</td>
</tr>
<tr>
<td>setmnt: establish mount</td>
<td>table.</td>
</tr>
<tr>
<td>get descriptor</td>
<td>table size INITGROUPS(3X).</td>
</tr>
<tr>
<td>descriptor section of the symbol</td>
<td>table. /provide access to per file</td>
</tr>
<tr>
<td>to access and add to the symbol</td>
<td>table. /to basic functions needed</td>
</tr>
</tbody>
</table>

Version 5.0

April 1990
Silicon Graphics

interface to the MIPS symbol
routines to print the symbol
system configuration information
classification and conversion
hdestroy: manage hash search
hyroute: set the HyperNet routing
tabs: set
	tabs: set tabs on a terminal.
script with an identifying/ tag:
tags: create a file.
talk: talk to another user.
talkd: remote user communication.
fcos, ftn, fasin, facos, / sin, cos,
hyperbolic functions. sinh, cosh,
tanh, fsinh, fcoosh, ftnanh:
tap: tape archiver.
ts: ISI VME-QIC2/X cartridge
tape controller.
xmt: Xylogics 1/2 inch magnetic tape controller.
anisitape: ANSI standard tape handler.
mtio: magnetic tape interface.
tps: SCSI 1/4-inch Cartridge tape interface.
list the contents of a given backup tape. List_tape:
mt: magnetic tape manipulating program.
mkboottape: make a boot tape.
vmsprep: VMS tape preparation aid.
the specified file or directory from tape. Restore: restore
tar: tape archiver.
/transferdevice for performing transferdevice for performing tar/
tar within the WorkSpace.
tarArchive: an interactive task.
task: operations on a task.
taskcreate: create a new task.
taskdestroy: destroy a task.
tasksetblockcnt: routines to
taskblock, taskunblock,
taskcreate: create a new task.
taskctl: operations on a task.
taskdestroy: destroy a task.
generate programs for simple lexical
routines to block/unblock
taskblock, taskunblock,
routines to taskblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
tasksetblockcnt: routines to/
taskblock, taskunblock,
Permutated Index

tcflush (int fildes, int/ .
primitives #include int/ tcdrain,
tcflush, tcflow: posix line control .
tcsendbreak, 
/state primitives #include int
 tcgetattrs (int fildes, struct/ .
tcssetattr, 
state primitives #include int/
tcgetattr: posix get/set terminal .
tcsetattr, 
group primitives #include int/ .
tcgetprgr (int fildes): /process .
tcgetprgr, 
convert date and time to string
Protocol.
tcp: Internet Transmission Control .
tcp(7P) .
mapper. portmap: TCP,UDP port to RPC program number
line control primitives #include int
(tcsendbreak (int fildes, int/ /posix .
tcsendbreak, 
(int fildes, pid_t pgpr_id); .
TCSENDBREAK(3T). tcsetprgr .
tcsendbreak, 
exception handler package .
tcsetattr(3T). floating-point .
tcsetattr(3T). 
pgrp_id); TCSENDBREAK(3T). .
tcgetprgr (int fildes, pid_t .
tcgetprgr, 
process group primitives #include/
tcsetprgr: posix get/set foreground .
tcgetprgr, 
trees. tsearch, tfind, .
tcdelete, tcwalk: manage binary search .
tsearch(3C)
 tee: pipe fitting. .
tee(1)
 initialization. init, .
telnet: process control .
telnetd: process control .
(closedir/ opendir, readdir, .
telldir, seekdir, rewinddir, .
telldir, seekdir, rewinddir, .
telnet: Internet TELNET protocol server.
telnet: User interface to the TELNET
server. .
telnet: Internet TELNET protocol
server.
temporary file. tmpnam, .
tmpnam: create a name for a .
tmpnam(3S)
tmpfile: create a .
tmpfile: create a
 temporary file. .
tmpfile(3S)
tmpnam, tempnam: create a name for a temporary file. .
tempnam(3S)
terminals. 
terminal. .
terminals.
term: format of compiled .
term: format of compiled .
term: format of compiled
 term file. .
term file. .
term file.
term: format of compiled term file.
term(4)
term(4)
term(4)
term(4)
termcap and terminal settings to .
termcap description. captoinfo: convert a
termcap description into a terminfo .
termcap(4)
terminfo: .
terminal capability data base. .
terminfo(4)
termcap(4)
terminals. .
terminal. .
terminal.
terminal.
terminal. .
terminal.
terminal.
ct: spawn getty to a remote .
terminal. .
terminal. .
terminal.
terminal emulator for X. .
terminal emulator.
terminal.
terminal. .
terminal.
ctermid: generate file name for
 tset: .
terminal dependent initialization. .
terminal dependent initialization.
tset(1)
tset(1)
tset(1)
terminal driver. .
terminal driver. .
terminal driver.
terminal.
terminal.
terminal.
pty: pseudo .
terminal. .
terminal.
terminal.
terminal.
terminal.
xterm: .
terminal emulator.
terminal.
terminal.
terminal.
pterm: NeWS .
terminal.
terminal.
terminal.
terminal.
launch applications that require a
 terminal. .
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
terminal.
Silicon Graphics

