AGENDA

MONDAY, SEPTEMBER 19, 1977

8:00  COFFEE AND ROLLS

8:30  WELCOME

9:00  INTRODUCTION AND INSTRUCTION OF THE BOARD  

9:30  HISTORY AND BUSINESS OVERVIEW OF THE XEROX PROGRAM  

10:45  LADC ORGANIZATION AND RESPONSIBILITIES

12:00  LUNCH

1:00  INTRODUCTION TO THE PDR  

CP-6 SOFTWARE FACTORY  

SHEL KLEE

BOB SMITH

HANK HAUGLAND

SHEL KLEE

DICK LITSCHGI

JOHN CATOZZI
AGENDA

TUESDAY, SEPTEMBER 20, 1977

8:30  CP-6 SYSTEM STRUCTURE OVERVIEW  DOUG HEYING
9:45  PROGRAM-TO-PROGRAM INTERFACE
      PROGRAM-TO-SYSTEM INTERFACE  DOUG HEYING
10:00 MEMORY MANAGEMENT AND JOB STEP CONTROL  JOHN COLLINS
11:00 CLIMB-PMME, DISPATCHING, FAULTS, EXCEPTIONS  LINDA AVERY
12:15 LUNCH
1:00  CONNECTIONS  SAM KEYS
2:00  FILE MANAGEMENT  PAT CRISMAN
3:30  USER-TO-SYSTEM INTERFACE  LARRY FELDMAN
4:30  T & D INTERFACE  DAVE MORGAN
4:50  MISCELLANEOUS INTERFACES  DAVE YOX
5:30  SUMMARY  DOUG HEYING
AGENDA

WEDNESDAY, SEPTEMBER 21, 1977

8:30 COMMUNICATIONS - INTRODUCTION
   L6 FACTORY
   COMMUNICATION SOFTWARE
   CHUCK MARTIN
   TERRY COX
   TOM MELTON

10:30 LOS ANGELES DEVELOPED PROCESSORS
   PL-6
   BASIC
   FORTRAN
   APL
   TEXT AND SUMMARY
   WING WONG
   RICH HANSON
   JIM GRIFFIN
   JOHN FLINT
   TOM MARTIN
   WING WONG

12:30 LUNCH

1:15 PHOENIX DEVELOPED PROCESSORS OVERVIEW
   COBOL 74; PL/I
   SORT MERGE
   ASSEMBLER
   GEORGE MANN
   DUANE DAVIS
   JOHN WERTZ
   GEORGE MANN

3:15 LOS ANGELES DEVELOPED PROCESSORS
   I-D-S/II, IDP/MANAGE
   RPG-II
   JOHN ROBERTSON
   DOUG CHESTER

4:00 TEST PLAN AND CONVERSION PLAN
   HERB GESHWIND

4:30 STAGING AND SCHEDULING -- THE WORK PLAN
   DICK LITSCHGI
AGENDA

THURSDAY, SEPTEMBER 22, 1977

8:30  LADC SUMMARY – RISKS AND ISSUES
      DICK LITSCHGI

9:30  GENERAL QUESTION PERIOD
      – BOARD DISCUSSION AND RISK DETERMINATION

12:00 LUNCH
      – MORE BOARD DISCUSSION
SOFTWARE FACTORY

- Requirements - What should it provide?
- History - How we got to where we are
- System implementation language - What is PL-6?
- Components of the Factory - Software to make software
- Use of the Factory - How it all works together
- Transport of the Factory to CP-6 base system
- Experience to date - How well does it work?
REQUIREMENTS OF SOFTWARE FACTORY

- Languages to suit needs of system implementation
- Tools to put programs together and onto target machine
- Debugging aids to help get code working sooner
- Base system with convenient, controlled access and file system
- Sufficient hardware to provide home for software factory
- Efficient environment to promote productivity and help retain programmers' sanity
A LITTLE HISTORY

- PL/1 – TOO MUCH OF A GOOD THING
- MULTICS – NICE TOOLS, WRONG MACHINE
- GCOS – ASSEMBLER, RIGHT MACHINE
- PL-6 – SMALL, CONTROLLABLE
- CP-V – FAMILIAR, AVAILABLE
- BMAP – GMAP ON CP-V
PL-6 THE SYSTEM IMPLEMENTATION LANGUAGE

- PL/1 - LIKE SYNTAX
- BLOCK STRUCTURED
- SIMPLE DATA TYPES
- MINIMAL RUN-TIME ROUTINES
- NO HIDDEN OVERHEAD
- INTERFACES TO SYSTEM SERVICES
- FACILITATES CODING IN NSA ENVIRONMENT
- USES CAPABILITIES OF L66 INSTRUCTION SET
HARDWARE

- DUAL 560 - PRIMARY FACTORY SYSTEM THROUGH 1Q79
- DUAL SIGMA6 - OVERFLOW/BACKUP SYSTEM FOR DUAL 560
- TWO L66's - FOR HANDS ON DEBUGGING THROUGH 3Q78
  - ONE BECOMES FACTORY SYSTEM 4Q78
- TERMINALS - IN OFFICES OF ALL PROGRAMMERS and A FEW TERMINAL ROOMS
- PHOENIX FACTORY SYSTEM - SIGMA.6 and L66 AVAILABLE
CP-V BASE SYSTEM

- CONTROLLED DEPENDABLE FILE SYSTEM
- EDIT - KNOWN ENTITY
- DEBUGGER - ALLOWS FOR FAST DEVELOPMENT OF BUILDING BLOCKS
- FORTRAN - MOST FACTORY SOFTWARE CODED IN FORTRAN
BUILDING BLOCKS OF THE FACTORY
(SOFTWARE TO BUILD SOFTWARE)

MINI-DELTA  L66 SIMULATOR
DEF       XDELTA
PL-6      LINK  BMAP

CP-V BASE SYSTEM
CP-6 OBJECT LANGUAGE

- DEVELOPED TO SPECIFICALLY FIT CP-6 ENVIRONMENT
- PROVIDES FOR STANDARD CALLING SEQUENCES
- ALLOWS IMPLEMENTATION OF A COMMON DEBUGGER FOR ALL LANGUAGES
- FACTORY FORMAT ACCOMODATES 36-BIT WORD SIZE WITHOUT ALTERING FILE RECORD LAYOUT (2 WORDS = 1 WORD)
PL-6

- Based on PL/H (based on control FORTRAN)
- Coded in FORTRAN
- Lives on CP-V system
- Produces CP-6 object units with debug schema
BMAP

- L66 ASSEMBLER (WITH NSA, EIS)
- CODED IN FORTRAN
- LIVES ON CP-V
- PRODUCES CP-6 OBJECT UNITS
- GMAP EQUIVALENT WITH EXTENSIONS
LINK

- LINKER WHICH CREATES PROGRAM RUN UNITS FROM OBJECT UNITS
- CREATES ALL EXECUTABLE PROGRAMS (BOOTSTRAP, MONITOR, USER PROGRAMS)
- SUPPORTS DEBUG SCHEMA
- CREATES LOAD MAP OF THE RUN UNIT
- CODED IN FORTRAN
- LIVES ON CP-V SYSTEM
- SUBSET OF CP-6 LINKER FUNCTIONALITY
L66 SIMULATOR

- Provides CP-6 / L66 simulation on CP-V
- Uses link built run units for input
- Includes:
  - L66 Instruction simulation (including EIS & NSA)
    - Interactive symbolic debugger
    - Minimal system services for I/O
- Allows checkout of code before system integration
- Coded in metasymbol (CP-V assembler)
DEF

- Creates the System Boot Tape (PO Tape)
- Bootstrap, Mini-Delta, Monitor, XDelta, Ghost1 Program Images extracted from Link Built Run Unit Files
- Monitor Debug Schema Sorted by FCG and Written to Tape
- System Files Written to Tape in CP-6 Tape Format (Run Unit Files & CP-V Files)
- Converts 32-Bit Word Formats to 36-Bit Format
- Coded in Metasymbol
MINI-DELTA

- Runs on L66 as standalone host debugging system
- Provides interactive debugging of CP-6 monitor at system console
- Reads PO tape, initializes MPC's, sets up monitor's running environment
- Debugger provides ➞ symbolic segment specification
  ➞ Absolute addressing within segment (allows symbol definition)
  ➞ Instruction breakpoints (up to seven)
  ➞ Memory dumps to console or printer
  ➞ Display and modification of memory and registers
- Reads input commands from system console, PO tape, or card reader
- Coded in BMAP

9/19/77 JRC
XDELTA

- Runs on L66 as standalone host debugging system
- Powerful interactive debugger - replaces Mini-Delta
- Utilizes PL-6 generated program schema
- Initial version interfaces with Mini-Delta
- Provides ➔ statement or instruction mode single step execution
  ➔ unlimited breakpoints (with conditions, attached commands)
  ➔ displays data, instructions in specifiable format
  ➔ performs disk, tape I/O to facilitate file system checkout
  ➔ allows display, modification of all memory and registers
- Coded in PL-6 (some BMAP)
- Availability scheduled for November 1977
MISCELLANEOUS TOOLS

- EXTRACT
- DECOMMENT
- DRAW
- XREF

▼ (LIST GROWS AS NEEDS ARISE)
THE FACTORY ON CP-V

CP-V ONLINE SYSTEM

EDIT

Controlled Files/Accts

PL-6

BMAP

LINK

DEF

PO

MINI-DELTA XDELTA

L66

DELTA L66 SIMULATOR

9/19/77 JRC
TRANSPORTING FACTORY TO CP-6

- All CP-6 Factory Building Blocks coded in either PL-6 or FORTRAN
- PL-6 Blocks are created on CP-V Factory
- CP-6 CodeGen for CP-V FORTRAN available June '78
  - Runs on CP-V
  - Produces CP-6 Object Units
- FORTRAN blocks are compiled with this modified FORTRAN then linked and transported just as the PL-6 blocks
BLOCKS TO BE RE-CODED IN PL-6

- LINK (FULL IMPLEMENTATION)
- DEF

BLOCKS TO BE CODED IN PL-6

- EDIT
- XDELTA
- DELTA
BLOCKS TO BE MOVED VIA FORTRAN VEHICLE

- PL-6 (MAIN REASON FOR CREATING VEHICLE)
- BMAP (???)
EXPERIENCE TO DATE
(Or How To Lick a Camel)

TWO MAJOR HUMPS —

1. LEARNING INTIMACIES OF NSA ADDRESSING
2. LEARNING TO CODE IN NEW (HIGHER-LEVEL) LANGUAGE
FACTORY TOOLS AIDED IMMENSELY

- MINI-DELTA – HANDS ON WITH L66 EASY TO SEE WHAT'S HAPPENING

- SIMULATOR – ABILITY TO SEE CODE WORK
WHERE ITS ALL GOTTEN US

- MINI-DELTA PATH (WITH L66)
  - BOOT PROCESS
  - MEMORY PROCESS
  - SCHEDULER
  - FAULT HANDLER
  - PMME HANDLER (SERVICE DECODER)
  - IOQ-10S
- SIMULATOR PATH
  - FILE MANAGEMENT
  - SERVICE PROCESSORS
- PL-6 — HOW HAS IT HELPED

ON SCHEDULE AND MOVING FORWARD
CP - 6

- SYSTEM STRUCTURE
- PROGRAM to PROGRAM INTERFACE
- USER to SYSTEM INTERFACE
- PROGRAM to SYSTEM INTERFACE
- INITIALIZATION, RMA, PERFORMANCE
CP-6 IS A SYSTEM

- STRONG INTERFACES
- MONITOR, PROCESSORS WORK TOGETHER
- FUNCTIONS NOT DUPLICATED (EFFORT or STORAGE)
- ONENESS OF THE SYSTEM
- FULL PROTECTION
- PL-6 RELATIONSHIP TO SYSTEM
STRENGTH OF CP-6 IS IN ITS INTERFACES

- USER STRUCTURE
- SYSTEM SERVICE INTERFACE
- PROGRAM BINDING INTERFACE
- PROGRAM CALLING INTERFACE
- USER to SYSTEM INTERFACE
SYSTEM SERVICE INTERFACE

- UNIFORM ABSTRACT INTERFACE
- ISOLATES USER PROGRAMS FROM HARDWARE/SYSTEM SOFTWARE
- ENABLES COMPLETE DEVICE INDEPENDENCE
- MAKES FILES INTERCHANGEABLE AMONG LANGUAGES
- CLIMB is EXCELLENT VEHICLE
USER STRUCTURE (DOMAINS)

- USER DOMAIN
- ASL DOMAIN
- COMMAND PROCESSOR DOMAIN
- DEBUGGER DOMAIN
- PLUS MONITOR DOMAIN via PMME
USER DOMAIN

- JOB INFORMATION TABLE (JIT)
- READ ONLY SEGMENT (ROS)
  - Data Control Blocks (DCB)
  - Task Control Block (TCB)
- INSTRUCTION SEGMENT (IS)
- DYNAMIC SEGMENT (DS1 - DS8)
INSTRUCTION SEGMENT

- STATIC DATA
- PROCEDURE and CONSTANTS
- DYNAMIC DATA
- SHARED RUNTIME LIBRARY
USER VIRTUAL SPACE

- 'THE USER' FROM STANDPOINT of MONITOR CONTROL
- CONTAINS ALL 'GLUE' HOLDING IT TOGETHER
  - Page Table
  - Linkage Segments
  - Safe Store Stack
  - Argument/Parameter Segments
- COLLECTS ALL PHYSICAL MEMORY ALLOCATED TO THIS USER
- LOCATES BUT DOES NOT CONTAIN SHARED PROCEDURE
  - Shared Processor
  - Runtime Library
USER VIRTUAL ADDRESS SPACE

PAGE TABLE

USER LS
NULL
R
NULL
R
IS: R/W/E
R/W
R/W
NULL

MONITOR LS
R/W
R/W
R/W
R/W
IS: R/W/E
R/W
R/W
R/W

LIBRARY
BOUND DATA
PROCEDURE: W
DYNAMIC DATA
UNUSED
LIB PROCEDURE: W
DYNAMIC SEGMENTS

0
128
224K
352
384
511

TO
MON
WSQ
CP-6 MEMORY UTILIZATION

- CP-V based on high speed swapper and new inventions not tolerable
- Rotating memory and CCD's considered
- Evaluation of RAM cost trend led to November '76 decision
  - Use 16K RAM in lieu of swapper
  - Decision irrevocable without schedule impact
- Current status shows no significant changes
  - Much too late for any swapper development
  - Projected memory requirements remain the same
  - 4K chip cost trend better than target (but not a solution)
  - But will volumes develop for 16K chips

9/19/77
SYSTEM VIRTUAL ADDRESS SPACE

IDS-II
DELTA
COMMAND PROCESSOR
MONITOR
SPECIAL SYSTEM TABLES

USER 1
USER 2
USER 3
USER 4
USER 500
MONITOR DOMAIN

- USER JIT, ROS, HJIT, FILE BUFFERS, T STACK, PAGE TABLE
- USER PARAMETER SEGMENT
- MONITOR IS
- MONITOR DS's
- REAL MEMORY
MULTIPROCESSING

- USERS RUN STRICTLY in USER CONTEXT - ANY CPU
- MOST MONITOR SERVICES RUN in USER CONTEXT - ANY CPU
- FAULT HANDLER RUNS in USER/CPU CONTEXT - ANY CPU
- SCHEDULER/PHYSICAL I/O+MONITOR SERVICES WHICH USE CPU GLOBAL DATA RUN ON MASTER ONLY
MONITOR MEMORY USAGE

- CPU SPECIFIC (UNIQUE COPIES PER CPU)
  - MONITOR JIT and HJIT
  - Page Table Directory, Page Table, TSTACK
  - STATIC DATA Protected by SOFT DISABLE

- CPU GLOBAL (SAME COPY USED BY ALL CPU's)
  - STATIC DATA
  - REAL (Allocated At Boot Time)
  - DYNAMIC SEGMENTS
  - DYNAMIC REAL (Will be in some Users PT for REF)
  - PROCEDURE and CONSTANTS
ONENESS OF THE SYSTEM

- ONE KIND of JOB
- ONE CPU SCHEDULER
- ONE COMMAND LANGUAGE, DEBUGGER
- ONE FILE MANAGEMENT SYSTEM
PL-6 RELATIONSHIP TO SYSTEM

- LANGUAGE BUILT TO FIT SYSTEM, NOT VICE VERSA
  - Operating System is the Runtime 'Library'

- FACILITIES INCLUDED TO FACILITATE
  - Building the system
  - Using the system

- DOES NOT PROVIDE HIDDEN CONTROL MECHANISMS

- SYSTEM IS NOT PREJUDICED TO ANY LANGUAGE
  - But Provides Facilities Necessary to Implement All
CP-6 IS A SYSTEM

- STRONG INTERFACES
- MONITOR, PROCESSORS WORK TOGETHER
- FUNCTIONS NOT DUPLICATED (EFFORT or STORAGE)
- ONENESS OF THE SYSTEM
- FULL PROTECTION
- PL-6 RELATIONSHIP TO SYSTEM
STRENGTH OF CP-6 IS IN ITS INTERFACES

• USER STRUCTURE
• SYSTEM SERVICE INTERFACE
• PROGRAM BINDING INTERFACE
• PROGRAM CALLING INTERFACE
• USER to SYSTEM INTERFACE
PROGRAM BINDING INTERFACE

- OBJECT LANGUAGE COMMON FOR ALL LANGUAGES
- DEBUG SCHEMA INCLUDED PERMITTING COMMON DEBUGGER
- ORIENTED TO SHARED PROCEDURE ENVIRONMENT
- PERMITS GENERAL LINK TIME BINDING
FEATURES OF OBJECT LANGUAGE/LINKER

- GENERAL RELOCATION of FIELDS
- COMPLETE DESCRIPTION of VARIABLES/PROCEDURES
- DETECTION of PROCEDURE DEFINITION/CALL MISMATCH
- SYSTEMIC DEFINITIONS SUPPLIED by LINKER
- PERMITS GENERAL LINK TIME BINDING to PROMOTE MODULAR PROGRAMMING
- DESIGNED FOR EASE of GENERATION AND SPEED of LINKING
- LINKER PRODUCES RUN UNIT
FEATURES OF RUN UNIT

- SAME FORMAT USED FOR ALL TYPES PROGRAM
- EXECUTABLE FORM CONTAINING STATIC, PROCEDURE, DCB's and TCB
- IDENTIFIES REQUIRED LIBRARY and ASL
- STANDARD FILE, of course
PROGRAM CALLING INTERFACE

- STANDARD SYSTEM CALLING SEQUENCE
- ACCOMODATES NEEDS of ALL LANGUAGES
- FACILITATES MIXED LANGUAGE PROGRAMS
- DESIGNED FOR EFFICIENT FORMAL INTERFACE
  Among PROGRAMS In a RUN UNIT
- PROMOTES COMMON LIBRARY ROUTINES
CALLING SEQUENCE ATTRIBUTES

- DESIGNED for NSA ENVIRONMENT
- ORIENTED TO PURE PROCEDURE ENVIRONMENT
- CONTAINS INFORMATION USEFUL to DEBUGGER
- INTEGRATED WITH PL/1, PL-6 STACK FRAME MANAGEMENT
- ENCOMPASSES LIBRARY FUNCTION CALL FORMAT
CP-6 MEMORY MANAGEMENT

- PURPOSE
- ORGANIZATION
- OVERVIEW OF VIRTUAL MEMORY STRUCTURE
- DOMAINS – INTERRELATIONSHIP OF ADDRESS SPACES
- INTERNAL FUNCTIONS
- PROGRAM INTERFACE
MEMORY MANAGEMENT IS BUILT IN LEVELS

- PHYSICAL PAGE ALLOCATION
- PAGE TABLE MANIPULATION
- WSQ VIRTUAL PAGE ALLOCATION
- LINKAGE SEGMENT MANIPULATION
- SEGMENT — RELATIVE PAGE ALLOCATION
- ABSTRACTIONS BUILT ON ABOVE CAPABILITIES:
  - DATA SEGMENT ALLOCATION
  - "DYNAMIC" DATA ALLOCATION
  - FILE MANAGEMENT BUFFER ALLOCATION
<table>
<thead>
<tr>
<th>JIT, TSTACK, ACCOUNTING</th>
</tr>
</thead>
<tbody>
<tr>
<td>HJIT, LINKAGE SEGMENTS, SAFESTORE STACK, PARAMETER STACK</td>
</tr>
<tr>
<td>FILE (&amp; COOP) BUFFERS</td>
</tr>
<tr>
<td>TCB, ECCB, TREE, DCBs</td>
</tr>
<tr>
<td>LIBRARY AND USER DATA</td>
</tr>
<tr>
<td>USER PROCEDURE</td>
</tr>
<tr>
<td>INSTRUCTION SEGMENT (256K)</td>
</tr>
<tr>
<td>DYNAMIC DATA</td>
</tr>
<tr>
<td>SHARED RUN-TIME LIBRARY</td>
</tr>
<tr>
<td>AUTOMATIC DATA, COMMON DATA, USER DATA SEGMENTS</td>
</tr>
<tr>
<td>SPECIAL SHARED PROCESSOR DATA SEGMENTS</td>
</tr>
</tbody>
</table>
# User and Monitor Virtual Memory Layout

## User WSQ

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JIT, TSTACK, ACCOUNTING</td>
<td></td>
</tr>
<tr>
<td>HJIT, LINKAGE SEGMENTS, SAFESTORE STACK, PARAMETER STACK</td>
<td></td>
</tr>
<tr>
<td>FILE (&amp; COOP) BUFFERS</td>
<td></td>
</tr>
<tr>
<td>TCB, ECCB, TREE, DCBs</td>
<td></td>
</tr>
<tr>
<td>Library and User Data</td>
<td></td>
</tr>
<tr>
<td>User Procedure</td>
<td>Instruction Segment (256K)</td>
</tr>
<tr>
<td>Dynamic Data</td>
<td></td>
</tr>
<tr>
<td>Shared Run-Time Library</td>
<td></td>
</tr>
<tr>
<td>Automatic Data, Common Data, User Data Segments</td>
<td></td>
</tr>
<tr>
<td>Special Shared Processor Data Segments</td>
<td></td>
</tr>
</tbody>
</table>

## Monitor WSQ

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JIT, TSTACK, ACCOUNTING</td>
<td></td>
</tr>
<tr>
<td>HJIT, LINKAGE SEGMENT, SAFESTORE STACK, PARAMETER STACK</td>
<td></td>
</tr>
<tr>
<td>Monitor &quot;Window&quot; Area</td>
<td></td>
</tr>
<tr>
<td>Monitor Initialization - Time Tables</td>
<td></td>
</tr>
<tr>
<td>Monitor Data</td>
<td>Instruction Segment (256K)</td>
</tr>
<tr>
<td>Monitor Procedure</td>
<td>(All pages marked housekeeping)</td>
</tr>
<tr>
<td>Monitor Data Segments</td>
<td></td>
</tr>
<tr>
<td>Reserved Space (64 pages)</td>
<td></td>
</tr>
</tbody>
</table>

9/19/77  JC
CP-6 DOMAINS OF REFERENCE

USER'S WORKING SPACE

Debugger's Domain

Command Processor Domain

Alternate Shared Library Domain

Monitor Domain

DELTA

IBEX

I-D-S/I

Monitor Domain Arguments

Monitor

CP Data

Data Segments

User Program Data

User Program

Runtime Library Program

ASL Data

Buffers

DCB
MEMORY MANAGEMENT INTERNAL FUNCTIONS

- GET and RELEASE PHYSICAL PAGE
- GET and RELEASE STOLEN PHYSICAL PAGE
- GET and RELEASE I/O CACHE PAGE

- MAP PHYSICAL PAGE TO VIRTUAL PAGE
- SET ACCESS CONTROL FLAGS FOR VIRTUAL PAGE
- SET SOFTWARE CONTROL FLAGS FOR VIRTUAL PAGE

- GET and RELEASE VIRTUAL PAGE
- CHANGE VIRTUAL MAP
- SET MEMORY PROTECTION
- EXCHANGE I/O CACHE PAGE FOR ONE CURRENTLY IN PAGE TABLE
MORE MM INTERNAL FUNCTIONS

- READ and MODIFY LINKAGE SEGMENT DESCRIPTORS
- GET and RELEASE SEGMENT - RELATIVE PAGES
- GET and RELEASE DATA SEGMENT SPACE
- GET and RELEASE DYNAMIC DATA PAGES
- GET DYNAMIC DATA LIMITS
- GET A FILE MANAGEMENT BUFFER
PROGRAM INTERFACE TO MEMORY MANAGEMENT

- M$GDS - GET DATA SEGMENT SPACE
- M$FDS - FREE DATA SEGMENT SPACE
- M$GDP - GET DYNAMIC DATA PAGES
- M$FDP - FREE DYNAMIC DATA PAGES
- M$GDDL - GET DYNAMIC DATA LIMITS
- M$GVP - GET VIRTUAL PAGE
- M$FVP - FREE VIRTUAL PAGE
- M$STLPP - STEAL PHYSICAL PAGE
- M$RSPP - RELEASE STOLEN PAGE
- M$CVM - CHANGE VIRTUAL MAP
- M$SMRPT - SET MEMORY PROTECTION
- M$SSC - SET SOFTWARE CONTROL FLAGS

9/19/77 JC
CP-6  EXECUTION CONTROLS

- CONCEPTS - USERS, PROGRAMS, JOB STEPS, COMMAND PROCESSORS
- PURPOSE
- BASIC FUNCTIONS
- PROGRAM INTERFACE
- EXAMPLE OF JOB STEP CYCLE
CP-6 JOB STEP CONCEPTS

- A USER IS NOT A PROGRAM
- THE INITIAL USER CONSISTS OF A
  - JIT
  - HJIT
  - INITIAL READ-ONLY SEGMENT
- JOB STEP EXTENDS FROM PROGRAM INITIATION TO RUNDOWN
- COMMAND PROCESSOR ALLOWS SPECIFICATION OF JOB STEPS
- ALL USERS HANDLED BY SAME EXECUTION CONTROL ROUTINES
PURPOSE OF EXECUTION CONTROL

- JOB STEP INITIATION
- JOB STEP TERMINATION
- INTRA-JOB STEP FUNCTIONS
  - OVERLAYS
  - LOAD-AND-LINK
  - ASSOCIATE SHARED LIBRARIES, ASL, OR DEBUGGER
EXECUTION CONTROL BASIC FUNCTIONS

- INITIATE OR CONTINUE A JOB STEP
- ASSOCIATE COMMAND PROCESSOR
- ASSOCIATE A SPECIFIED PROGRAM/PROCESSOR
- TERMINATE A PROGRAM-EXIT, ERROR, ABORT
- RUNDOWN A USER - TERMINATE A JOB STEP
- LOAD-AND-LINK TO ANOTHER PROGRAM
- LOAD AN OVERLAY FROM A RUN UNIT
- PROVIDE EXIT CONTROL DISPATCHING
SHARING PROCEDURE

- SHARABILITY DETERMINED AT JOB STEP INITIATION
- SHARED PROCESSOR INITIAL DATA, READ ONLY SEGMENT ALLOCATED TO INDIVIDUAL USER's WSQ
- SHARED PROCESSOR PROCEDURE MAPPED INTO EVERY USER's WSQ
- UNSHARED PROGRAMS READ ENTIRELY INTO MEMORY ALLOCATED TO INDIVIDUAL WSQ's
PROGRAM INTERFACE

- M$CPEXIT - INITIATES NEW JOB STEPS
- M$RUND - TERMINATES A JOB STEP - RUNS DOWN A USER
- M$EXIT - PROGRAM NORMAL EXIT
- M$ERR - PROGRAM ERROR EXIT
- M$XXX - PROGRAM ABORT EXIT
- M$OLAY - BRINGS IN A PROGRAM OVERLAY
- M$LDTRC - TRANSFERS CONTROL TO ANOTHER PROGRAM
- M$LINK - TRANSFERS CONTROL TO ANOTHER PROGRAM, SAVING THE CURRENT PROGRAM STATE FOR LATER RESUMPTION
SAMPLE JOB STEP CYCLE

- NEW USER CREATED and LOGGED ON
- SCHEDULER CALLS CPEXIT TO ASSOCIATE COMMAND PROCESSOR
- COMMAND PROCESSOR ENTERED
- CP is REQUESTED TO RUN A PROGRAM
- CP ISSUES M$CPEXIT TO INITIATE PROGRAM
- EXECUTION CONTROL FETCHES PROGRAM INTO WSQ, CALLS SCHEDULER
- PROGRAM EXECUTES and EVENTUALLY EXITS
- CP IS REENTERED
- CP IS REQUESTED TO RUN ANOTHER PROGRAM
- CP TERMINATES CURRENT JOB STEP VIA M$RUND, INITIATES ANOTHER VIA M$CPEXIT.
PROGRAM TO SYSTEM INTERFACE

- CONTROL PATHS BETWEEN CP-6 DOMAINS
- MONITOR SERVICES INTERFACE
- SCHEDULING (DISPATCHING)
- FAULT and EXCEPTIONAL CONDITION HANDLING
INTER-DOMAIN INTERFACE

• DOMAINS HAVE ESTABLISHED LEVEL OF PRIORITY
  - USER
  - ALTERNATE SHARED LIBRARY
  - DEBUGGER
  - COMMAND PROCESSOR
  - MONITOR

• INWARD CLIMB TO DOMAIN OF HIGHER PRIORITY
  - SAVES ENVIRONMENT
  - OPTIONALLY PREPARES NEW PARAMETERS
  - ESTABLISHES NEW ENVIRONMENT

• OUTWARD CLIMB TO RETURN
  - RESTORES ENVIRONMENT
  - RELINQUISHES CONTROL
CONTROL PATHS BETWEEN CP-6 DOMAINS
CP-6 MONITOR SERVICES INTERFACE

- SERVICES AVAILABLE IN ALL LANGUAGES
- SERVICES AVAILABLE TO ALL DOMAINS
- OPERATION IS IDENTICAL FOR BATCH and ON-LINE
- BROAD SPECTRUM of SERVICES AVAILABLE
<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB STEP CONTROL</td>
<td>15</td>
</tr>
<tr>
<td>RUN-TIME SERVICES</td>
<td>25</td>
</tr>
<tr>
<td>MEMORY MANAGEMENT</td>
<td>15</td>
</tr>
<tr>
<td>FILE MANAGEMENT/DEVICE I/O</td>
<td>25</td>
</tr>
<tr>
<td>ON-LINE TERMINAL CONTROL</td>
<td>10</td>
</tr>
<tr>
<td>DIAGNOSTIC SERVICES</td>
<td>10</td>
</tr>
<tr>
<td>SPECIAL SHARED PROCESSOR SERVICES</td>
<td>5</td>
</tr>
</tbody>
</table>
CP-6 MONITOR SERVICES REQUEST

- INVOKED VIA PMME FORM of CLIMB
  - SAVE ENVIRONMENT
  - VALIDATES PARAMETER ADDRESSES/SIZE
  - ESTABLISHES MONITOR ENVIRONMENT

- INPUTS ASSOCIATED with EACH REQUEST
  - SERVICE REQUEST CODE
  - ERROR RETURN SPECIFICATION
  - WHERE REQUIRED - USER PARAMETERS

- OUTPUTS SUPPLIED for ERROR ROUTINES
  - COPY of SAFE STORE FRAME
  - SERVICE REQUEST CODE
  - DCB# (IF APPLICABLE)
  - ERROR CODE
Monitor's Parameter Stack

- $P_0$ Descriptor
- $P_1$ Descriptor
- $P_n$ Descriptor

Index Register 0

<table>
<thead>
<tr>
<th>ERR</th>
<th>SERVICE REQUEST CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

User's WSQ

- Parameter 1
- FPT Values
- Parameter $n$
FUNCTIONAL PARAMETER TABLE

- FIXED FORMAT for EACH REQUEST
- SUPPLIES MONITOR with USER SPECIFIC INFORMATION
  - AREAS in USER's MEMORY
  - PRESENCE/ABSENCE of OPTIONAL PARAMETERS
  - VALUES to OVER-RIDE DEFAULTS
- AREAS IN USER MEMORY SPECIFIED AS VECTORS
- ALL VALUES PASSED AS ONE PARAMETER

VALUE BLOCK VECTOR

P₁ VECTOR

Pₙ VECTOR

VALUE BLOCK

BASE

BOUND
LDPO    FPT

CLIMB    EA - BIT 0 = ERR RETURN SPECIFICATION
         BITS 0-17 = SERVICE REQUEST CODE
SEGRID   = PMME
C        = INWARD CLIMB
         LOAD XO WITH EA
E        = 1 - PREPARE PARAMETERS (OR ZERO)
P        = # of PARAMETERS - 1

TRA ERROR  (if EA BIT 0 = 1)

USER SIDE OF PMME
PL-6 EXAMPLE

DCL INBUF CHAR (80) STATIC;

% INCLUDE CP-6;

% FPT - READ (FPTN = SIFPT,
   BUF = INBUF,
   DCB = M$SI);

CALL M$READ (SIFPT) ALTRET (READERR);
   
READERR: /*ERROR HANDLER*/
CP-6 SCHEDULER

• STATE QUEUE MAINTENANCE
• QUANTUM TIMING and CONTROL
• EXECUTION SCHEDULING
• ASYNCHRONOUS EVENT PROCESSING
STATE QUEUES

- EXECUTING

- EXECUTABLE
  PRIORITY 1
  PRIORITY 2
  PRIORITY 3
  ...
  ...
  ...
  PRIORITY n

- NON-EXECUTABLE
  I/O WAIT
  SLEEPING
  QUEUED FOR RESOURCE
  TERMINAL INPUTTING
  TERMINAL OUTPUTTING
STATE QUEUE MAINTENANCE