ttyname, isatty: find name of a
line discipline. getty: set
terminal type, modes, speed, and
terminal types by port.
"hangup" the current control
terminals. last:
terminals. conventional names for
terminals.
kill: terminate a process.
esac: terminate case.
endif: terminate conditional.
fi: terminate conditional.
core dump. abort:
terminate current process with a
end: terminate loop.
done: terminate loop.
exit, _exit: terminate process.
endsw: terminate switch.
wait for child processes to stop or
terminate. wait, waitpid, wait3:
termios *termios_p):
infocmp: compare or print out
terminfo description. captinfo:
terminfo description. infocmp:
terminfo: terminal capability data
termios: general System V
and POSIX terminal interfaces.
termio, termios: general System V and POSIX
termio, termios, termio, termio,
#include speed_t cgetispeed (struct
termios *termios_p): /primitives
cgetspeed, speed_t cgetispeed (struct
termios *termios_p): /speed_t
GETLVENTS(3C). cgetispeed (struct
termios *termios_p, speed_t speed);
termios速度(p),速度
termios *termios_p): /speed_t speed;
termios *termios_p, speed_t speed;
termios *termios_p, speed_t speed;
termios *termios_p, speed_t speed;
choice: selection
int tcgetattr (int fildes, struct
termios速度(p),速度
*termios_p): /include
termios速度(p),速度
*termios_p): /include
*termios_p, speed_t speed;
*termios_p, speed_t speed;
*termios_p, speed_t speed;
*termios_p):
choice: selection
 expansion