- EVERY JOB in SYSTEM HAS ASSIGNED PRIORITY
- EVERY JOB HAS SINGLE ENTRY in STATE QUEUES
- EXECUTABLE STATE QUEUES HAVE a PRIORITY
- 'EVENTS' CAUSE CHANGE in STATE
  - BLOCKING EVENT MOVES USER to NON-EXECUTABLE STATE
  - UNBLOCKING EVENT MOVES USER TO EXECUTABLE STATE
    BASED ON PRIORITY ASSOCIATED WITH THE EVENT.
  - EXECUTION SCHEDULER MOVES HIGHEST PRIORITY
    EXECUTABLE USER TO EXECUTING STATE
EVENTS

I/O IN PROGRESS
I/O COMPLETE

SLEEP
WAKE-UP

NEED MEMORY PAGE
PAGE AVAILABLE

NO DISK SPACE
DISC SPACE AVAILABLE

COMMUNICATIONS INPUT IN PROCESS
COMMUNICATIONS INPUT COMPLETE

BREAK CHARACTER RECEIVED
CONTROL Y RECEIVED
USER ABORTED BY OPERATOR
I/O COMPLETE with EVENT INFO
QUANTUM TIMING

- SYSTEM PARAMETERS

QUAN - MAXIMUM TIME ALLOWED COMPUTE - BOUND USER

- USED TO CAUSE TIMER RUNOUT FAULT at QUANTUM END

QMIN - AMOUNT OF TIME GUARANTEED TO USER

- USER MAY BE FORCED to RELINQUISH CONTROL to HIGHER PRIORITY USER ONCE QMIN IS REACHED
EXECUTION SCHEDULER

- ENTERED FOLLOWING CLIMB FOR: FAULT INTERRUPT PMME

- DETERMINE IF JOB ALLOWED TO CONTINUE
  - SUSPEND USER WITH ENVIRONMENT SAVED in SAFE-STORE
  - RELOAD SSR FOR NEW USER

- PROCESS ASYNCHRONOUS EVENTS
  - ENTER COMMAND PROCESSOR
  - ENTER DEBUGGER
  - ENTER USER ASYNCHRONOUS EVENT HANDLER

- RET THROUGH FRAME LAI D DOWN BY: FAULT INTERRUPT PMME
CP-6  FAULT  PROCESSING

ACTION TAKEN DEPENDS ON:

- TYPE OF FAULT
  ARITHMETIC
  PROGRAMMED
  PROCEDURE ERRORS
  SYSTEM

- DOMAIN IN CONTROL

- REQUEST FOR CONTROL
CLASSES OF FAULTS

ARITHMETIC
OVERFLOW
DIVIDE CHECK

PROGRAMMED
MASTER MODE ENTRY
DERAIL
FAULT TAG

PROCEDURE ERRORS
MEMORY
COMMAND
LOCKUP
ILLEGAL PROCEDURE
MISSING SEGMENT
MISSING PAGE
SECURITY 2
SAFE STORE
SECURITY 1

SYSTEM
TIMER, RUN-OUT
PARITY
OP NOT COMPLETE
CONNECT
DYNAMIC LINK
MISSING WORKING SPACE
EXECUTE
START UP
SHUT DOWN

9/19/77 LLA
SYSTEM FAULTS

TIME RUN-OUT - CALL SCHEDULE
PARITY
OP NOT COMPLETE - CALL T&D
CONNECT - PERFORM COMMANDS IN MAILBOX

DYNAMIC LINK
MISSING WORKING SPACE
EXECUTE
STARTUP
SHUTDOWN

ALL OTHERS
USER IN CONTROL - ABORT OR ENTER USER TRAP HANDLER
MONITOR IN CONTROL - CALL RECOVERY
EXCEPTIONAL CONDITION PROCESSING

- TYPES OF EXCEPTIONAL CONDITIONS
- ESTABLISHING CONTROL
- EXCEPTIONAL CONDITION ENVIRONMENT
- ENTERING EXCEPTIONAL CONDITION Routines
- EXIT FROM EXCEPTIONAL CONDITION Routines
TYPES OF EXCEPTIONAL CONDITIONS

• FAULTS
• PMME ERRORS
• OPERATOR ABORTS
• LIMITS EXCEEDED
• PROGRAM EXITS AND ABORTS
• TIMER RUN-OUT
• EVENT COMPLETION
• BREAK KEY INTERRUPT
ESTABLISHING CONTROL

- PROGRAM MAY REQUEST CONTROL FOR ANY OR ALL
  - M$STIMER  TIMER RUN-OUT
  - M$INT  CONSOLE INTERRUPT
  - M$EVENT  COMPLETION OF SPECIFIED EVENT
  - M$TRAP  MACHINE TRAPS – 3 CLASSES
  - PMME ERRORS/NO ERR RETURN SPECIFIED
  - M$XCON  NORMAL OR ABNORMAL EXIT OF CURRENT PROGRAM
  - MONITOR SERVICES ERROR RETURN SPECIFICATION

- SEPERATE LEVEL of CONTROL for EACH DOMAIN
EXCEPTIONAL CONDITION CONTROL BLOCK (ECCB)

- CONTAINS USER SPECIFIED HANDLER ADDRESSES
- SEPARATE ECCB FOR EACH DOMAIN

<table>
<thead>
<tr>
<th>Stimer</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break</td>
<td>XCON</td>
</tr>
<tr>
<td>PMME</td>
<td>Arithmetic</td>
</tr>
<tr>
<td>Programmed</td>
<td>Error</td>
</tr>
</tbody>
</table>

CONTROL FLAGS
**Exceptional Condition Environment**

- **ECC**
  - 0 - Timer Runout
  - 1 - Event Completion
  - 2 - Break
  - 3 - XCON
  - 4 - Bad PMME/No ERRARTN
  - 5 - Arithmetic Fault
  - 6 - Programmed Fault
  - 7 - Procedure Fault
  - 99 - PMME Error Return
TASK CONTROL BLOCK

- STORAGE FOR RELEVANT INFORMATION
- SINGLE FRAME FOR PMME ERROR RETURNS
- STACKED FRAMES FOR ALL OTHER CONDITIONS
- SEPARATE TCB FOR EACH DOMAIN

CONTROL WORDS
PMME ERRORS
EXCEPTIONAL CONDITION STACK

<table>
<thead>
<tr>
<th>PTR TO PMME FRAME</th>
<th>PTR TO CURRENT EXC. COND. FRAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIZE OF UNUSED EXC. COND. STACK</td>
<td>SIZE OF CURRENT EXC. COND. FRAME</td>
</tr>
</tbody>
</table>

PMME ENVIRONMENT

EXCEPTIONAL CONDITION FRAME 1

EXCEPTIONAL CONDITION FRAME 2

EXCEPTIONAL CONDITION FRAME n

SIZE OF FRAME n-1 | ECC

UNUSED STACK SPACE

9/19/77 LLA
ENTERING EXCEPTIONAL CONDITION HANDLERS

- SYNCHRONOUS EVENTS

  DOES NOT CAUSE CHANGE OF DOMAIN
  
  SAFE - STORE COPIED TO TCB
  IC IN SAFE-STORE SET TO CONDITION HANDLER
  RET CAUSES HANDLER TO BE ENTERED

- ASYNCHRONOUS EVENTS

  MAY CAUSE CHANGE OF DOMAIN
  — CONDITION FOR EXECUTING DOMAIN
    SAME AS FOR SYNCHRONOUS EVENTS
  — CONDITION FOR DOMAIN WITH HIGHER PRIORITY
    LTRAD TO CONDITION HANDLER
  — CONDITION FOR DOMAIN OF LOWER PRIORITY
    DEFER PROCESSING
EXITING EXCEPTIONAL CONDITION ROUTINE

- MONITOR SERVICE ERROR RETURNS
  -- NO ACTION is REQUIRED
  -- M$MERC MONITOR ERROR HANDLING
  -- M$RETRY MODIFY ENVIRONMENT and RETRY PMME

- STACK CONDITION ROUTINES
  -- M$TRTN MODIFY ENVIRONMENT and RETURN
  -- M$CLRSTK POP STACK FRAME and CONTINUE
CONNECTING USERS TO DEVICES AND FILES

- TO CENTRAL SITE PERIPHERALS
- TO DISK FILES
- TO PVT. VOL. FILES
  - BOTH DEVICE AND FILE
- TO COMMUNICATIONS DEVICES
  - THESE MUST ALSO BE CONNECTED THEMSELVES
THE DATA CONTROL BLOCK (DCB)

- READ ONLY USER CONTEXT
- EACH USER HAS HIS OWN
- CONNECTION POINT FOR USERS TO DEVICE
- BASIC OPERATIONS: OPEN, READ, WRITE, CONTROL, CLOSE
- OPEN IS THE CONNECTING PROCESS
TYPES OF ACCESS

- PUBLIC FILE
- PRIVATE DISK FILE
- MANAGED TAPE FILE
- VIRTUAL DEVICE
- WSN/DEVICE
- MASTER T.S. TERMINAL
- COMGROUP
OPEN

- BASIC ELEMENTS SPECIFIED
- ASN: FILE, MANAGED TAPE, DEVICE, COMGROUP, ME
- WSN: SUBSET OF DEVICES
- RN: TYPE OF DEVICE/VIRTUAL DEVICE/OPLABEL
- SER: TAPE OR PRIVATE DISK VOLUME
- N.A.P: IDENTIFIES FILE
- SOME OR ALL MAY BE SPECIFIED ON DCB CREATION OR ISET
STANDARD FILE IDENTIFIER (FID)

RN# SER/N.A.P. @WSN

- ALL FIELDS OPTIONAL
- # SAYS DEVICE
- @ SAYS DEVICE
- / SAYS FILE
- . SAYS FILE
- RESERVED RN’S: MT, LT, DP,
  DC, CG, OPLABELS
FID EXAMPLES

PUBLIC FILE : A.B.C
PRIVATE DISK FILE : DP#1234/A.B.C
MANAGED TAPE FILE : LT#1234/A.B.C
VIRTUAL DEVICE : MTO1#123
WSN/DEV : LP@BOSTON
OPLABEL : LP
COMGROUP : CG#OSNET
MASTER TERMINAL : ME

HISTORICAL PROBLEM:
IS 'ZZ' DEVICE OR FILE?

9/19/77 SK
FUNCTIONS BETWEEN USER AND DEVICES

- OPLABELS
- WORKSTATIONS
- SYMBIONTS (SPOOLING)
- STREAMS
- RESOURCE MANAGEMENT
- FILE MANAGEMENT
  - CONVERTS FILE COMMANDS TO SENSIBLE I/O TO REAL DEVICE
OPLABELS

- BATCH, ONLINE, GHOST

<table>
<thead>
<tr>
<th>OPLABEL</th>
<th>BATCH</th>
<th>ONLINE</th>
<th>GHOST</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td>LP01, CR01</td>
<td>UC</td>
<td>OC</td>
</tr>
<tr>
<td>SI</td>
<td>CR01</td>
<td>UC</td>
<td>OC</td>
</tr>
<tr>
<td>LO</td>
<td>LP01</td>
<td>UC</td>
<td>OC</td>
</tr>
<tr>
<td>LP</td>
<td>LP01</td>
<td>LP01</td>
<td>LP01</td>
</tr>
</tbody>
</table>

- ALLOW STANDARD ASSIGNMENT TO "NATURAL" DEVICE FOR MODE
- HEAVILY USED FOR LP, CP, UC IN CP-V AND CP-6 BY PROCESSORS

9/19/77 SK
WORKSTATIONS

- EVERY USER HAS WORKSTATION OF ORIGIN
- WSN IS SUBGROUP OF ALL DEVICES
- DEVICE TYPE TRANSLATED THROUGH WSN DEFINITION TO PRODUCE REAL DEVICE AND "SYMBOIONTNESS"

WSN DEFINITION

LP: SYMBOIONT, OTHER ATTRIBUTES
   LP01 ON TERMINAL BOSTON
   LP02 ON TERMINAL BOSTON

QQ: NON-SYMBOIONT, OTHER ATTRIBUTES
   ZZ01 ON TERMINAL BIGBOSS

MT: USE CENTRAL SITE DEVICES
SYMBIONTS

- PROVIDES FILE BUFFERING FOR SLOW U.R. DEVICES
- MANY USERS USING FEW DEVICES
- TALK TO COMMUNICATION DEVICES NOT CONNECTED
- SENSIBLE OUTPUT SCHEDULING
  - FORMS
  - SIZE
  - PRIORITY
- BATCH QUEUE
- NON-CONTROL INPUT FILES
STREAMS

- ALLOW SEVERAL DCBS TO MIX OUTPUT
  - DIFFERENT OR SAME M$LO, M$DO
- STREAM IS VIRTUAL DEVICE NAMED VIA LDEV
- LDEV LP02, LPC, ATTRIBUTES
- STREAM DCB IS ASSIGNED TO WSN/DEV
- LP01 IS ALWAYS AUTOMATICALLY THERE - POINTS TO LP@ WSN OF ORIGIN
- ALWAYS SYMBIONT IN 1ST RLSE
RESOURCE MANAGEMENT

- RESOURCE IS VIRTUAL DEVICE NAMED VIA LIMIT MT01, MT, OTHER ATTRIBUTES
- ACTUAL DEVICE IS IN WSN/DEV FORM
- ALWAYS NON-SYMBIONT
- LIMIT PREVENTS CONTINUING UNTIL DESCRIBED REAL DEVICE IS ACQUIRED
- EXCEPTIONS:
  - ONLINE I LIMIT
  - ONLINE TAPES: MT01, MT02
OPENING TO PUBLIC FILE

- ASN = FILE
- RN  = DC
- NAME (.ACCOUNT, PASSWORD)
- SER: USUALLY NOT PRESENT

GOES DIRECTLY TO FILE MANAGEMENT

9/19/77     SK
OPENING TO PRIVATE VOLUME FILE

- ASN = FILE (DISK) OR MANAGED TAPE (TAPE)
- RN = DPXX(DISK), MTXX, LTXX (TAPE)
  IF 'XX' = '00 - '99' RN IS VIRTUAL DEVICE
  IF 'XX' = ' ' RESOURCE MANAGEMENT PICKS VIRTUAL DEVICE
- SER: USUALLY PRESENT
- NAME(.ACCOUNT, PASSWORD)
  GOES TO RESOURCE MANAGEMENT TO GET DEVICE THEN TO FILE MANAGEMENT
OPENING TO VIRTUAL DEVICE

ASN = DEVICE

RN· = VIRTUAL DEVICE (MTxx, LP02)

SER = IS USED FOR TAPES

• THE OPEN IS TO A STREAM IF ONE EXISTS BY THE RN NAME - OTHERWISE TO RESOURCE VIA RESOURCE MANAGEMENT
OPENING TO WSN/DEVICE

ASN = DEVICE
RN = OPLABEL or DEVICE TYPE ('LP', 'QQ')
SER: IS NOT USED
WSN: MAY BE PRESENT

- IF WSN IS NULL, WORK STATION OF ORIGIN IS APPLIED
- IF OPLABEL, TRANSLATION PRODUCES STREAM
- ASSIGNMENT IS TO SYMBIONT FILE OR RESOURCE BASED ON WSN DEFINITION
- REMEMBER VIRTUAL DEVICES USE THIS METHOD
- SPECIAL DEVICE TYPES: JE, JF
OPENING TO MASTER T.S. TERMINAL

- SPECIAL UC STREAM POINTS TO TERMINAL

ASN = DEVICE
RN = 'ME' OR OPLABEL POINTING TO UC STREAM

9/19/77 SK
OPENING TO COMGROUP

- ASN = COMGROUP
- SER = COMGROUP NAME

- WAY TO ASSIGN ONE DCB TO MANY DEVICES
- USUALLY COMMUNICATIONS DEVICES
- NEW CONCEPT IN CP-6
  - USED ONLY BY MONITOR IN 1ST RLSE
  - WILL BE USED EVENTUALLY FOR T.P.
- SYMBIONTS and KEYIN USE THEM
- ALLOW RAT, RNT, WLS, WNS
CONNECTING COMMUNICATIONS DEVICES

- ALL CONNECT AND LOG ON TO HOST LOG ON PROCESSOR
- HLP CONNECTS THEM TO
  1) NEW T.S. USER (T.S. TERMS)
  2) RESOURCE (ODD PERIPHERALS)
  3) COMGROUP (LP, CP, CR, OC, TP)

9/19/77    SK
CONNECTING USERS TO DEVICES AND FILES

- PUBLIC FILE
- PRIVATE DISK FILE
- MANAGED TAPE FILE
- VIRTUAL DEVICE
- WSN/DEVICE
- MASTER T.S. TERMINAL
- COMGROUP
FILE MANAGEMENT DEFINITIONS

- GRANULE  UNIT OF ALLOCATION OF DISK STORAGE
  = 1024 WORDS = 1 MEMORY PAGE

- ACCOUNT  1) LOGICAL GROUPING OF FILES, 2) IDENTIFICATION
  OF USER FOR LOGON AND FILE ACCESS

- ACCOUNT DIRECTORY  CATALOG OF ACCOUNTS WITHIN THE SYSTEM OR ON
  A PACK SET

- FILE DIRECTORY  CATALOG OF FILES WITHIN AN ACCOUNT

- FIT  FILE INFORMATION TABLE

- PACK SET  LOGICAL GROUP OF DISK PACKS MAY CONSIST OF
  ONE OR MORE PACKS. MAY CONTAIN FILES FOR
  ONE OR MORE ACCOUNTS
FILE ORGANIZATIONS

- SIX DISK FILE ORGANIZATIONS

  KEYED — EACH RECORD IDENTIFIED BY UNIQUE NAME (KEY) RECORDS STORED IN SORTED ORDER VARIABLE LENGTH RECORDS AND KEYS KEY MAXIMUM LENGTH = 31 CHARACTERS

  CONSECUTIVE — NO UNIQUE RECORD IDENTIFIERS RECORDS WRITTEN AND READ IN SEQUENTIAL ORDER VARIABLE LENGTH RECORDS

  RANDOM — NO KNOWN STRUCTURE

  RELATIVE — NO UNIQUE RECORD IDENTIFIERS OTHER THAN RECORD NUMBER FIXED LENGTH RECORDS

  INDEXED — RECORD IDENTIFIER IS WITHIN DATA VARIABLE LENGTH RECORD AND IDENTIFIER (KEY) KEY LENGTH MAXIMUM = 255 CHARACTERS

  INTEGRATED — RANDOM FILE STRUCTURED BY IDS

- EIGHT ANS TAPE FORMATS

  ANS F — FIXED LENGTH RECORDS
  D — VARIABLE LENGTH WITH DECIMAL CONTROLS
  V — VARIABLE LENGTH WITH ASCII CONTROLS
  U — UNDEFINED

  CP-6 K — KEYED (INDEXED)
  C — CONSECUTIVE (RELATIVE)
  R — RANDOM
  B — BLOCKED
FILE MANAGEMENT SERVICES

• NORMAL SERVICES – OPEN, CLOSE, READ, WRITE, POSITION, DELETE RECORD, DELETE FILE
  AUTOMATIC: BLOCK, DEBLOCK, GRANULE ALLOCATION

• NOT-SO-NORMAL – TEXT DATA COMPRESSION
  WRITE-MORE/READ-LESS (SCATTER I/O)
  SHARE (INPUT, NONE, ALL)
  REWRITE (KEYED, CONSECUTIVE)
  READ/WRITE SEQ OR RANDOM (KEYED)
  AUTOMATIC FILE EXTENSION OR CONCATENATION
  DCB CORRESPONDENCE

• FEATURES – STAR FILES – UNIQUE TEMP FILES
  JOURNAL MODE – (OUTPUT CONSEC)
FILE SECURITY

- GRANULE ACCESS: STAMP
- ACCOUNT ACCESS: READ or CREATE
- FILE ACCESS: MULTIPLE TYPES
- DATA ACCESS: PASSWORD and ENCRYPTION

FILE ACCESS TYPES: READ
UPDATE
WRITE NEW
DELETE RECORDS
DELETE FILE
LIST
FILE INFORMATION
ACCESS VEHICLE
FILE MANAGEMENT PROCESSORS

- INITVOL  INITIALIZE PACK SETS
- CAT      CATALOG PACK SETS and THEIR BACKUP MEDIA
- GAC      GRANULE ACCOUNTING
- LABEL    INITIALIZE ANS TAPES
- EFT      BACKUP/RESTORE and STOW/FETCH
- HGPRECON GRANULE POOL RECONSTRUCTION
INTERNAL FEATURES

- CFU RETENTION: ACTIVE FILES DO NOT REQUIRE DIRECTORY SEARCH for M$OPEN
- READ-AHEAD, WRITE-AHEAD: NO DELAY for PHYSICAL I/O
- I/O CACHE (STICKY PAGES): SIGNIFICANT GRANULES REMAIN IN MEMORY
- RECOVERY and JOB STEP RUNDOWN: DEFAULT BUFFER TRUNCATION and FILE CLOSE
- GRANULE STAMP: NO SCRUB
- COMPRESSED KEYS: SPACE and ACCESS SAVING
- IDS: MODIFICATION DATE per GRANULE
- EFT: BLOCK ACCESS and FILE DESCRIPTORS
USER'S VIEW OF CP-6

- USER AUTHORIZATION
- COMMON COMMAND LANGUAGE
- IBEX (INTERACTIVE and BATCH EXECUTIVE)
- CP-6 PROCESSORS
- USER SERVICES
USER AUTHORIZATION

- CENTRALIZED CONTROL
- INDIVIDUAL AUTHORIZATION
- SYSTEM DEFAULTS and LIMITS
- BUDGET CHECK
- PRE-SCANNED BATCH JOBS
COMMON COMMAND LANGUAGE

- SIMPLE and COMMON SYNTAX RULES
- STANDARD CALLING SEQUENCE
- UNIVERSAL FILE IDENTIFIER (FID)
STANDARD CALLING SEQUENCE

- IANSFORT A OVER B, LP; LS, GO, SI
- IMPLICIT DCB ASSIGNMENT
- OPTIONS FOLLOW THE SEMICOLON
IMPLICIT DCB ASSIGNMENT

" ANSFORT A OVER B, LP; LS, GO, SI"

• LINK RELATES POSITION TO ACTUAL DCB
• IBEX RELATES POSITION TO FID
• STEP CONNECTS FID TO ACTUAL DCB
• IBEX PROVIDES POINTER TO OPTIONS
IBEX

- SINGLE REPLACEMENT FOR TEL/CCI
- EXECUTION CONTROL
- COMMAND FILES ("CATALOGUED PROCEDURES")
- FILE MANAGEMENT INTERFACE
- COMMUNICATIONS INTERFACE
- ACCOUNTING
EXECUTION CONTROL

- STEP CONDITION CODE
- COMMAND VARIABLES
- STEP LIMITS and RESOURCE RETURN
- INTERRUPT and CONTINUE
- BUDGET OVERRUN ABORT
EXAMPLE

ISTEP IF STEPCC ~ = 0, LET A =2

ISTEP IF A ~ = 0, GOTO ABC

ISTEP(ABC)

ILIMIT(STEP) L0 = 99, TIME = 1

ISTEP IF STEPCC ~ = 0, GOTO STEP

...
COMMAND FILES

- AVAILABLE TO ALL MODES
- NESTING ALLOWED
- STRING AND FIELD SUBSTITUTION
FILE MANAGEMENT INTERFACE

- ASSOCIATE A DCB WITH A FID (ISET)
- ARCHIVE STORAGE QUALIFICATION (ISTOW)
- ARCHIVE RETRIEVAL REQUEST (IFETCH)
- STREAM DEFINITION and MANIPULATION
  ILDEV LP07 LP, FORM = CHECKS, LINES = 9
  :
  IERASE LP07
COMMUNICATIONS INTERFACE

- ACCESS TO NETWORK
- TERMINAL PROFILE
- STATISTICS
ACCOUNTING

- BASIC ACCOUNTING
- BUDGET ACCOUNTING
- JOB-STEP ACCOUNTING
- PROPRIETARY PROCESSOR ACCOUNTING
- FORMS ACCOUNTING
CP-6 PROCESSORS

- DELTA
- UTILITY PROCESSORS
- SYSTEM MANAGEMENT PROCESSORS
DELTA

- SINGLE REPLACEMENT FOR 4 CP-V DEBUGGERS
- EXTERNAL TO THE DEBUGGED PROGRAM
- MULTILINGUAL
- EXECUTIVE VERSION IS MONITOR INDEPENDENT
DELTA

- PROGRAM INTERRUPT on SPECIFIED CONDITIONS
- PROCEDURE and DATA MODIFICATION and DISPLAY
- HISTORY TRACE of EXECUTION
- POST-ABORT ASSOCIATION and DEBUGGING
UTILITY PROCESSORS

- EDIT - GENERAL PURPOSE TEXT EDITOR
- PCL - GENERAL PURPOSE FILE MANIPULATOR
- LINK - LINK LOADER
- LEMUR - LIBRARY EDITOR and MAINTENANCE ROUTINE
SYSTEM MANAGEMENT PROCESSORS

- SUPER - INDIVIDUAL USER AUTHORIZATION
- RATES - CHARGE TABLE MAINTENANCE
- CONTROL - SYSTEM PERFORMANCE CONTROL
- DEF - SYSTEM BOOT TAPE CREATION
- ANLZ - SYSTEM CRASH ANALYZER
USER SERVICES

- HELP COMMAND
- STANDARD ERROR HANDLING
- FID DECODER (M$FID)
- GENERAL PURPOSE OUTPUT FORMATTER (M$FORMAT)
- CP-V TO CP-6 JCL CONVERTER
- GENERAL PURPOSE SYNTAX PARSER (M$PARSE)
M$PARSE

- SINGLE PARSING ROUTINE FOR ALL UTILITY PROCESSORS
- POWERFUL TABLE DRIVEN STRUCTURE
- EASILY EXTENDABLE
MISCELLANEOUS TOPICS

- SYSTEM INITIALIZATION and CONFIGURATION CONTROL
- SYSTEM RECOVERY
- PERFORMANCE MONITORING and CONTROL
- ERROR LOGGING and LISTING
SYSTEM INITIALIZATION AND CONFIGURATION CONTROL-1

- GOALS
  - MINIMIZE CODE SPECIFIC TO SYSTEM BUILD
  - SIMPLIFY SYSTEM BUILD PROCESS OVER CP-V
  - BOOT-TIME RECONFIGURATION CAPABILITY
  - STANDARD SYSTEM, READY TO RUN
    - SYSTEM PACKAGING
      - THE MINIMUM SYSTEM
      - DEFAULT CONTROL PARAMETERS
  - RELOAD SYSTEM WHEN ADDITIONAL FEATURES REQUIRED
    - USES STANDARD PROCESSORS
SYSTEM INITIALIZATION AND CONFIGURATION CONTROL-II

- BOOTING THE SYSTEM
  - BOOT ROUTINE (AARDVARK)
    .MINI - I/O
    .MINI - XDELTAS (SUBSET OF XDELTAS)
    .MPC INITIALIZATION (TAPE, DISK, UNIT RECORD)
  - SYSTEM IMAGE
  - TIGR
  - XDELTAS
  - DEBUG SCHEMA
  - GHOST 1
  - PATCHES
  - PROCESSOR PATCHES (GENMDS)
  - CONFIGURATION CONTROL CARDS
  - LABELLED TAPE PORTION
  - PROCESSORS
SYSTEM INITIALIZATION AND CONFIGURATION CONTROL-III

- TIGR - TABLE INITIALIZATION and GENERATION ROUTINE
  - ESTABLISH HARDWARE CONFIGURATION TABLES
    .DCT
    .CIT
    .PPUT
    .ETC.
  - ESTABLISH DYNAMIC INSTALLATION DEPENDENT TABLES
    .USER TABLES
    .I/O QUEUE SPACE
    .ETC.

- SYSCON
  - RUN TIME PARTITIONING PROCESSOR
SYSTEM INITIALIZATION AND CONFIGURATION CONTROL - IV

- CONTROL (STANDARD SYSTEM PROCESSOR)
  - SETS INSTALLATION DEPENDENT CONTROL AND DEFAULT PARAMETERS
  - USED TO CHANGE CONTROL/DEFAULT PARAMETERS AS WORK PROFILE CHANGES
  - TYPES OF PARAMETERS
    . NUMBER OF USERS
    . CORE USAGE
    . MULTIPROCESSING CONTROL
    . SCHEDULER CONTROL
    . BATCH STREAM CONTROL
    . I/O ACCELERATOR CONTROL
    . JOB SERVICE LIMITS
    . JOB RESOURCE LIMITS
    . JOB DEFAULTS
RECOVERY

- GOALS
  - CENTRALIZED SYSTEM ROUTINE (SCRECH)
  - AUTOMATIC, NO OPERATOR INTERVENTION
    . INITIATED VIA FAULT OR LOGICAL INCONSISTENCY
  - FAST, 20 SEC TO 2 MINUTES
  - MINIMIZE FILE LOSS

- RECOVERY TYPES
  - ZAP/WARM BOOT
  - SCRECH
    . SINGLE USER ABORT
    . MULTI-USER ABORT
    . FULL SYSTEM

- POWER FAIL SAFE RECOVERY
  - NON-VOLATILE MEMORY
  - MPC STATE

- CREATES DUMP WITH FORMATTED OUTPUT
PERFORMANCE AND CONTROL

• GOALS
  - LIKE CP-V ONLY BETTER

• TOOLS
  - INTERNAL
    - BUILT IN DATA COLLECTION
      - RESPONSE TIMES
      - DEVICE and CHANNEL BUSY TIMES
      - CPU UTILIZATION
    - PERFORMANCE TUNING VARIABLES
      - SCHEDULER CONTROLS
        - QUANTUMS: QMIN, QUAN
        - I/O BLOCK and UNBLOCK LIMITS
        - BASE EXECUTION PRIORITIES: O, B, G
        - I/O TIME ALLOWANCE
      - I/O ACCELERATOR CONTROL
        - READ AHEAD
        - DISASSOCIATED WRITE
        - SEVERAL CACHE TYPES
  - EXTERNAL
    - STATS
    - CALMON
    - SUMMARY
### CALMON OUTPUT

#### SUMMARY STATISTICS

<table>
<thead>
<tr>
<th>CALI,1 REPORTS</th>
<th>CAL COUNT</th>
<th>% OF CALS</th>
<th>CPU TIME</th>
<th>% CAL</th>
<th>% ALL</th>
<th>AVG CPU TIME</th>
<th>AVG CPU TIME PER CAL</th>
<th>I/O COUNT</th>
<th>% CAL</th>
<th>AVG I/O</th>
<th>AVG I/O PER CAL</th>
<th>AVG BYTE PER CAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td>141</td>
<td>.05</td>
<td>158</td>
<td>.02</td>
<td>0</td>
<td>1.12</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>693.3</td>
</tr>
<tr>
<td>WRITE</td>
<td>705</td>
<td>.25</td>
<td>976</td>
<td>.10</td>
<td>0</td>
<td>1.36</td>
<td>640</td>
<td>.60</td>
<td>.91</td>
<td>99.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEVICE(PAGE)</td>
<td>572</td>
<td>.20</td>
<td>1,500</td>
<td>.16</td>
<td>.01</td>
<td>2.77</td>
<td>47</td>
<td>.04</td>
<td>.08</td>
<td>371.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEVICE(VFC)</td>
<td>506</td>
<td>.18</td>
<td>240</td>
<td>.03</td>
<td>0</td>
<td>.49</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>257.3</td>
</tr>
<tr>
<td>SEEK</td>
<td>17,606</td>
<td>6.25</td>
<td>8,516</td>
<td>.87</td>
<td>.04</td>
<td>.48</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>917.1</td>
</tr>
<tr>
<td>DEVICE(DNC)</td>
<td>9</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>.67</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3,097.7</td>
</tr>
<tr>
<td>DELREC</td>
<td>51</td>
<td>.02</td>
<td>352</td>
<td>.04</td>
<td>0</td>
<td>6.70</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.06</td>
<td></td>
<td>117.6</td>
</tr>
<tr>
<td>MOVE</td>
<td>652</td>
<td>.23</td>
<td>146,746</td>
<td>15.01</td>
<td>.74</td>
<td>225.06</td>
<td>15,065</td>
<td>15.02</td>
<td>24.64</td>
<td>1.195</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILE</td>
<td>26</td>
<td>.01</td>
<td>408</td>
<td>.04</td>
<td>0</td>
<td>15.69</td>
<td>76</td>
<td>.07</td>
<td>3.00</td>
<td>9,377.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>READ(FILE)</td>
<td>96,007</td>
<td>34.27</td>
<td>211,120</td>
<td>21.60</td>
<td>1.06</td>
<td>2.20</td>
<td>27,100</td>
<td>25.34</td>
<td>.28</td>
<td>519.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>READ(LBL TAPE)</td>
<td>2,153</td>
<td>.77</td>
<td>4,252</td>
<td>.43</td>
<td>.02</td>
<td>1.97</td>
<td>95</td>
<td>.09</td>
<td>.04</td>
<td>80.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>READ(DEVICE)</td>
<td>10</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>3.00</td>
<td>10</td>
<td>0</td>
<td>1.00</td>
<td>-</td>
<td>-</td>
<td>6.4</td>
</tr>
<tr>
<td>READ(COC)</td>
<td>3,700</td>
<td>1.32</td>
<td>14,432</td>
<td>1.48</td>
<td>.07</td>
<td>3.90</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>112.5</td>
</tr>
<tr>
<td>READ(SYMBOINT)</td>
<td>791</td>
<td>.28</td>
<td>2,392</td>
<td>.24</td>
<td>.01</td>
<td>3.02</td>
<td>74</td>
<td>.07</td>
<td>.09</td>
<td>112.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRITE(FILE)</td>
<td>26,706</td>
<td>9.53</td>
<td>55,414</td>
<td>5.69</td>
<td>.28</td>
<td>2.08</td>
<td>4,761</td>
<td>4.45</td>
<td>.18</td>
<td>233.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRITE(LBL TAPE)</td>
<td>27,453</td>
<td>9.80</td>
<td>56,424</td>
<td>5.77</td>
<td>.28</td>
<td>2.06</td>
<td>6,383</td>
<td>5.97</td>
<td>.23</td>
<td>631.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRITE(DEVICE)</td>
<td>10,657</td>
<td>3.59</td>
<td>17,210</td>
<td>1.76</td>
<td>.09</td>
<td>1.71</td>
<td>9,534</td>
<td>8.92</td>
<td>.95</td>
<td>552.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRITE(COC)</td>
<td>8,937</td>
<td>3.19</td>
<td>43,330</td>
<td>4.43</td>
<td>.22</td>
<td>4.85</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>14.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRITE(SYMBOINT)</td>
<td>30,970</td>
<td>11.05</td>
<td>60,294</td>
<td>6.17</td>
<td>.30</td>
<td>1.95</td>
<td>2,405</td>
<td>2.25</td>
<td>.08</td>
<td>89.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>THRU</td>
<td>66</td>
<td>0.02</td>
<td>190</td>
<td>.02</td>
<td>0</td>
<td>2.88</td>
<td>88</td>
<td>.08</td>
<td>1.33</td>
<td>80.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPEN(FILE)</td>
<td>5,156</td>
<td>1.83</td>
<td>70,700</td>
<td>7.29</td>
<td>.36</td>
<td>13.78</td>
<td>5,103</td>
<td>4.85</td>
<td>1.01</td>
<td>4,015.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPEN(LBL TAPE)</td>
<td>1,198</td>
<td>.43</td>
<td>11,904</td>
<td>1.22</td>
<td>.06</td>
<td>9.94</td>
<td>4,788</td>
<td>4.48</td>
<td>4.00</td>
<td>452.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPEN(DEVICE)</td>
<td>59</td>
<td>0.02</td>
<td>296</td>
<td>.03</td>
<td>0</td>
<td>5.02</td>
<td>96</td>
<td>0.96</td>
<td>1.63</td>
<td>477.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPEN(COC)</td>
<td>176</td>
<td>.06</td>
<td>390</td>
<td>.04</td>
<td>0</td>
<td>2.22</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>8.8</td>
</tr>
<tr>
<td>OPEN(SYMBOINT)</td>
<td>524</td>
<td>.19</td>
<td>1,236</td>
<td>.13</td>
<td>.01</td>
<td>2.36</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2.2</td>
</tr>
<tr>
<td>OPEN(ISI-FIL)</td>
<td>8,096</td>
<td>2.89</td>
<td>69,976</td>
<td>7.16</td>
<td>.35</td>
<td>8.64</td>
<td>599</td>
<td>.56</td>
<td>.07</td>
<td>5,301.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLOSE(FILE)</td>
<td>4,052</td>
<td>1.45</td>
<td>41,196</td>
<td>4.21</td>
<td>.21</td>
<td>10.15</td>
<td>9,970</td>
<td>9.32</td>
<td>2.45</td>
<td>1,016.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLOSE(LBL TAPE)</td>
<td>1,171</td>
<td>.42</td>
<td>5,586</td>
<td>.57</td>
<td>.03</td>
<td>4.77</td>
<td>8,099</td>
<td>7.57</td>
<td>6.92</td>
<td>327.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLOSE(DEVICE)</td>
<td>736</td>
<td>.26</td>
<td>842</td>
<td>.09</td>
<td>0</td>
<td>1.14</td>
<td>9</td>
<td>.01</td>
<td>.01</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLOSE(COC)</td>
<td>232</td>
<td>.08</td>
<td>302</td>
<td>.03</td>
<td>0</td>
<td>1.30</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>17.4</td>
</tr>
</tbody>
</table>
STATISTICS ON-LINE