ed, red: text editor.
ex:
exedit: simple
text editor for X.
jot: a simple mouse-based
text editor.
casual users). edit:
text editor (variant of ex for
newform: change the format of a
text file.
fspec: format specification in
text files.
fmt: simple
text formatter.
plock: lock process,
text, or data in memory.
jotview: a simple mouse-based
text viewer.
textcolors: set the colors used by a
text window.
textcolors: set the colors used by a
text window.
texturebind: SGI graphics system
texturebind(2)
search trees. tsearch,
tfind, tdelete, twalk: manage binary
tftp: trivial file transfer program.
Transfer Protocol server: tftpd: Internet Trivial File
fatan2: trigonometric functions and
if, then: conditional statement.
manush: display a man page and
then prompt for input.
usputinfo: exchange information
through a universe.
three colored lights bouncing around
merges: three-way file merge.
tic: terminfo compiler.
and system activity.
time: time a command; report process data
time: time a command.
batch: execute commands at a later
time. at
graphs demographic data in 3D over
time. demographic
settimeofday: get/set date and
time. gettimeofday,
oclock: display
time of day.
profil: execution
time profile.
setting up an environment at login
time. profile.
timed: time server daemon.
stime: set
time.
time: time a command.
time: time command.
system clock.
adjtime: correct the
time to allow synchronization of the
strftime, tzset: convert date and
time to string TCGETPGRP(3T).
clock: report CPU
time used.
timezone: set default system
time zone.
timed: timed control program.
function. sginap:
timed sleep and processor yield
timed: time server daemon.
timedc: timed control program.
blanktime: set the screen blanking
timeout.
setitimer: get/set value of interval
times.
times: get process and child process
difference between two calendar
times INSQUE(3), compute
update access and modification
times of a file. touch.
times: print accumulated times.
times: print accumulated
etimes: set file
times.
times: get process and child process
set file access and modification
times. utime:
etimes: set file
better one.
timeslave: 'slave' local clock to a
process data and system activity.
time zone.
symbolic links.
tlmk: clone a file tree using
tmpfile: create a temporary file.
tmpnam, tempnam: create a name for a
/fisgraph, isascii, tolower, toupper,
/tolower, _toupper, _tolower,
and white.
tobw: convert a color image to black
topen, pclose: initiate pipe
to/from a process.
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>toupper, tolower, _toupper, _tolower, toascii: translate/</td>
<td>conv(3C)</td>
</tr>
<tr>
<td>/isascii, toupper, toascii, _tolower, _toupper, sethclass/</td>
<td>ctype(3C)</td>
</tr>
<tr>
<td>/ispunct, ispunct, isgraph, isascii, toascii: translate/ toupper, toascii, _tolower, _toupper,</td>
<td>ctype(3C)</td>
</tr>
<tr>
<td>local and network disk mounts tool: disks: interactive</td>
<td>disks(1G)</td>
</tr>
<tr>
<td>inst: software installation tool</td>
<td>inst(1M)</td>
</tr>
<tr>
<td>interactive system administration tool: vadm:</td>
<td>vadm(1G)</td>
</tr>
<tr>
<td>versions: software versions tool</td>
<td>versions(1M)</td>
</tr>
<tr>
<td>4Sight event record and playback toolchest: journalchest:</td>
<td>journalchest(1W)</td>
</tr>
<tr>
<td>4Sight utility and windowing toolchest, windowchest, demochest:</td>
<td>toolchest(1W)</td>
</tr>
<tr>