INTERVAL FROM 14:51 TO 14:55

MINS SINCE STARTUP = 992
NUMBER OF USERS = 37
NUMBER OF ONLINE = 26
NUMBER OF BATCH = 5
NUMBER OF GHOSTS = 6
90% RESPONSE TIME = 1500

<table>
<thead>
<tr>
<th>CPU %</th>
<th>ALL SNAP</th>
<th>I/O PER MIN</th>
<th>ALL SNAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATCH EXEC</td>
<td>42.6 94.3</td>
<td>SERVICE REQ</td>
<td>2897 9605</td>
</tr>
<tr>
<td>BATCH SERV</td>
<td>5.0 16.4</td>
<td>INTERACTIONS</td>
<td>19 29</td>
</tr>
<tr>
<td>ONLINE EXEC</td>
<td>14.4 34.2</td>
<td>CHAR IN</td>
<td>265 293</td>
</tr>
<tr>
<td>ONLINE SERV</td>
<td>5.2 14.9</td>
<td>CHAR OUT</td>
<td>2414 3572</td>
</tr>
<tr>
<td>GHOST EXEC</td>
<td>1.6 7</td>
<td>TERM WRITES</td>
<td>98 150</td>
</tr>
<tr>
<td>GHOST SERV</td>
<td>1.3 3</td>
<td>I/O ACCESSES</td>
<td>676 1269</td>
</tr>
<tr>
<td>MONITOR SERV</td>
<td>8.5 15.7</td>
<td># TRUNCS</td>
<td>0 0</td>
</tr>
<tr>
<td>IDLE</td>
<td>48.3 0</td>
<td>AIR ATTEMPTS</td>
<td>71 89</td>
</tr>
<tr>
<td>SWAP WAIT</td>
<td>.4 7</td>
<td>AIP HITS</td>
<td>56 35</td>
</tr>
<tr>
<td>I/O WAIT</td>
<td>8.6 1.9</td>
<td>SYMBIONT</td>
<td>74 68</td>
</tr>
<tr>
<td>I/O&amp;Swp WAIT</td>
<td>1.3 12.1</td>
<td>IN SWAPS</td>
<td>30 99</td>
</tr>
<tr>
<td>TOTAL</td>
<td>137.2 191.3</td>
<td>OUT SWAPS</td>
<td>21 67</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SCPU USE %</th>
<th>ALL SNAP</th>
<th>EVENT RATE/MIN</th>
<th>ALL SNAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCPU #1 EXEC</td>
<td>37.3 91.0</td>
<td>MASTER CALS</td>
<td>2062 6224</td>
</tr>
<tr>
<td>SCPU #1 IDLE</td>
<td>62.1 9.2</td>
<td>SCPU #1 CALS</td>
<td>836 3381</td>
</tr>
<tr>
<td>SCPU #1 TOT</td>
<td>99.4 100.2</td>
<td>MASTER SCHEDS</td>
<td>2135 6417</td>
</tr>
<tr>
<td>SCPU #1 SCHEDS</td>
<td>751 3178</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HOW DO WE KNOW WE'LL BE FAST ENOUGH

- ALGORITHMS PROVEN IN CP-V
- USAGE PATTERNS KNOWN
- INTEGRAL PERFORMANCE MONITOR
- HIGH LEVEL LANGUAGE PERMITS SYSTEM WIDE OPTIMIZATION
- USE OF STANDARD BENCHMARKS
- NSA HARDWARE USED FOR CONTEXT SWITCH
- OFFLOADING OF COMMUNICATIONS TO L6 FRONT ENDS
ERROR LOGGING AND LISTING

• GOALS
  - NEW TOOL COMMON WITH GCOS, TCOS
  - ONLINE and BATCH OPERATION
  - DESIGNED BY FED FOR FED

• ERROR TYPES
  - 100 INFORMATIONAL
  - 200 OPERATIONAL
  - 300 HARDWARE ERRORS
  - 400 SOFTWARE ERRORS

• REPORT TYPES
  - MASS RAW DATA DUMPS
  - CHRONOLOGICAL LISTING
  - MEDIA ALERTS
  - ERROR BY DEVICE/COMMAND
  - ERROR BY MEDIA ID/UNIT
  - SUMMARY REPORT
    - RECOVERY
      - MPC STATISTICS
      - EDAC SYNDROME

9/19/77 DAY
STRENGTH OF CP-6 IS IN ITS INTERFACES

- USER STRUCTURE
- SYSTEM SERVICE INTERFACE
- PROGRAM BINDING INTERFACE
- PROGRAM CALLING INTERFACE
- USER to SYSTEM INTERFACE
COMMON/SHARED PARTS OF CP-6

• USER STRUCTURE
• SCHEDULER
• ALL I/O INCLUDING FILE MANAGEMENT
• SHARED PROCESSORS
• SHARED LIBRARIES (RUNTIME AND ALTERNATE)
• COMMAND PROGRAM
• DEBUGGER
• SPECIAL SERVICES (M$FID, M$PARSE, M$ERRPRT, HELP)

9/19/77
CP-6/1.0 RT&C PROJECT

- PROJECT INTRODUCTION
  -- ORGANIZATION
  -- KEY EVENTS
  -- PROJECT OVERVIEW
  -- ISSUES
- SOFTWARE FACTORY
  -- IMPLEMENTATION LANGUAGE
  -- PROGRAM TRANSPORTATION
  -- DEBUGGING
  -- SUPPORT TOOLS
  -- RELEASE SUPPORT
- RELEASE 1.0
  -- TECHNICAL DESCRIPTION - OVERVIEW
  -- TECHNICAL DESCRIPTION - HARDWARE
  -- TECHNICAL DESCRIPTION - SOFTWARE
REAL-TIME AND COMMUNICATIONS

C. MARTIN
Manager

---

CP-6 1.0
T. Cox
Project Leader

- S. Bradford
- M. Hasson
- T. Melton
- P. Stendal
- R. Wu
- New Hire

---

RT&C Architecture
I. Greenwald
Project Leader

- J. Hellerstein
- S. Lamoree
- G. Mallory
- R. Nixon
- (D. Heying*)
- (S. Keys*)

* Principal participants from CP-6 OS Section, as required.
RT&C PROJECT – KEY EVENTS

- SCHEDULE CONSTRAINTS
- COUPLER
- CONTROL FORTRAN AND SIL-6
RT&C PROJECT - SCHEDULE CONSTRAINTS

- LANGUAGE PROCESSOR CHECKOUT
  -- APPROACH APRIL/MAY, 1978
  -- PRODUCTIVE JULY, 1978

- SERVICE PROCESSOR CHECKOUT
  -- BEGIN T/S ENVIRONMENT CHECKOUT JAN/FEB, 1978
  -- COMPLETE CHECKOUT MARCH/APRIL, 1978

- BASIC T/S CAPABILITY REQUIRED BY JANUARY, 1978
RT&C PROJECT - COUPLER

- MARCH 1977 - COUPLER (PROTOTYPE) COMMITTED 4Q77.
- MAY 1977 - COUPLER COMMITTED TO 746.
- JULY 1977 - COUPLER CDR HELD
- NOVEMBER 1977 - INSTALL COUPLER AT LADC
- DECEMBER 1977 - COUPLER OPERATIONAL AT LADC
RT&C PROJECT – CONTROL FORTRAN AND SIL-6

- 1976 - ASSUMED AVAILABILITY OF CF IN FEBRUARY 1977
- FEBRUARY 1977 - DELIVERY SLIPPED TO MARCH 1977
- APRIL 1977 - CF PDR HELD
- MAY 1977 - CF RESPONSIBILITY MOVED TO BILLERICA
- MAY 1977 - SIL-6 DECISION MADE
- JULY 1977 - WORKING VERSION OF SIL-6 AVAILABLE FOR USE
RT&C PROJECT – RELEASE CONTENT

- RELEASE 1.0
  -- CP-V BASED
  -- TTY TERMINAL SUPPORT
  -- IRBT, HASP
  -- RBT, 2780/3780
  (--- CP-V SOFTWARE FACTORY)

- RELEASE 2.0
  -- REAL TIME
  -- REMOTE CP
  -- VIP TERMINAL SUPPORT
  -- CP-6 SOFTWARE FACTORY
  -- TRANSACTION PROCESSING

- RELEASE 3.0
  -- 66/85 SUPPORT
A - SYSTEM (805T/S) - JAN 1978

-- INTEGRATED INTO HOST H-SYSTEM

-- SUPPORT FOR DELTA, CP AND USER PROGRAMS

-- TI SUPPORT

-- ECHOPLEX

-- CANNED SALUTATION AND PROMPTS

-- NO ESCAPE SEQUENCES

-- CP-V ACTION FOR CONTROL Y,X AND BREAK AND RUB-OUT
B - SYSTEM - JULY 1977

-- INTEGRATED INTO HOST L-SYSTEM

-- T/S FOR LANGUAGE PROCESSOR CHECKOUT

-- TAB STOPS

-- TYPE AHEAD

-- ASCII APL TERMINAL SUPPORT

-- SYMBIONT I/O MODE

-- OC I/O MODE

-- USER PROGRAM ACCESS TO TERMINAL CONTROL ATTRIBUTES

-- MAJORITY OF ESCAPE SEQUENCES

9/19/77 PDR/CM
C - SYSTEM

-- INTEGRATED INTO HOST O-SYSTEM

-- RELEASE 1.0 FUNCTIONALITY

-- 2780/3780 AND HASP SUPPORT

-- AUTOBAUD

-- FULL TAB CAPABILITY

-- PAGINATION

-- TRANSPARENT I/O

-- FULL ESCAPE SEQUENCE CAPABILITY
RT&C PROJECT - ISSUES

- SNA AND HDNA
- CP-6 SOFTWARE FACTORY
- DIAGNOSTICS
- HOST RESIDENT SYSTEM
- REAL TIME
RT&C PROJECT - SNA, HDNA ISSUE

- NOT PART OF ORIGINAL PROJECT STRATEGY - CHANGE OF SCOPE

- SNA AND IBM CO-RESIDENCY SEEN AS HIGHER PRIORITY BY MARKETING

- HDNA AS DEFINED IS NOT CONSISTENT WITH CP-6 RESPONSE TIME REQUIREMENTS

- CONTINUES TO EVOLVE

- NET RESULT - NOT CONSIDERED IN CP-6/1.0
RT&C PROJECT - CP-6 SOFTWARE FACTORY

- MOVE FROM CP-V TO CP-6

- SIL-6

- DUAL (DYNAMIC UNIVERSAL ASSEMBLY LANGUAGE)

- L6 ASSEMBLER

- L6/PL-6

- BILLERICA SYSTEMS IMPLEMENTATION LANGUAGE
  -- PL-6
  -- PLH

9/19/77  PDR/CM
16 SOFTWARE FACTORY ENVIRONMENTS

1. INITIAL (RELEASE 1.0) CP-6 DEVELOPMENT USING CP-V RESIDENT SOFTWARE FACTORY

2. RELEASE 2 AND 3 DEVELOPMENT USING CP-6 RESIDENT SOFTWARE FACTORY

3. CUSTOMER REAL-TIME DEVELOPMENT (RELEASE 2 AND BEYOND) USING CP-6 RESIDENT FACTORY
CP-V RESIDENT L6 SOFTWARE FACTORY

L6 ASM TO SIL6 ASM CONVERTER

SIL6 SOURCE

1. SIL6 ASM
2. SIL6 HIGH LEVEL LANGUAGE
3. MLCP MACROS

CP-V EDIT

SIL6 SOURCE

SIL6 COMPILER

CP-V OBJECT UNIT FORMAT

CP-V OBJECT UNIT FORMAT

LYNX

PROGRAM TRANSPORTATION SYSTEM

9 TRK TAPE

BOOTABLE TAPE

9 TRK TAPE

L6 RUN UNIT FILE TRANSFER

CP-V L66

L6

COUPLER

9/14/77 TC
SIL6 LANGUAGE

- HIGH LEVEL LANGUAGE
- DATA INDEPENDENCE
- BLOCK PROGRAM STRUCTURES
- ACCESSIBILITY TO L6 REGISTERS
- SYMBOL REF/DEF STACK FOR SYMBOLIC DEBUGGING
- SUPPLEMENTARY ASSEMBLER MNEMONICS FOR TIME-CRITICAL OR HARDWARE RELATED CODE
- MLCP MNEMONICS
- IMPLEMENTATION IN AP-EASY TO EXPAND
SUMMARY OF SIL6 CAPABILITIES

- PROGRAM STRUCTURE COMMANDS
  - LOOP/EXITLOOP/ENDLOOP
  - IF/ELSEIF/OTHERWISE/ENDLOOP
  - DOCASE/CASE/ENDCASE
  - CALL/SUBROUTINE/RETURN

- ITEM DEFINITION FUNCTIONS
  - TYPE - BIT, BYTE, WORD, DOUBLEWORD, POINTER
  - ATTRIBUTES - SIGNED, WRITE PROTECT, DIMENSION, FIELD SPECIFICATION WITHIN A WORD, OFFSET FROM A BASE, FIXED ADDRESS
  - ITEM MAY BE MEMORY LOCATION OR REGISTER
  - LITERAL ADDRESSES AND VALUES

9/14/77 TC
SUMMARY OF SIL6 CAPABILITIES - CONTINUED

- ITEM MANIPULATION COMMANDS
  - MOVE
  - INCREMENT/DECREMENT
  - SETT/RESET
  - CLEAR

- COMPARISON FUNCTIONS
  - NOT
  - EQUAL/NEQUAL
  - GT/AGT
  - GTE/AGTE
  - LT/ALT
  - LTE/ALTE
• COMPUTING FUNCTIONS
  • SUM
  • DIF
  • PRODUCT
  • QUOTIENT
  • MODULO
  • AND
  • OR
  • EOR/XOR

• NON-COMPUTING FUNCTIONS
  • HIGHDIM/LOWDIM
  • ASC
  • MAXOFFSET/MINOFFSET/NEXTOFFSET
  • AMAX/AMIN
  • ADDR
SUMMARY OF SIL6 CAPABILITIES - CONTINUED

• ASSIGNED HARDWARE REFERENCES
  • R1 - R7
  • B1 - B7
  • CARRY/NOCARRY
  • OVERFLOW/NOOVERFLOW
  • IOACK/IONACK
  • LEVEL
  • INHIBIT_LEVEL

• MISCELLANEOUS COMMANDS
  • ENABLE
  • DISABLE/INHIBIT
  • TRIGGER
SUMMARY OF SIL6 CAPABILITIES - CONTINUED

- DECISION TABLE PROCEDURES
  - TABLE
  - TEST(S)
  - RULE
  - ACTION(S)
  - REPEAT
  - RETURN_TRUE
  - RETURN_FALSE
  - ENDTABLE
SUMMARY OF SIL6 CAPABILITIES - CONTINUED

- ASSEMBLY LIST CONTROL
  - PAGE
  - TITLE
  - WARNING
  - Lolist
  - HEX

- ASSEMBLY CONTROL STATEMENTS
  - DEBUG
  - STARTSIL6
  - STARTASM
  - LABEL
  - ORG
  - ODDBOUND/EVENBOUND
  - DC
  - RESV
  - ATEXT/ATEXTC
SIL6 EXAMPLE

ALPHA  FNAME  WORD, (OFFSET,3), (FIELD,3,7), WP
BETA   FNAME  WORD, (OFFSET,15), (FIELD,1,9), (DIM, 10)
GAMMA  FNAME  BYTE, (OFFSET,1)
BASE   FNAME  (BREG, 1)
INDEX  FNAME  (RREG, 2), (FIELD, 0, 7)
       .
       .
IF EQUAL (GAMMA(BASE), ALPHA(BASE))
MOVE ALPHA(BASE), TO, BETA(BASE, INDEX))
OTHERWISE
MOVE SUM(ALPHA(BASE), BETA(BASE, INDEX)), TO, GAMMA(BASE)
INCREMENT INDEX
ENDIF

9/14/77 TC
SIL6 DECISION TABLE EXAMPLE

<table>
<thead>
<tr>
<th>TESTCASE</th>
<th>TABLE</th>
<th>PASSTEXT,TEXT1,TEST2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RULE</td>
<td>T,F,T</td>
<td>Action1,Action2,REPEAT</td>
</tr>
<tr>
<td>ACTIONS</td>
<td>ACTION1,ACTION2,REPEAT</td>
<td>SUM(B3,B4),TO,B4</td>
</tr>
</tbody>
</table>

ACTION2 SUBROUTINE NOLOCAL(B4)

IF R4 RETURN TRUE
ENDIF
RETURN FALSE
ENDSUB

ACTION2 SUBROUTINE SAVE(R2,R4)
MOVE R5,TO,R4
ENDSUB
DEBUGGING

• SIMULATOR AND L6 DEBUGGER - SUBSET OF CP-6 DELTA FUNCTIONALITY AND COMMAND SYNTAX
  • BREAKPOINTS
  • TRACE
  • STEP
  • MODIFY
  • DISPLAY
  • DUMP
  • SYMBOLIC INPUT
DEBUGGING - CONTINUED

- BES2 MDUMP
- UPLINE DUMP VIA COUPLER
- DUMP ANALYZER
- MLCP DEBUGGER
- EVENT RECORDING

9/14/77 TC
SIMULATOR

- HANDLES INTERRUPTS, TRAPS
- L6/36 OR L6/43 INSTRUCTION SET ACCEPTED
- PROVIDES INSTRUCTION TIMING SUMMATION
- I/O INSTRUCTIONS CAUSE DISPLAY OF PERTINENT INFORMATION – NO I/O SIMULATION IS ATTEMPTED
- EXTENSIVE DEBUGGER

9/14/77 TC
SUPPORT TOOLS

- EXTRACT
- DRAW
- L6 ASM TO SIL6 ASM CONVERTER
- GLOBAL XREF
RELEASE SUPPORT

- PATCHER - WILL PATCH L6 RUN UNIT ON CP-V OR CP-6
- REMOTE DEBUGGER
- UPLINE DUMP VIA COUPLER
- DUMP ANALYZER
CONVENTIONS AND STANDARDS

- CP-6 COMPATIBLE
- DOCUMENTATION STANDARDS
- NAMING CONVENTIONS AND FUNCTIONAL CODE GROUPS
- CODING STANDARDS
- CONTROLLED DEVELOPMENT ACCOUNTS
CP-V RESIDENT L6 SOFTWARE FACTORY

1. SIL6 ASM
2. SIL6 HIGH LEVEL LANGUAGE
3. MLCP MACROS

DONE

L6 ASM TO SIL6 ASM CONVERTER

SIL6 SOURCE

CP-V EDIT

SIL6 SOURCE

SIL6 COMPILER

CP-V OBJECT UNIT FORMAT

LYNX

PROGRAM TRANSPORTATION SYSTEM

CP-V EDIT

EXISTS IN CP-V

SIL6 SOURCE

ESSENTIALLY DONE, UNDERGOING MINOR ENHANCEMENTS

SIL6 COMPILER

CP-V OBJECT UNIT FORMAT

LYNX

EXISTS IN CP-V

SIMULATOR

ESSENTIALLY DONE, UNDERGOING MINOR ENHANCEMENTS

RUN UNIT FILE TRANSFER

L6 RUN UNIT FILE TRANSFER

CP-6 L66

IN DESIGN PHASE

COUPLER

L6

9 TRK TAPE

BOOTABLE TAPE

IN CODING AND CHECKOUT PHASES

9 TRK TAPE

IN DESIGN PHASE

L6

9/14/77 TC
CP-6 COMMUNICATIONS - RELEASE 1.0

- FUNCTIONALITY
- HARDWARE
  - OVERVIEW
  - L66/L6 COUPLER
  - LEVEL 6 MLCP
- LEVEL 6 SOFTWARE DESIGN
  - OVERVIEW
  - NUCLEUS
  - FRONT END INTERFACE
  - COC HANDLER
  - RBT/IRBT HANDLER
- LEVEL 66 SOFTWARE
COMMUNICATIONS SOFTWARE REQUIREMENTS – RELEASE 1

- OFFLOAD PROCESSING FROM THE L66
- USE LEVEL 6/43
- CP-V STYLE TERMINAL SUPPORT
  - TTY COMPATIBLE TERMINALS
  - 2780/3780 COMPATIBLE RBTS
  - HASP COMPATIBLE IRBTS
  - CUSTOM DEVICES (TRANSPARENT I/O)
- HOST TO HOST COMMUNICATION (CP-6 TO CP-6, CP-6 TO CP-5)
- ESTABLISH A BASE FOR FUTURE RELEASES

9/19/77  TM
L66/L6 COUPLER HARDWARE

- HOST INITIATED BOOT LOAD
- BYTE ORIENTED ASCII MODE; 1 MEGABYTE TRANSFER
- BINARY MODE; 500 KB TRANSFER
- LEVEL 6 INITIATED I/O WITH COMMAND CHAINING
MLCP HARDWARE CAPABILITIES

• GENERAL CAPABILITIES
  o PROGRAMMABLE
  o EIGHT RECEIVE/TRANSMIT CHANNEL PAIRS
  o CONTROL CHARACTER AND END-OF-MESSAGE DETECTION
  o L6 COMMAND CHAINING CAPABILITY

• ASYNCHRONOUS LINE SUPPORT (COC)
  o PROGRAMMABLE LINE SPEED AND CHARACTER FORMAT (AUTOBAUD)
  o SPEEDS UP TO 19.2K BITS/SEC

• SYNCHRONOUS LINE SUPPORT
  o AUTOMATIC CYCLIC REDUNDANCY CHECK
  o SPEEDS UP TO 72K BITS/SEC
LEVEL 6 SOFTWARE STRUCTURE FOR RELEASE 1

LEVEL 6

HOST FRONT END HANDLER

LEVEL 6 HOST

SOFTWARE STRUCTURE

LEVEL 6

LEVEL 6 NUCLEUS SERVICES

LEVEL 6 HANDLER

RBT HANDLER

HASP (IRBT) HANDLER

COC HANDLER

LEVEL 6 COUPLER

MLCP

MLCP

OTHER HOST MONITOR SERVICE FUNCTIONS

LEVEL 66

9/19/77 TM
LEVEL 6 NUCLEUS SERVICES

- CLOCK SERVICES
- MEMORY ALLOCATOR
- SCHEDULER
  - WAKE-UP
  - GENERAL SERVICE
  - LOGICAL PATH SERVICE
  - MESSAGE SERVICE
- ADMINISTRATIVE SERVICES
  - LOGICAL PATH CONNECTION, DISCONNECTION
  - ERROR LOGGING
  - CONFIGURATION CONTROL

9/19/77  TM
L6 FRONT END INTERFACE STRUCTURE AND MESSAGE FLOW

- **L66 Input Circular Buffer**
  - L66 Input Messages
  - **L66 COUPLER**
  - **FRONT-END L66 MESSAGE SENDER (SEND)**
  - FROM TRANSMITTING PROCESSES BUFFER

- **L66 Output Circular Buffer**
  - L66 Output Messages
  - **FRONT-END L66 OUTPUT MESSAGE RECEIVER (RCVR)**
  - TO SCHEDULER FOR MESSAGE DELIVERY

- **L66 Symbiont Buffer Blocks**
  - Symbiont Data
  - **FRONT-END SYMBIONT READER (SYMB)**
  - TO RECEIVING PROCESSES BUFFER

---

LEVEL 66 | LEVEL 6

**MESSAGE PATHS**

**CONTROL PATHS**

9/19/77   TM
CP-6 LEVEL 6 COC HANDLER CAPABILITIES

- TYPE AHEAD
- TERMINAL TYPES AND TIMING ALGORITHMS
- TRANSPARENT I/O
- PAGINATION
- INPUT EDITING AND CONVERSION
- TABULATION CONTROL
- TERMINAL COUPLING
- TERMINAL TAPE INPUT
COC USER CONTEXT MAINTAINED IN THE LEVEL 6

- INPUT PROMPTS
- TAB STOPS
- ACTIVATION SET
- TERMINAL TYPE, AND TRANSLATION
- TIMING ALGORITHM
- MODES OF OPERATION IN EFFECT
- IMAGE OF PREVIOUS INPUT RECORD FOR RECALL
CP-6 LEVEL 6 COC HANDLER STRUCTURE

FROM MESSAGE SERVICE SCHEDULER

MESSAGE REQUEST HANDLER

MESSAGE BUFFERS

OUTPUT PROCESSOR

CHARACTER OUTPUT BUFFERS

OUTPUT INTERRUPT RECEIVER

MESSAGE ASSEMBLER

CHARACTER INPUT BUFFERS

INPUT INTERRUPT RECEIVER

INPUT MESSAGE HANDLER

MESSAGE BUFFERS

SEND MESSAGE TO HOST

ECHOPLEX CHARACTER BUFFERS

ASYNCHRONOUS MLCP HANDLER

SEND MESSAGE TO HOST

9/19/77 TM
CP-6 REMOTE BATCH CAPABILITIES

- SUBMIT JOBS, RECEIVE RESULTS, AND TRANSMIT FILES
- 2780/3780 AND HASP-MULTILEAVING PROTOCOL SUPPORT
- HOST TO HOST COMMUNICATION
  - MASTER OR SLAVE OVER EACH CONNECTION
- WORKSTATION CONCEPT
  - CP-6 HOST SUPPLIES CHARACTERICS DURING CONNECTIONS
  - WIDE RANGE OF DEVICES CAN BE SUPPORTED
- LOGICAL PATH TO HOST FOR EACH ACTIVE DEVICE
- REMOTE OPERATOR COMMANDS
  - DEVICE AND FILE CONTROL
  - OBTAIN STATUS
  - COMMUNICATION WITH HOST OPERATOR
CP-6 LEVEL 6 RBT/IRBT HANDLER STRUCTURE

FROM FEISYMB

GETREC

HASP OUT

SYMBOLNT

RECORDERD

2780 OUT

HASP INPUT BLOCKS

SYNCIOINT

HASP BLOCKS

2780 OUTPUT BLOCKS

HASP OUTPUT BLOCKS

RBT/O

2780 IN

INPUT RECORDS

2780 INPUT BLOCKS

FROM MESSAGE SERVICE SCHEDULER

SEND MESSAGE TO HOST

TO HOST
HOST RESIDENT COMMUNICATIONS SUPPORT

- ADMINISTRATIVE SERVICES
- INITIALIZATION AND RECOVERY
- LEVEL 6 CRASH SUPPORT
- WORKSTATION DEFINITION
- HOST LOGON
- COMMUNICATION RELATED I/O FUNCTIONS
  - L66 COUPLER HANDLER
  - INTERFACE TO READ/WRITE COMMANDS
  - INTERFACE TO LOGICAL CONNECTIONS (OPEN)
  - PROGRAM CONTROL OF TERMINAL MODES
CP-6 LANGUAGE PROCESSORS
(DEVELOPED AT LADC)

PL-6

BASIC

ANS FORTRAN

APL

TEXT
CP-6 LANGUAGE PROCESSORS
HIGHLIGHTS

FUNCTIONALITY > CP-V LANGUAGES

COMPATIBILITY WITH ANS

COMPATIBILITY WITH H.I.S UNIFICATION

COMMON RUN-TIME LIBRARY FOR APL, BASIC, FORTRAN

SHARED PROCESSORS
PL-6 OVERVIEW

- LANGUAGE REQUIREMENTS
- COMPILER HISTORY
- COMPILER DESIGN
- LANGUAGE DEFINITION
- CODE GENERATION
- EFFICIENT USE OF HARDWARE
- EFFICIENT ACCESS TO MONITOR SERVICES
<table>
<thead>
<tr>
<th>LANGUAGE DEFINITIONS</th>
<th>PL-6</th>
<th>PL-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDRESS RESOLUTION</td>
<td>COMPILATION</td>
<td>COMPILATION/RUNTIME</td>
</tr>
<tr>
<td>ASYNCHRONOUS EVENT PROCESSING</td>
<td>USER CONTROLLED</td>
<td>ON CONDITION CHAIN IN AUTOMATIC</td>
</tr>
<tr>
<td>AUTO ALLOCATION STRATEGY</td>
<td>USER CONTROLLED</td>
<td>NO CONTROL</td>
</tr>
<tr>
<td>EFFICIENCY FEATURES</td>
<td>PL-6</td>
<td>PL-1</td>
</tr>
<tr>
<td>----------------------------------------------------------</td>
<td>------------------------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>DEDICATED POINTER REGISTER</td>
<td>ASSUME CLAUSE</td>
<td>NONE</td>
</tr>
<tr>
<td>REGISTERS PRESERVED ACROSS CALLS</td>
<td>PRESERVED CLAUSE</td>
<td>SAVES ALL REGISTERS</td>
</tr>
<tr>
<td>REGISTER HISTORY CONTROL ON BASED ASSIGNMENTS</td>
<td>REMEMBERS REGISTER HISTORY SPOIL CLAUSE</td>
<td>FORGETS HISTORY</td>
</tr>
<tr>
<td>STRING FACILITY</td>
<td>EIS INSTRUCTION</td>
<td>ALTRET FUNCTION</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>CONCAT/INSERT</td>
<td>MLR</td>
<td>TRUNCATION</td>
</tr>
<tr>
<td>XLATE</td>
<td>MVT</td>
<td>TRUNCATION</td>
</tr>
<tr>
<td>INDEX</td>
<td>NONE</td>
<td>NOT FOUND</td>
</tr>
<tr>
<td>INDEX1/INDEX1R</td>
<td>SCM/SCMR</td>
<td>NOT FOUND</td>
</tr>
<tr>
<td>INDEX2/INDEX2R</td>
<td>SCD/SCDR</td>
<td>NOT FOUND</td>
</tr>
<tr>
<td>SEARCH/SEARCHR</td>
<td>TCT/TCTR</td>
<td>NOT FOUND</td>
</tr>
<tr>
<td>BINCHAR/BINXCHAR</td>
<td>BTD</td>
<td>--</td>
</tr>
<tr>
<td>CHARBIN/XCHARBIN</td>
<td>DTB</td>
<td>--</td>
</tr>
<tr>
<td>EDITSTR</td>
<td>MVE</td>
<td>--</td>
</tr>
<tr>
<td>EDITCHAR/</td>
<td>MVNE</td>
<td>--</td>
</tr>
</tbody>
</table>
MONITOR SERVICES

MACRO DEFINITIONS

ENTRY DEFINITIONS

VECTOR/SIZEV FUNCTIONS

EXAMPLE:

%MAC FPT_TRUNC (FPTN=FPT=TRUNC, STCLAS=STATIC, DCB=NIL);

DCL 1 FPT STCLASS DALIGNED,
2 P,
3 V_ BIT(72) DALIGNED INIT(VECTOR(FPTN.V)),
2 V DALIGNED,
3 DCB# UBIN(18) UNAL INIT(DCBNUM(DCB));

%MEND;

DCL M$TRUNC ENTRY(1) CONV(1, 14) ALTRET;

%FPT_TRUNC (DCB=M$SI);

CALL M$TRUNC(FPT_TRUNC) ALTRET(ERROR);
PL-6 EXECUTION/INPUT-OUTPUT FLOW

SOURCE

PRE-PROCESSOR

EXPANDED SOURCE

LEX

MIIL 1

SYNTAX

MIIL 2

SEMANTICS

MIIL 3

CODEGEN

LISTING

EXPANDED SOURCE

OBJECT UNIT

LO/DM/ DIAGNOSTIC LISTING

EXPANDED SOURCE LISTING
PL-6 OVERLAY STRUCTURE

```
ROOT (1.5K)
   |-------------------|
   | CONTROL CARD      | (0.5K)
   |                   |
   | PRE-PROCESSOR     | (18K)
   |                   |
   | OBJECT CODE ROUTINES | (19K)
   |                   |
   |                   |
   | LEX/SYNTAX        | (12K)
   | SEMANTICS         | (32K)
   | CODE GENERATION   | (37K)
   |                   |
   | LISTINGS          | (16K)
```
BASIC

- SHORT REVIEW OF BASIC LANGUAGE FEATURES
- DESIGN GOALS
- DESIGN TECHNIQUES
- DESIGN OVERVIEW
- WHERE ARE WE NOW?
BASIC DESIGN GOALS

- RELIABILITY
- MAINTAINABILITY
- EXTENDIBILITY
BASIC DESIGN TECHNIQUES

- TOP-DOWN DESIGN

- FUNCTIONAL FLOWS-FLOWCHARTS

- MODULARITY

- STRUCTURE
BASIC PROCESSOR DESIGN OVERVIEW

- THREE MAJOR MODULES
  - EDITOR
  - COMPILER
  - RUNTIME

- THREE MINOR MODULES
  - GENERAL UTILITIES
  - MEMORY MANAGEMENT
  - CP-6 INTERFACE
INPUTS

- BASIC Commands
- BASIC Statements

PROCESSING

Editor

1. Perform Command
2. Invoke Compiler

Compiler

1. Analyse Statement
2. Generate Code
3. Execute Program
4. Return to Editor

Runtime

1. Input/Output
2. Math Functions
3. Matrix Functions
4. String Functions
5. Miscellaneous

OUTPUTS

BASIC Statement
User Response
Error Messages

L66 Machine Code
Error Messages

CP-6 Files
Error Messages
Terminal I/O
Various Math/Matrix & String Results

CP-6 Files
Binary
ASCII
Arguments
Common Data

Data Flow
Control Flow
INPUTS
- Monitor Call With Parameters
- CP-6 File Record
  - Terminal Record
  - Card Reader Record
- BASIC Command
- BASIC Statement

PROCESSING
1. Initialize BASIC
   - Determine Mode (Batch or Online)
2. Invoke Input Handler
3. Request specified Trap Control

OUTPUTS
- Flags, Pointers, Tables, Trap Control
- BASIC Command
- BASIC Statement
- Error Message
- User Response
- BASIC Statement

Command Handlers
1. Interpret Command
2. Issue Errors
3. Perform Command

Statement Handler
1. Set mode for Direct, Syntax check or compile?
2. Invoke Compiler

BASIC EDITOR - HIPO
BASIC COMPILER - HIPO
INPUTS

Various Numeric or String Arguments

PROCESSING

Math Functions
1. Algebraic
2. Trigonometric

String Functions
1. Conversions
2. Management

Matrix Operations
1. Mathematic
2. Input/Output

Output/Output
1. Terminal
2. CP-6 Files
3. Peripherals

OUTPUTS

1. Numeric/String Results
2. Error Messages

1. CP-6 Files
2. Terminal Messages
3. Printer Files

BASIC RUNTIME - HIPO
WHERE ARE WE NOW?

- EPS-1 RELEASED
- PRELIMINARY DESIGN COMPLETE
- DETAIL DESIGN ON SCHEDULE NEAR 20% COMPLETE
- ONE MINOR MODULE CODED/CHECKED OUT
<table>
<thead>
<tr>
<th>EVENTS</th>
<th>1977</th>
<th>1978</th>
<th>1979</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAMJ</td>
<td>JASON</td>
<td>JFMAMJ</td>
</tr>
<tr>
<td>1. PRELIMINARY BASIC DESIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. DETAIL BASIC DESIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. MODULE IMPLEMENTATION</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a. UTILITIES</td>
<td>▼</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. BASIC KERNEL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. EDITOR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d. FRONTLE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ANS FORTRAN FOR CP-6

LANGUAGE FEATURES

- 77 ANS FORTRAN LANGUAGE

- IBM FORTRAN LANGUAGE EXTENSIONS

- XEROX CP-V ANS FORTRAN EXTENSIONS
IBM FORTRAN LANGUAGE EXTENSIONS

- DIRECT ACCESS I/O
  - DEFINE FILE STATEMENT
  - FIND STATEMENT
  - INDEXED READS AND WRITES

- NAMELIST DRIVEN I/O
  - NAMELIST STATEMENT (WITH N/L NAMES)
  - READS AND WRITES WITH NAMELIST NAMES

- ALTERNATE RETURNS FROM SUBPROGRAMS
XEROX CP-V ANS FORTRAN LANGUAGE EXTENSIONS

- ABNORMAL STATEMENT
- CONSTANT STATEMENT
- DOUBLE COMPLEX DATA TYPE
- GLOBAL STATEMENT
- INCLUDE STATEMENT
- INPUT STATEMENT
  - FREE FORM
  - WITH LIST
- NAMELIST STATEMENT
  - WITHOUT ID LIST
  - WITH ID LIST, WITHOUT N/L NAMES
- NORMAL STATEMENT
- OUTPUT STATEMENT
- PARAMETER STATEMENT (INIVAC FORM)
- READ/WRITE DISK STATEMENTS
- VIRTUAL STATEMENT
- ENCODE STATEMENT
- DECODE STATEMENT
ANS FORTRAN COMPILER DESCRIPTION

- Design based of CP-V ANS FORTRAN
  - Similar language set
  - Block optimization
  - Basic design is transportable
  - Familiarity of design

- Multipass compiler

- Overlayed processor
OVERLAY STRUCTURE OF THE
CP-6 ANS FORTRAN COMPILER

ROOT
PROGRAM CONTROL

SYNTAX ANALYSIS - SOURCE EDITOR

DECLARATION PROCESS - ALLOCATION PROCESSING

SEMANTIC ANALYSIS AND PLEX TREE BUILDING

OPTIMIZATION - TRANSLATION

CODE GENERATION PHASE

LOAD AND GO CODE EXECUTION
COMPILER OPTIMIZATION

- BLOCK OPTIMIZATION

- CONGRUENT SUBSCRIPTS EXPRESSIONS

- CONGRUENT ARITHMETIC EXPRESSIONS

- ALTERNATE RECOGNITION
  - VARIABLES KNOWN AS EXPRESSIONS
  - VARIABLES KNOWN AS CONSTANTS
  - EXPRESSIONS KNOWN AS VARIABLES

- COMPILTIME CONSTANT ARITHMETIC
  - ADD, SUB, MULTIPLE, DIVIDE, EXPONENTIATION

- CONGRUENT ADDRESS REGISTER USAGE
ALTERNATE RECOGNITION

- VARIABLES KNOWN AS EXPRESSIONS

ASSOCIATION OF A VARIABLE WITH AN EXPRESSION:

\[
A = B \times C
\]

\[
D = A \quad [.:, D=\text{RESULT OF } B \times C]
\]

- VARIABLES KNOWN AS CONSTANTS

ASSOCIATION OF A VARIABLE WITH A CONSTANT AT COMPILE TIME:

\[
A = 10.0
\]

\[
D = A \times 4 \quad [.:, D=40.0]
\]

- EXPRESSIONS KNOWN AS VARIABLES

ASSOCIATION OF AN EXPRESSION WITH A VARIABLE:

\[
A = B \times C
\]

\[
D = B \times C \quad [.:, D=A]
\]
FRONT END PROCESSOR

BATCH & TERMINAL

PROGRAM SOURCE

TERMINAL

EDITING COMMANDS

ENCODER:
SYNTAX ANALYSIS

EDIT COMMAND

SOURCE EDITING

NEW SOURCE FILE OF EDITED SOURCE

EDITED COMMAND

LEXICAL ITEMS STREAM

DICTIONARY OF IDS AND LABELS

SEMANTICAL TABLE FOR IDS AND LABELS (SKELETON)
SEMANTICAL PHASE PROCESSING

SEMANTIC TABLE (SKELETON)

DECLARATION STATEMENT PROCESSING, LABEL USAGE CHECKING, FORMAT STATEMENT ENCODING

LEXICAL ITEMS STREAM

ALLOCATION PROCESSING: EQUIVALENCE AND COMMON STATEMENTS

EXECUTABLE STATEMENT ANALYSIS: TREE BUILD

PLEX (TRIAD) TREE

UPATED SEMANTICAL TABLE

STATEMENT PROCESSING LABEL

LABEL USAGE CHECKING

FORMAT STATEMENT ENCODING

STATEMENT PROCESSING

ALLOCATION PROCESSING

EXECUTABLE STATEMENT ANALYSIS

TREE BUILD

FURTHER UPDATED SEMANTICAL TABLE

M.M. UPDATED SEMANTICAL TABLE
CODE GENERATION PROCESSING

LOAD AND GO

INSTRUCTIONS IN MEMORY

O.U. GENERATION

O.U. MODULE

M.M. UPDATED SEMANTICAL TABLE

OPTIMIZATION PHASE

PLEX (TRIAD) TREE

TRANSLATOR

OPTIMIZED PLEX (TRIAD) TREE

(ALSCMLER LEVEL CODE)

CODE GENERATION

ALC (OPERAND, OPERATOR)
COMPETITIVE ADVANTAGES OF CP-6 APL

1. A rich set of functions exceeding those of competitive products.

2. An environment which admits expansion easily.

3. Relatively efficient operations on large data aggregates.

4. Workspace capacity significantly greater than that of competitive products.
DIFFERENCES BETWEEN CP-V APL & CP-6 APL

1. Hardware differences

2. Implementation vehicle differences

3. Operating system differences

4. New features

5. Delayed features
**PROCESOR STRUCTURE**

- INIT 500
- FUNCS 5500
- FILE 6200
- CMOS 6400
- DEF 5500
- MAT 2600
- SPROG 6200

---

**Name** | **Includes**
---|---
ROOT | Principal monitor interface, input, output, workspace management, codestring, codestring executor, operator execution drivers
INIT | Processor initialization
FUNCS | Certain function evaluators, formatted output
FILE | File I/O and shared variables
CMDS | All commands
DEF | Function definition & error reporting
MAT | The domino operator
SPROG | System programmer functions: workspace management, text editing, and canonical representation

Sizes above are in 36 bit words and have a tolerance of ±20%, for procedure only.
WORKSPACE DATA

STATIC DATA - 3 pages in the ROOT and 1 page in most overlays

AUTOMATIC DATA - 1 page

DYNAMIC SEGMENT 1 - The symbol table and the data blocks

DYNAMIC SEGMENT 2 - The execution stack
INTERNALLY PERCEIVED RISKS

1. THE IMPLEMENTATION VEHICLE, PL6

2. THE CHECKOUT ENVIRONMENT AND LIMITED CHECKOUT TIME

3. PERSONNEL
Competitive Advantages of CP-6 APL

I. A rich set of functions exceeding those of many competitive products.
   A. File I/O, blind I/O
      1) The full CP-6 file system facilities are available to the APL user. This includes shared update as well as enqueue/dequeue to control access in the shared mode.
      2) Both translated and untranslated(blind) I/O is provided.
   B. Several unique commands are provided
      1) }SEAL allows the installation to provide proprietary APL functions for execution only. Also, individual functions in a workspace may be locked to likewise prevent inspection of the algorithm.
      2) }OBSERVE provides the ability to inspect the evaluations undertaken for execution of any line in the minutest detail.
      3) }CATCH allows the user to intercept all assignments to a particular variable to facilitate debugging.
   C. System commands may be executed via the unary epsilon operator.
   D. Sidetracking of errors and breaks is provided.
   E. The system programmer functions
      1) Canonical representation to change function definition to and from text.
      2) Workspace management to investigate attributes of workspace constituents.
      3) Text editing to provide text index, search and replacement and comparison.

II. An environment which admits expansion easily.
   A. Both batch and on-line operations are accommodated.
   B. A very flexible terminal interface mechanism admits considerable variety.
      1) Files are included.
      2) Terminals without APL characters are accommodated.
   C. The PL6 software factory has several beneficial aspects.
      1) Listings can contain much more documentary content.
      2) The language promotes the use of "structured" constructs.
   D. The modularization of the CP-6 APL product will separate data, procedure and monitor interface.
   E. The commentary in the source for the processor will contain all the information to produce the technical and data base manuals via an automated process.
Competitive Advantages of CP-6 APL - Contd.

III. Relatively efficient operations on large data aggregates.
   A. Code is "compiled" for inner loops.
   B. Use is made of the common library at all radix translation and elementary function evaluation times.
   C. Codestring is used to facilitate execution.

IV. Workspace capacity is significantly greater than that of the competition.
   A. A large virtual memory is provided with little operating system preemption.
   B. Multiple segment dynamic memory will be utilized.
   C. Typically CP-6 APL will provide 256K bytes vs. 32K bytes.
   D. CP-6 APL will provide overflow symbol table processing beyond the "hashed" symbol table capacity.
Differences between CP-V APL & CP-6 APL

It should be emphasized that the CP-V APL design is, to a large extent, being copied over into the APL processor for CP-6. CP-V APL is a field-proven, highly competitive product. However, there are differences which should be exposed during review.

I. Differences caused by change of hardware
   A. Word and byte size, but fortunately 36>32 and 9>8.
   B. Floating point range is reduced which may require scaling of some processes.
   C. The instruction set and the CPU organization are quite different.
      1) The compiled code for interpretation of operators is new.
      2) The use of PL6 minimizes much of this effect.
   D. Traps are different.

II. Differences caused by change of primary implementation vehicle.
   A. The "structured" and documentary effects have been reviewed earlier.
   B. PL6 generated code will be far from that which would be generated to provide minimum procedure residency as was the intent in the CP-V product.
   C. The ability to produce technical documentation in an automated manner was mentioned earlier.
   D. As an emphasis note, all of the above noted differences tend to promote maintainability.

III. Differences caused by change of operating system.
   A. Some additional effects of hardware change are reflected in the operating system interface.
      1) Exceptional conditions vary.
      2) NSA hardware is different from the Sigma map.
      3) File granule size is different.
   B. Many of the facilities of CP-6 expand upon those which had been in CP-V.
      1) File open
      2) File attributes
      3) "Privilege"
   C. Dynamic memory management is particularly different.
Differences between CP-V APL & CP-6 APL - Contd.

IV. New features have been added to CP-6 APL.
   A. Distinguished names
      1) Had been available as I-BEAM and T-BAR functions.
      2) Must be made capable of localization.
   B. Shared variables are a new CP-6 APL facility which will be provided by utilizing field-proven CP-V facilities.
      1) Shared update files
      2) Enqueue and dequeue

V. Delayed features - Two features of the CP-V APL processor have been eliminated from the initial release of the CP-6 product.
   A. Graphics functions
   B. The interface to the data base manager, IDS-II.
Processor Structure

I. ROOT
   A. Principal monitor interfaces
      1) Memory management
      2) SI, LO, DO management
      3) Exceptional condition handlers
   B. Input
      1) Mnemonic translation
      2) Visual fidelity
   C. Output
      1) Mnemonic translation
      2) Line formation
   D. Workspace management
      1) Symbol table
      2) Data blocks
      3) Execution stack
   E. Command recognition
      1) Initial processing of commands
      2) Root interface to error processes
   F. Codestring
      1) Translate to/from codestring
      2) Codestring execution
   G. Operator execution drivers
      1) Simple operators
      2) Mixed and composite operators
      3) Index processes
      4) Intrinsic functions

II. INIT - Once per user processor invocation processes.

III. FUNCS
   A. Certain function evaluators
   B. Formatted output
Processor Structure - Contd.

IV. FILE
   A. File and transparent I/O
   B. Shared variables

V. CMDS - All commands starting with 

VI. FUNDEF
   A. Function definition
   B. Error management

VII. MAT - Matrix inversion

VIII. SPROG - System programmer functions; namely, canonical representation, workspace management, and text editing.
Workspace Data

I. Static data - 3 pages for the ROOT and 1 page in several overlays. This is the main internal working space for the workspace containing the fixed size data which is, in general, not displayable.