<td>highest CPU usage. top: display processes having</td>
<td>top(1)</td>
</tr>
<tr>
<td>tosrt: topological sort.</td>
<td>tosrt(1)</td>
</tr>
<tr>
<td>acctmerg: merge or add total accounting files.</td>
<td>acctmerg(1M)</td>
</tr>
<tr>
<td>modification times of a file. touch: update access and</td>
<td>touch(1)</td>
</tr>
<tr>
<td>/tolower, toupper, toascii, _tolower, _toupper, _tolower, _toupper, sethclass: character/</td>
<td>ctype(3C)</td>
</tr>
<tr>
<td>/ispunct, isgraph, isascii, toupper,</td>
<td>ctype(3C)</td>
</tr>
<tr>
<td>translate/toupper, tolower, toascii: translate/ interface.</td>
<td>conv(3C)</td>
</tr>
<tr>
<td>terminfo database.</td>
<td>conv(3C)</td>
</tr>
<tr>
<td>tpnt: initialize a terminal or query</td>
<td>tpt(1)</td>
</tr>
<tr>
<td>tr: translate characters.</td>
<td>tr(1)</td>
</tr>
<tr>
<td>ptrace: process trace.</td>
<td>ptrace(2)</td>
</tr>
<tr>
<td>goto: command transfer.</td>
<td>csh(1)</td>
</tr>
<tr>
<td>selecting entries in the workspace transfer menu /visual interface for</td>
<td>transfermanager(1G)</td>
</tr>
<tr>
<td>ftp: Internet file transfer program.</td>
<td>ftp(1C)</td>
</tr>
<tr>
<td>ftp: trivial file transfer program.</td>
<td>tftp(1C)</td>
</tr>
<tr>
<td>ftpd: Internet File Transfer Protocol server.</td>
<td>tftp(1M)</td>
</tr>
<tr>
<td>tftp: Internet Trivial File Transfer Protocol server.</td>
<td>tftp(1M)</td>
</tr>
<tr>
<td>specification for extending the/ transferdevice: a shell script</td>
<td>transferdevice(4)</td>
</tr>
<tr>
<td>within/ cpioArchive: an interactive transferdevice for performing cpio</td>
<td>cpioarchive(1)</td>
</tr>
<tr>
<td>within/ repDevice: an interactive transferdevice for performing rep</td>
<td>repdevice()</td>
</tr>
<tr>
<td>within/ tarArchive: an interactive transferdevice for performing tar</td>
<td>tararchive(1)</td>
</tr>
<tr>
<td>interface for selecting entries in/ transfermanager: provide a visual</td>
<td>transfermanager(1G)</td>
</tr>
<tr>
<td>_toupper, _tolower, toascii: translate characters. _tolower, _toupper,</td>
<td>conv(3C)</td>
</tr>
<tr>
<td>tr: translate characters.</td>
<td>tr(1)</td>
</tr>
<tr>
<td>RGB image. mapping: translates a screen image into an</td>
<td>mapping(1G)</td>
</tr>
<tr>
<td>tcp: Internet Transmission Control Protocol.</td>
<td>tcp(7P)</td>
</tr>
<tr>
<td>encode/decode a binary file for transmission via mail /udecode:</td>
<td>uencode(1C)</td>
</tr>
<tr>
<td>uucico: file transport program for the uucp</td>
<td>uucico(1M)</td>
</tr>
<tr>
<td>the scheduler for the uucp file transport program. unsched:</td>
<td>unsched(1M)</td>
</tr>
<tr>
<td>scripts. trap: process interrupts in command</td>
<td>sh(1)</td>
</tr>
<tr>
<td>ftw: walk a file tree.</td>
<td>ftw(3C)</td>
</tr>
<tr>
<td>tlink: clone a file tree using symbolic links.</td>
<td>tlink(1)</td>
</tr>
<tr>
<td>tdelete, twalk: manage binary search trees. search, tfind,</td>
<td>search(3C)</td>
</tr>
<tr>
<td>/fian, fasin, facos, fatan, fatan2: trigonometric functions and their/</td>
<td>trig(3M)</td>
</tr>
<tr>
<td>tftp: trivial file transfer program.</td>
<td>tftp(1C)</td>
</tr>
<tr>
<td>server. tftp: Internet Trivial File Transfer Protocol</td>
<td>tftp(1M)</td>
</tr>
<tr>
<td>true, false: provide truth values.</td>
<td>true(1)</td>
</tr>
<tr>
<td>/ceil, fceil, fmod, fabs, int, length. truncate, truncatem: truncate a file to a specified</td>
<td>truncate(2)</td>
</tr>
</tbody>
</table>

April 1990 –105– Version 5.0
to a specified length.
absolute value, nearest integer, and
rhosts: list of
hosts.equiv: list of
trusted hosts.
mips, 4d, 4d60: get processor type
true, false: provide
debugging on. Uutry:
controller.
t entail, tfind, tdelete, twalk:
initialization.
tset: terminal dependent
tsort: topological sort.
tty: controlling terminal interface.
tty: get the name of the terminal.
file of the current user.
by port.
/prtacct, runacct, shutacct, startup,
tsearch, tfind, tdelete,
Window System.
file: determine file
type.
sysfs: get file system
type information.
type, modes, speed, and line
by type.
type: data base of terminal types
type rules. isSuper: supertype
checking utility for use with file
ttype: data base of terminal
types by port.
types: primitive system data types.
syss: script: make
typescript of terminal session.
tzset: convert date and time to/
tztime: convert time to
mips, 4d, 4d60: get/ pdp11,
4d60: get/ pdp11, u3b, u3b2, u3b5,
/mips: 4d, 4d60: get/ pdp11, u3b
/mips: 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k,
/u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/mips: 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
mips, 4d, 4d60: get/ pdp11, u3b
/u3b5, u3b15, vax, m68k, m68000,
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file systems</td>
<td>mountall, unmountall, unalias, umount, umount multiple</td>
</tr>
<tr>
<td></td>
<td>sh(1)</td>
</tr>
<tr>
<td>gather</td>
<td>system, unalign references, unalign(3X)</td>
</tr>
<tr>
<td>statistics</td>
<td>system, unname: get identity of current IRIX, unname(2)</td>
</tr>
<tr>
<td>on system</td>
<td>unname: identify the current IRIX, unname(1)</td>
</tr>
<tr>
<td>blockprocall</td>
<td>blockproc, unblockproc, setblockprocent, blockproc(2)</td>
</tr>
<tr>
<td>/blockprocall</td>
<td>setblockprocent, blockprocall:/ blockproc(2)</td>
</tr>
<tr>
<td>cachectl</td>
<td>cachectl(2)</td>
</tr>
<tr>
<td></td>
<td>uncacheable, uncompress, zcat: compress and compress(1)</td>
</tr>
<tr>
<td></td>
<td>under the mouse pointer, snoop, snoop(6D)</td>
</tr>
<tr>
<td>magnify</td>
<td>Interactive Generation of figures</td>
</tr>
<tr>
<td>and report on</td>
<td>under X11. xfig: Facility for xfig(1)</td>
</tr>
<tr>
<td>the screen</td>
<td>ul: do</td>
</tr>
<tr>
<td></td>
<td>underlining, ul(1)</td>
</tr>
<tr>
<td></td>
<td>unget: undo a previous get of an unget(1)</td>
</tr>
<tr>
<td></td>
<td>SCCS file unget: undo a previous get of an unget(1)</td>
</tr>
<tr>
<td></td>
<td>input stream: ungetc: push character back into ungetc(3S)</td>
</tr>
<tr>
<td></td>
<td>unhash: discard command hash table, sh(1)</td>
</tr>
<tr>
<td></td>
<td>define: strip or reduce ifdefs in C undef(1)</td>
</tr>
<tr>
<td>/rand48, seed48</td>
<td>drain: capture file unicom: report repeated lines in a uniq(1)</td>
</tr>
<tr>
<td></td>
<td>mktemp: make a mktemp(3C)</td>
</tr>
<tr>
<td></td>
<td>gethostid: get/set constants</td>
</tr>
<tr>
<td></td>
<td>uniqu: file header for symbolic unistd(4)</td>
</tr>
<tr>
<td></td>
<td>routines that provide a compilation</td>
</tr>
<tr>
<td></td>
<td>unit symbol table interface, stcu: stcu(3X)</td>
</tr>
<tr>
<td></td>
<td>units: conversion program</td>
</tr>
<tr>
<td></td>
<td>cu: call another UNIX system cu(1C)</td>
</tr>
<tr>
<td></td>
<td>prfstat, prfdc, prfsm, prfrpr: prfprf(1M)</td>
</tr>
<tr>
<td></td>
<td>execution: uux UNIX system command uux(1C)</td>
</tr>
<tr>
<td></td>
<td>uucp, uulog, uuname: UNIX-to-UNIX system copy, uucp(1C)</td>
</tr>
<tr>
<td></td>
<td>uuto, uupick: public UNIX-to-UNIX system file copy, uuto(1C)</td>
</tr>
<tr>
<td></td>
<td>unlink: link and unlink(1M)</td>
</tr>
<tr>
<td></td>
<td>link, link: link and unlink(1M)</td>
</tr>
<tr>
<td></td>
<td>link: remove directory entry, link(2)</td>
</tr>
<tr>
<td></td>
<td>munmap: munmap pages of memory munmap(2)</td>
</tr>
<tr>
<td></td>
<td>umount: umount a file system, umount(2)</td>
</tr>
<tr>
<td></td>
<td>mountall, umountall: mount, mountall(1M)</td>
</tr>
<tr>
<td></td>
<td>pack, unpack: compress and expand files, pack(1)</td>
</tr>
<tr>
<td>host entry</td>
<td>unregisterhost: remove the existing unregisterhost(3N)</td>
</tr>
<tr>
<td>in yp hosts</td>
<td>unset: discard shell variables, sh(1)</td>
</tr>
<tr>
<td>data base</td>
<td>unsetenv: remove environment, sh(1)</td>
</tr>
<tr>
<td></td>
<td>variables.</td>
</tr>
<tr>
<td></td>
<td>debugging: setsym: set up a debug kernel for symbolic setsym(1)</td>
</tr>
<tr>
<td></td>
<td>profile: setting up an environment at login time, profile(4)</td>
</tr>
<tr>
<td></td>
<td>client: xstart: start up the sgi X server as a NeWS xstart(1)</td>
</tr>
<tr>
<td></td>
<td>show how long system has been of a file. touch:          up time: uptime(1)</td>
</tr>
<tr>
<td></td>
<td>programs: make: maintain, update, and regenerate groups of make(1)</td>
</tr>
<tr>
<td></td>
<td>Isearch, Ifind: linear search and update Isearch(3C)</td>
</tr>
<tr>
<td></td>
<td>sync: update super block, sync(2)</td>
</tr>
<tr>
<td></td>
<td>sync: update the super block, sync(1M)</td>
</tr>
<tr>
<td></td>
<td>been up. uptime: show how long system has uptime(1)</td>
</tr>
<tr>
<td></td>
<td>du: summarize disk usage. du(1M)</td>
</tr>
</tbody>
</table>

April 1990

--107--

Version 5.0
Permutated Index

display processes having highest CPU usage.
usmalloc, usmalloc, usmalloc, usmalloc:
configuration operations.
supervisor, and fails if not
usmslock: spinlock/ usmslock:
operations.
exchange information though an area
information about a specific lock
manual page "whatis" database for
function key binding facility for
supertype checking utility for
textcolors: set the colors
speed and terminal settings
clock: report CPU time
id: print
setuid, setegid, setgids: set
setuid, setgids: set
talk: remote
cron: user cron tab file.
login: login new
get character login name of the
udp: Internet
/gemuid, getgid, getegid: get real
environ: user environment.
generate disk accounting data by
whoami: print effective current
autologin: set autologin
setreuid: set real and effective
fingerprint: user information lookup program.
finger: user information server.
demos. buttonfly: a pretty
protocol. telnet:
ulimit: get and set
logname: return login name of

ds: generic
xset:
/setup: initialize system for first
/usmalloc, usmalloc, usmalloc:
su: become super-user or another
talk: talk to another
slot in the umtp file of the current
write: write to another
last: indicate last logins of
terminator (variant of ex for casual
mail: send mail to
rhosts: list of trusted hosts and
wall: write to all
usmalloc, usmalloc, usmalloc,
usage in a window.
gr_top: gr_top(1)
top: top(1)
usmalloc, usmalloc, usmalloc:
usconfig: semaphore and lock arena
uspsmap: attempts to acquire a
usmslock, usmslock, usmslock:
usmslock: lock control operations.
uscspsmap: semaphore control
USDUMPLOCK(3P), uspinfo:
USDUMPSEMAMAP(3P), dump out
usdumplock:
use with apropos. makewhatis: make
use with. bindkey:
use with file type rules. isSuper: issupe(1)
used by getty. gettydefs:
used.
clock(3C)
id(1)
setreuid, getegid: set
setuid(3C)
setreuid(2)
talkd(1M)
talkd(1M)
cron(1G)
csh(1)
cuserid(3S)
udp(7P)
getuid(2)
environ(5)
disksh(1M)
whoami(1)
autologin(4)
setreuid(2)
finger(1)
finger(1M)
user interface for Silicon Graphics
buttonfly(6D)
telnet(1C)
ulimit(2)
logname(3X)
ds(7M)
xset(1)
getuid(2)
setup(1)
usmalloc(3P)
su(1M)
talk(1)
usyslog(3C)
write(1)
users and terminals.
lst(1)
edi(1)
mail, mailrc(1)
rhosts(4)
users.
users.
users.
usmalloc(3P)
meter display of system performance
true, false: provide truth
values: machine-dependent
varargs: variable argument list.
stdarg: variable argument list.
print formatted output of a
set: change value of shell
for the NEWSERVERS
environment
@: arithmetic on shell
unset: discard shell
unsetenv: remove environment
sgiopt: get or set non-volatile RAM
fpathconf: get configurable pathname
sysconf: get configurable system
readonly: make shell
export: add shell
edit: text editor
get/ pdp11, u3b, u3b2, u3b5, u3b15,
vax, m68k, m68000, mips, 4d, 4d60:
mach(1)
get option letter from argument
display editor based on/ vi, view,
stored without run length encoding.
assert: program
converter. x10tox11: X
x10tox11: X version 10 to
vc: version control.
set group access list (berkeley 4.3
get: get a
get group access list (berkeley 4.3
scsdiff: compare two
versions: software
virtual memory efficient way.
output of a variable/ vprintf,
(current control terminal.
(visual) display editor based on/
a binary file for transmission
mail: receive mail
setmon: set the default monitor
shadow: full-screen armchair pilot's
(visual) display editor based/ vi,
gview:
jetview: a simple mouse-based text
rlnotes: on-line release notes
xscape: X Window Protocol
page: file perusal filter for crn
values: machine-dependent values.
symeter:
values.
values: machine-dependent
varargs: variable argument list.
stdarg: variable argument list.
varargs: variable argument list.
print formatted output of a
set: change value of shell
for the NEWSERVERS
environment
@: arithmetic on shell
variables.
variables.
variables.
variables.
variables.
variables.
variables.
variables.
vax, m68k, m68000, mips, 4d, 4d60:
mach(1)
vc: version control.
get option letter from argument
vector. getopt: 
display editor based on/ vi, view,
stored without run length encoding.
assert: program
converter. x10tox11: X
x10tox11: X version 10 to
vc: version control.
set group access list (berkeley 4.3
get: get a
group access list (bsd 4.3
get group access list (berkeley 4.3
scsdiff: compare two
versions: software
virtual memory efficient way.
output of a variable/ vprintf,
(current control terminal.
(visual) display editor based on/
a binary file for transmission
mail: receive mail
setmon: set the default monitor
shadow: full-screen armchair pilot's
(visual) display editor based/ vi,
gview:
jetview: a simple mouse-based text
rlnotes: on-line release notes
xscape: X Window Protocol
page: file perusal filter for crn
values: machine-dependent values.
symeter:
values.
values: machine-dependent
varargs: variable argument list.
stdarg: variable argument list.
varargs: variable argument list.
print formatted output of a
set: change value of shell
for the NEWSERVERS
environment
@: arithmetic on shell
variables.
variables.
variables.
variables.
variables.
variables.
variables.
variables.
vax, m68k, m68000, mips, 4d, 4d60:
mach(1)
vc: version control.
get option letter from argument
vector. getopt: 
display editor based on/ vi, view,
stored without run length encoding.
assert: program
converter. x10tox11: X
x10tox11: X version 10 to
vc: version control.
set group access list (berkeley 4.3
get: get a
group access list (bsd 4.3
get group access list (berkeley 4.3
scsdiff: compare two
versions: software
virtual memory efficient way.
output of a variable/ vprintf,
(current control terminal.
(visual) display editor based on/
a binary file for transmission
mail: receive mail
setmon: set the default monitor
shadow: full-screen armchair pilot's
(visual) display editor based/ vi,
gview:
jetview: a simple mouse-based text
rlnotes: on-line release notes
xscape: X Window Protocol
page: file perusal filter for crn
values: machine-dependent values.
symeter:
values.
values: machine-dependent
varargs: variable argument list.
stdarg: variable argument list.
varargs: variable argument list.
print formatted output of a
set: change value of shell
for the NEWSERVERS
environment
@: arithmetic on shell
variables.
variables.
variables.
variables.
variables.
variables.
variables.
variables.
vax, m68k, m68000, mips, 4d, 4d60:
mach(1)
vc: version control.
get option letter from argument
vector. getopt: 
display editor based on/ vi, view,
stored without run length encoding.
assert: program
converter. x10tox11: X
x10tox11: X version 10 to
vc: version control.
set group access list (berkeley 4.3
get: get a
group access list (bsd 4.3
get group access list (berkeley 4.3
scsdiff: compare two
versions: software
virtual memory efficient way.
output of a variable/ vprintf,
(current control terminal.
(visual) display editor based on/
a binary file for transmission
mail: receive mail
setmon: set the default monitor
shadow: full-screen armchair pilot's
(visual) display editor based/ vi,
gview:
jetview: a simple mouse-based text
rlnotes: on-line release notes
xscape: X Window Protocol
page: file perusal filter for crn
values: machine-dependent values.
symeter:
values.
values: machine-dependent
varargs: variable argument list.
stdarg: variable argument list.
varargs: variable argument list.
print formatted output of a
set: change value of shell
for the NEWSERVERS
environment
@: arithmetic on shell
variables.
variables.
variables.
variables.
variables.
variables.
variables.
variables.
vax, m68k, m68000, mips, 4d, 4d60:
mach(1)
vc: version control.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vfork:</td>
<td>spawn new process in a virtual memory efficient way.</td>
</tr>
<tr>
<td>vi, view, vedit:</td>
<td>screen-oriented terminal. vhangup:</td>
</tr>
<tr>
<td>entries/ transmanager:</td>
<td>provide a virtual interface for selecting</td>
</tr>
<tr>
<td>flyray:</td>
<td>a network. a visualized raytracer.</td>
</tr>
<tr>
<td>dglray:</td>
<td>a visualized raytracer running across</td>
</tr>
<tr>
<td>visuallogin:</td>
<td>login process control.</td>
</tr>
<tr>
<td>gamcal:</td>
<td>visually check display calibration.</td>
</tr>
<tr>
<td>driver for National Instruments</td>
<td>VME IEEE-488 controller. gpib:</td>
</tr>
<tr>
<td>controller. ts:</td>
<td>ISI</td>
</tr>
<tr>
<td>vmsprep:</td>
<td>VMS tape preparation aid.</td>
</tr>
<tr>
<td>lvinit:</td>
<td>initialize logical volume devices.</td>
</tr>
<tr>
<td>lv: logical</td>
<td>volume Disk driver.</td>
</tr>
<tr>
<td>dvhtool: modify and obtain disk</td>
<td>volume header information.</td>
</tr>
<tr>
<td>prtvtoc: print</td>
<td>volume header information.</td>
</tr>
<tr>
<td>vh: disk</td>
<td>volume header.</td>
</tr>
<tr>
<td>mklv: construct or extend a logical volume</td>
<td>volumes.</td>
</tr>
<tr>
<td>and restore consistency of logical volumes. lvck: check</td>
<td></td>
</tr>
<tr>
<td>lvtab: information about logical volumes.</td>
<td></td>
</tr>
<tr>
<td>formatted output of a variable/</td>
<td>vprintf, fprintf, vfprintf: print</td>
</tr>
<tr>
<td>a variable/ vprintf, fprintf,</td>
<td>vprintf: print formatted output of</td>
</tr>
<tr>
<td>/openlog, closelog, setlogmask, doing.</td>
<td>vsyslog: control system log.</td>
</tr>
<tr>
<td>complete. wait:</td>
<td></td>
</tr>
<tr>
<td>terminate. wait, waitpid, wait3:</td>
<td></td>
</tr>
<tr>
<td>/release blocked signals and wait for child processes to stop or</td>
<td></td>
</tr>
<tr>
<td>/release blocked signals and wait for interrupt (4.3BSD).</td>
<td></td>
</tr>
<tr>
<td>to complete.</td>
<td></td>
</tr>
<tr>
<td>to complete.</td>
<td></td>
</tr>
<tr>
<td>processes to stop or terminate. wait, waitpid, wait3:</td>
<td></td>
</tr>
<tr>
<td>stop or terminate. wait, waitpid, wait3:</td>
<td></td>
</tr>
<tr>
<td>processes to stop or/ wait,</td>
<td></td>
</tr>
<tr>
<td>ftw:</td>
<td>walk a file tree.</td>
</tr>
<tr>
<td>insect: simulates a walking, six-legged creature/robot...</td>
<td></td>
</tr>
<tr>
<td>wall: write to all users.</td>
<td></td>
</tr>
<tr>
<td>dialwarp, keywarp: set input</td>
<td>warping parameters. mousewarp,</td>
</tr>
<tr>
<td>xeyes:</td>
<td>watch over your shoulder.</td>
</tr>
<tr>
<td>of the surface of an idealized waterbed. waterbed. /real-time simulation</td>
<td></td>
</tr>
<tr>
<td>surface of an idealized waterbed.</td>
<td></td>
</tr>
<tr>
<td>wc: word count.</td>
<td></td>
</tr>
<tr>
<td>whatis: describe</td>
<td>what a command is.</td>
</tr>
<tr>
<td>what: identify SCCS files.</td>
<td></td>
</tr>
<tr>
<td>w: who is on and what they are doing.</td>
<td></td>
</tr>
<tr>
<td>whodo: who is doing what.</td>
<td></td>
</tr>
<tr>
<td>makes: make manual page</td>
<td>&quot;whatis&quot; database for use with/</td>
</tr>
<tr>
<td>whatis: describe what a command is.</td>
<td></td>
</tr>
</tbody>
</table>

whereis: locate source, binary, and
which: locate a program file
while: repeat commands
while: repeat commands
while/for loop.
while/foreach loop.
white.
who is doing what.
who is on and what they are doing.
who is on the system.
who is on the system.
who is on the system.
who is doing what.
who's logged in on local machines.
w.
lstres: list resources in
width output device.
window and request a response.
window based debugger.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.
window.

widgets.
listres(1)
fold(1)
confirm(1G)
edge(1)
clock(6D)
xdpr(1)
xpr(1)
window(5W)
inform(1G)
xwininfo(1)
xlswins(1)
muncher(1)
xsetroot(1)
plaid(1)
xscope(1)
wh(1G)
xinit(1)
xlogo(1)
twm(1)
xserver(1)
x(1)
xman(1)
xs(1G)
xwd(1)
toolchest(1W)
toolchest(1W)
winicons(5W)
cpioarchive(1)
within the WorkSpace.../interactive
transferdevise for performing rcp within the WorkSpace.../interactive... rcpdevice()
transferdevise for performing tar within the WorkSpace.../interactive... tararchive(1)
/force an image to be stored without run length encoding... verbatim(6D)
w: word count... wc(1)
fgets, getw: get character or word from a stream. getc, getwchar... getc(3S)
putc, putw: put character or word on a stream. putc, putwchar... putc(3S)
do, for: loop over list of words... sh(1)
cd: change working directory... cd(1)
chdir: change working directory... chdir(2)
getcwd: get path-name of current working directory... getcwd(3C)
file descriptor. fchdir: change working directory, given an open... fchdir(2)
pwd: working directory name... pwd(1)
getwd: get current working directory pathname... getwd(3C)
for performing cpio within the WorkSpace.../transferdevice... cpioarchive(1)
file system. WorkSpace: graphical interface to... workspace(1G)
specification for extending the WorkSpace menu functions.../script... transferdevice(4)
for performing rcp within the WorkSpace.../transferdevice... rcpdevice()
for performing tar within the WorkSpace.../transferdevice... tararchive(1)
for selecting entries in the workspace transfer menu.../interface... transfermanager(1G)
gse: Silicon Graphics 5080 workstation interface card... gse(7)
allocate internet address for workstation. registerinethost... registerinethost(3N)
write: write on a file... write(2)
GETUSAGE(3). write output gathered from buffers... writev()
putpwent: write password file entry... putpwent(3C)
wall: write to all users... wall(1)
write: write to another user... write(1)
write: write on a file... write(2)
write: write to another user... write(1)
read input to scattered buffers... WRITERV(3C)
open: open for reading or writing... open(2)
shell. wsh: creates and specifies a window... wsh(1G)
uttmp, wttmp: utmp and wttmp entry formats... uttmp(4)
uttmp, wttmp: utmp and wttmp entry formats... uttmp(4)
accounting records. ftwtmp, fwtmpfix: manipulate connect... fwtmp(1M)
window system. X: a portable, network-transparent... x(1)
xauth: X authority file utility... xauth(1)
editor and converter utilities for X. bitmap, bmont, atomb: bitmap... bitmap(1)
xclipboard: X clipboard client... xclipboard(1)
MIT X Consortium... x consortium(1)
xdm: X Display Manager... xdm(1)
xev: print contents of X events... xev(1)
xgc: X graphics demo... xgc(1)
handling system. xmh: X interface to the MH message... xmh(1)
puzzle: puzzle game for X... puzzle(1)
xkill: kill a client by its X resource... xkill(1)
xrefresh: refresh all or part of an X screen... xrefresh(1)
xstart: start up the sgi X server as a NeWS client... xstart(1)
xrdb: X server resource database utility... xrdb(1)
sm: a session manager for X... sm(1)
xtdcmap: X standard colormap utility... x cmap(1)
X Standards... x standards(1)

--114--

April 1990

Version 5.0
uwm: a window manager for X. ................. uwm(1)
converter. x10ox11: X version 10 to version 11 protocol x10ox11(1)
xdrp: dump an X window directly to a printer. xdrp(1)
xpr: print an X window dump. xpr(1)
draw interesting patterns in an X window. muncher: muncher(1)
paint some plaid-like patterns in an X window. plaid(1)
xscope: X Window Protocol Viewer. xscope(1)
xinit: X Window System initializer. xinit(1)
xlogo: X Window System logo. xlogo(1)
program.. xmessage: X window system message display xmessage(1)
x: X Window System server. xserv(1)
twm: Tab Window Manager for the X Window System. twm(1)
Manual page display program for the X Window System.. xman: xman(1)
xsgi: SGI Iris server for the X Window System. xsgi(1)
xwd: dump an image of an X window. xwd(1)
x: X Window System server. xserv(1)

xfbiff: mailbox flag for X. xfbiff(1)
xclock: analog / digital clock for X. xclock(1)
xcalc: scientific calculator for X. xcalc(1)
ex: X Window System display server x(1)
xload: load average display for X. xload(1)
server font list display for X. xfsfonts: xfsfonts(1)
server window list display for X. xswins: xswins(1)
utility for modifying keymaps in X. xkmmap: xkmmap(1)
xprop: property display for X. xprop(1)
xset: user preference utility for X. xset(1)

window parameter setting utility for X. xssetroot: root xssetroot(1)
xterm: terminal emulator for X. xterm(1)
window information utility for X. xwininfo: xwininfo(1)
xwud: image display for X. xwud(1)
protocol converter. x10ox11: X version 10 to version 11 x10ox11(1)
automated maze program... [ demo ] X11 ". maze: an maze(1)
BDF to SNF font compiler for X11. bdftosnf: bdftosnf(1)
& click interface for selecting X11 font names. xfonts: point xfonts(1)
calendar with a notebook for X11. xcalendar: xcalendar(1)
Generation of figures under X11. xfig: Facility for Interactive xfig(1)
xargs: construct argument list(s) xargs(1)
xauth: X authority file utility. xauth(1)
xfbiff: mailbox flag for X. xfbiff(1)
xclock: analog / digital clock for X. xclock(1)
xcalc: scientific calculator for X. xcalc(1)

for X11. xcalendar: calendar with a notebook xcalendar(1)
xclipboard: X clipboard client. xclipboard(1)
x: X Window System display server x(1)
xclock: analog / digital clock for X. xclock(1)
xcol: interchange between cut and xcol(1)
xditview: display ditoff DVI files. xditview(1)
xdm: X Display Manager. xdm(1)
buffer and selection. xdpr: dump an X window directly to a xdpr(1)
utility for X. xdpinfo: display information xdpinfo(1)
rename the existing hostname in yp hosts database.
remove the existing host entry in compress, uncompress,
zero: source of zeroes.
timezone: set default system time closeup:
zoom in on an image.

renamehost: rename(3N)
unregisterhost: unregisterhost(3N)
compress(1)
zero(7)
timezone(4)
closeup(6D)