II. Automatic data - 1 page is required for certain local data which is temporary in nature.

III. Dynamic segment 1
   A. Most of the displayable and copyable fixed length workspace data.
   B. The symbol table.
   C. The data blocks.

IV. Dynamic segment 2
   A. The fixed length state control data.
   B. The execution stack.
Internally Perceived Risks

I. The implementation vehicle, PL6
   A. Effects on processor and workspace size.
   B. Effects on speed of various processes.

II. Personnel - It is an exceptional circumstance which could cause this to be surfaced as a risk, but the health of the only team member from the original CP-V team is of concern.

III. Checkout
   A. The novelty of CP6 is bound to cause problems, not the least of which is concern about timely availability.
   B. The current schedule does not admit sufficient time between earliest assumed availability of checkout hardware/software and delivery time to accommodate even the most optimistic checkout schedule.
CP-6 TEXT

SHARED PROCESSOR UNDER CP-6

FAST DOCUMENT CREATION

AUTOMATIC FORMATTING CAPABILITIES

NAME-AND-ADDRESS FILES

ON-LINE AND BATCH OPERATION
INPUTS

- Build Document
  - Secretary types document

- Edit Document
  - Secretary types changes

- Print Document
  - Secretary types commands

PROCESSING

- Editor
  - Stores text
  - 1 line = 1 record
  - Reads record
  - Updates record

- Formatter
  - Reads file
  - Interprets commands
  - Formats document

OUTPUTS

- Creates file
- Writes updated file
- Prints document
CP-6 LANGUAGE PROCESSORS

SUMMARY

DESIGN PHASE - EXCELLENT VISIBILITY

BUILDING UPON PROVEN CP-V DESIGN

CODING PHASE - HIGH CONFIDENCE

CHECKOUT PHASE - LOTS OF EXPOSURE
CP-6 LANGUAGE PROCESSORS
RISKS

CONCURRENT DEVELOPMENT OF OPERATING SYSTEM
DEBUGGING - WHERE IS ERROR?
CONTENTION FOR COMPUTER TIME

COMPUTER RESOURCES
ADEQUATE COMPUTER TIME
SUFFICIENT TERMINALS

PL-6 PERFORMANCE
SIZE
SPEED

STAFFING
CP-6 PRELIMINARY DESIGN REVIEW
PHOENIX SUPPLIED SOFTWARE

- PRODUCTS
  - COBOL-74
  - SORT/MERGE

- IMPLEMENTATION TOOLS
  - PL/I
  - GMAP6

- TOPICS
  - PRODUCT STRUCTURE
  - COMPATIBILITY
  - PERFORMANCE
  - TESTING
  - SCHEDULE
  - MAINTENANCE STRATEGY
PRODUCT STRUCTURE

- BASE DESIGN
  - EXISTING GCOS PRODUCTS

- DEVELOPMENT STRATEGY
  - MAXIMIZE COMMON SOURCE BASE
  - ISOLATE HOST SYSTEM INTERFACES
  - ADAPT TO CP-6 I/O

- EXCEPTION - SORT/MERGE
  - MAJOR REDEVELOPMENT
  - PERFORMANCE SENSITIVITY
  - INTIMATE USE OF MONITOR SERVICES
COMPATIBILITY

• WITH CP-V
  • MAXIMIZE CONSISTENT WITH OVERALL CP-6 GOALS

• WITH GCOS
  • HIGHLY DESIRABLE TO FACILITATE:
    • COMMON PROCESSORS - COSTS
    • ULTIMATE MIGRATION

• WITH IBM
  • NOT Addressed IN WORK LEADING TO RELEASE 1
PERFORMANCE

• EXPECTATIONS
  • EXISTING DESIGN
  • EXISTING BASE PROCESSOR

• EXCEPTIONS
  • MONITOR I/O
  • COMMON I/O ADAPTATION
  • PURE PROCEDURE
TESTING

- BY DEVELOPMENT SHOP LEADING TO INITIAL RELEASE AS OPPOSED TO CURRENT GCOS PRACTICE OF USING SEPARATE AND FORMALIZED TEST ORGANIZATION

MAINTENANCE STRATEGY

- CP-6 FIELD SUPPORT GROUP
  - SCREEN STARS
  - RESPONSE DATABASE
  - INTERFACES WITH THE FIELD AND THE DEVELOPMENT GROUPS

- PHOENIX DEVELOPMENT GROUP
  - INTERFACES WITH CP-6 FIELD SUPPORT GROUP
  - COMMON PROCESSOR FIXES TESTED ON GCOS
  - HOST INTERFACE FIXES TESTED ON CP-6
CP-6 PRELIMINARY DESIGN REVIEW

COBOL-74

- STRUCTURE
- COMPATIBILITY
  - CP-5
  - GCOS-III
- PERFORMANCE
- TESTING
- SCHEDULE & MILESTONES
- MAINTENANCE STRATEGY

PL/I

- STRUCTURE
- SCHEDULE
STRUCTURE
COBOL-74 COMPILER EXECUTIVE

- CONTROLS FLOW OF COMPILER FROM PHASE TO PHASE
- PROVIDES ABORT RECOVERY AND WRAP-UP
- PROVIDES I/O INTERFACE WITH CP-6
- PROVIDES CP-6 MONITOR INTERFACES
- PROVIDES OTHER MISCELLANEOUS SUPPORT FUNCTIONS
COBOL-74 GENERATOR PHASE

TRANSLATION OF INTERNAL LANGUAGE LIST (ILL) AND EXPRESSION TRIAD TABLE (ETT) ITEMS INTO OBJECT CODE.

• INPUT
  • INTERMEDIATE FILE: ILL FROM ANALYZER
  • CORE RESIDENT TABLES: ETT, ST, LTS, SYSSM, ETC.

• OUTPUT
  • CP-6 OBJECT FILE
  • *1 FILE (LSTOU INFORMATION)
COBOL-74 ANNOTATOR

- PRODUCES SOURCE LISTING OF COBOL COMPILER
- PRODUCES SYMBOLIC LISTING OF GENERATED OBJECT CODE
- PRODUCES SYMBOL CROSS REFERENCES REPORT
- PRODUCES STORAGE MAP
- PRODUCES OBJECT DEBUG SCHEMA
- PRODUCES COMPILER STATISTICS
COMPATIBILITY

- CP-5
  - FLOATING POINT DATA TYPES
  - MULTIPLE ENTRY POINTS
  - LABEL PROCESSING USE PROCEDURES
  - RUNTIME OPTIONS
  - 30 CHARACTER ID- NAMES

- GCOS-III
  - INCORPORATION OF CP-5 ENHANCEMENTS
  - SINGLE COMPILER SOURCE
PERFORMANCE

- COMPILER
  - SIZE  42K BASE
  - SPEED  3000 LPM
  - NOT SHARED

- OBJECT
  - EFFICIENCY
  - SIZE
  - SHAREABLE
LANGUAGE FEATURE TESTING

- USE EXISTING TEST PROCEDURES
- TEST IN LEVEL 66 ENVIRONMENT

CP-6 OBJECT TESTING

- USE EXISTING TEST PROGRAMS
- COMPILE ON LEVEL 66; EXECUTE ON CP-6
- REQUIRES EXTENSIVE MODIFICATION OF TEST PROCEDURES

COMPILER TESTING

- USE EXISTING TEST PROGRAMS, TRANSPORTED TO CP-6 IN SOURCE FORM
- USE EXISTING (SELF-CHECKING) TEST PROCEDURES MODIFIED TO EXECUTE ON CP-6
## SCHEDULE & MILESTONES

<table>
<thead>
<tr>
<th>ITEM</th>
<th>START</th>
<th>COMPLETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUNCTIONAL ENHANCEMENTS</td>
<td>732</td>
<td>834</td>
</tr>
<tr>
<td>FLOATING POINT DATA TYPE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MULTIPLE ENTRY POINTS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LABEL PROCESSING USE PROCEDURES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP-6 OBJECT I/O</td>
<td>733</td>
<td>827</td>
</tr>
<tr>
<td>CP-6 RUNTIME LIBRARY</td>
<td>733</td>
<td>826</td>
</tr>
<tr>
<td>CP-6 OBJECT UNIT</td>
<td>730</td>
<td>830</td>
</tr>
<tr>
<td>DEBUG SCHEMA</td>
<td>728</td>
<td>832</td>
</tr>
<tr>
<td>COMPILATION TIME MODIFICATIONS</td>
<td>730</td>
<td>830</td>
</tr>
<tr>
<td>COMPILER INTEGRATION</td>
<td>826</td>
<td>839</td>
</tr>
<tr>
<td>SYSTEM TEST</td>
<td>840</td>
<td>913</td>
</tr>
</tbody>
</table>
MAINTENANCE STRATEGY

INTERFACE - LADC

- RESPONSES/VERIFICATION
- TESTING ON GCOS-III (LANGUAGE)
- TESTING ON CP-6
- CORRECTIONS ISSUED

OBJECTIVE - SINGLE COMPILER SOURCE

- CP-6 I/O INTERFACE
- CP-6 MONITOR INTERFACE
- CP-6 SORT INTERFACE
- CP-6 IDS-II INTERFACE
- CP-6 OBJECT UNITS
- CP-6 RUNTIME LIBRARY
- CP-6 DEBUG SCHEMA
PL/I

- IMPLEMENTATION LANGUAGE (SUBSET)
  - COBOL-74
  - IDS-II

- CROSS COMPILER ON GCOS-III

- EXPORT FACILITY TO CP-6 (OBJECT)

- PL/I RUNTIME ON CP-6
PL/I STRUCTURE
PL/I SCHEDULE

GCOS-III PL/I

- CURRENT STATUS
  RECOMPILED COBOL-74
  RECOMPILING ITSELF

- FIELD TEST FW744
- PRODUCT PL/I FW752

CP-6 PL/I CROSS COMPILER

- IMPLEMENTATION FW747-FW831
- SYSTEM TEST FW832-FW843
- AVAILABILITY (COBOL-74) FW835
- TWO ADDITIONAL PEOPLE FW826

CP-6 PL/I COMPILER

- AVAILABLE FW852
CP-6 PRELIMINARY DESIGN REVIEW

SORT/MERGE

- PRODUCT STRUCTURE
- CP-V COMPATIBILITY
- PERFORMANCE
- TESTING
- SCHEDULE
- MAINTENANCE STRATEGY
SORT/MERGE PRODUCT STRUCTURE

- USES BOTH CP-V AND GCOS III ALGORITHMS
  - REPLACEMENT/SELECTION SORT TOURNAMENT
  - BOTH POLYPHASE AND STANDBY TAPE SORT
  - MINIMUM TREE DISK SORT
- USES CP-6 MONITOR FOR INPUT/OUTPUT
- IMPLEMENTED IN COMBINATION OF PL/6 AND ASSEMBLY LANGUAGE
- IMPLEMENTATION AND DOCUMENTATION VIA CP-6 SOFTWARE FACTORY ON CP-V
SORT/MERGE PRODUCT STRUCTURE

• COMBINATION OF
  • LINKABLE SUBROUTINES
  • SHARED SUBROUTINE LIBRARY MODULES

• LINKABLE SUBROUTINES
  • COBOL-74/SORT INTERFACE
  • COBOL-74/MERGE INTERFACE
  • SORT FREESTANDING DRIVER
  • MERGE FREESTANDING DRIVER

• SHARED SUBROUTINE LIBRARY MODULES
  • PARAMETER VALIDATION
  • KEY COMPARISON COMPILER
  • SORT TOURNAMENT DRIVER
  • SORT COLLATION DRIVER
  • SORT COLLATION INPUT ROUTINE
  • SORT COLLATION OUTPUT ROUTINE
  • SORT STRING DISTRIBUTION MANAGER
  • SORT STRING COMBINATION MANAGER
  • MERGE COLLATION DRIVER
COBOL-74 SORT/MERGE

USER DATA SPACE  |  SHARED LIBRARY SPACE  |  COBOL-74 OBJECT-UNIT  |  SORT (MERGE) INTERFACE  |  SHARED SUBROUTINE LIBRARY

"CO-RESIDENT" SORT/MERGE

USER DATA SPACE  |  SHARED LIBRARY SPACE  |  USER ROUTINES  |  SORT (MERGE) DRIVER  |  SHARED SUBROUTINE LIBRARY

FREESTANDING SORT/MERGE

DATA SPACE  |  COMMON SPACE*  |  SHARED LIBRARY SPACE  |  SORT (MERGE) DRIVER  |  SHARED SUBROUTINE LIBRARY

*WHEN INVOKED VIA M$LINK
SORT/MERGE COMPATIBILITY

- HIGH CP-V COMPATIBILITY IN
  - KEY DESCRIPTION AND TRANSLATION
  - RECORD DESCRIPTION
  - FILE POSITIONING
  - USER OWN-CODE INTERFACES

- FULL COMPATIBILITY WITH COBOL-74

- WILL EXECUTE IN EITHER BATCH OR TIMESHARING MODE

- FULL COMPATIBILITY WITH ANS FORMAT TAPES AND ALL CP-6 FILE TYPES

- DIFFERS FROM CP-V IN
  - BLOCK DIRECTIVE DELETED IN FAVOR OF LABEL INFORMATION
  - LIMIT DIRECTIVE MODIFIED
  - NEW DATA TYPES ADDED FOR COBOL-74

JAW
770914
SORT/MERGE PERFORMANCE

- GOALS
  - PERFORMANCE EQUIVALENT TO GCOS III
  - DISK SORT SPACE NOT TO EXCEED 1.2 X INPUT FILE SIZE

- FINAL PERFORMANCE HIGHLY DEPENDENT ON MONITOR I/O PERFORMANCE
SORT/MERGE TESTING

- PLAN TO USE SORT/MERGE SYSTEM AS OPERATING SYSTEM TEST VEHICLE

- PLAN EXTENDED UNIT AND SYSTEM TESTING TO REFINE PERFORMANCE

- PLAN EXTENDED COBOL-74 SORT/MERGE TESTING VIA FEDERAL AUDIT ROUTINES
SORT/MERGE SCHEDULE

<table>
<thead>
<tr>
<th>System Design</th>
<th>1977</th>
<th>1978</th>
<th>1979</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ASOND</td>
<td>JFMAM</td>
<td>JASOND</td>
</tr>
</tbody>
</table>

- Publish EPS-1
- Sort Module Design
- Merge Module Design
- Publish EPS-2
- Draft Reference Manual
- Sort Implementation
- Merge Implementation
- Sort Unit Test
- Merge Unit Test
- Factory to Native CP-6
- System Test
- Packaging
- Delivery

JAW
770914
SORT/MERGE MAINTENANCE STRATEGY

MAINTAINED ON CP-6 SYSTEM.
CP-6 PRELIMINARY DESIGN REVIEW

GMAP6

- PROCESSOR STRUCTURE
- COMPATIBILITY
- PERFORMANCE
- TESTING
- SCHEDULE
- MAINTENANCE STRATEGY
GMAP6 STRUCTURE
FILE UTILIZATION
GMAP6 STRUCTURE
STORAGE UTILIZATION

PASS 0
HOST INTERFACE Routines
ALTER SORT
PASS 1
PASS 2
SCHEMA GENERATOR
DATA SPACE
30K

GAM
770914
GMAP6 COMPATIBILITY

- GCOS66
  - COMMON PROCESSOR/COMMON LANGUAGE
  - SYSTEM PERSONALITY Macros DIFFER
  - ALTER MECHANISM MAY DIFFER
  - LOGICALLY SIMILAR OBJECT UNITS

- CP-V
  - NONE

- GCOS-III
  - LARGELY COMMON LANGUAGE
  - GMAP6 ADDS SECTION/SEGMENT VISIBILITY
  - SOME DIFFERENCES IN PSEUDO-INSTRUCTIONS
  - DIFFERENT OBJECT OUTPUT FORMATS
GMAP6 PERFORMANCE

- TARGET IS PARITY WITH GCOS EXECUTION OF GMAP AND APPROXIMATELY SAME CORE REQUIREMENTS.

- GMAP ON GCOS VARIES (DEPENDING ON MACRO USE) FROM 5000 - 20000 SOURCE STATEMENTS PER MINUTE OF PROCESSOR TIME ON A 66/80.

- ACTUAL PERFORMANCE MAY VARY FROM THIS TARGET AS A FUNCTION OF CP-6 I/O. HOWEVER, NO PERFORMANCE PROBLEMS ENVISIONED AT THIS TIME.

- GMAP6 IS NOT SHARED IN RELEASE 1.
GMAP6 TESTING

- IMPLEMENTATION VERSIONS
  - LIMITED SPECIFIC TESTING FOR NEW FEATURES
  - LIMITED GENERAL TESTING
  - ACTUAL USE IN DEVELOPMENT OF CP-6 AND LANGUAGE SYSTEM SUPPORT ROUTINES

- FINAL VERSION
  - SELF ASSEMBLED
  - LIMITED GENERAL TESTING
  - ACTUAL USE IN ASSEMBLY OF ALL SYSTEM TEST VERSIONS OF CP-6 AND ITS PRODUCT SOFTWARE

GAM
770914
GMAP6
DEVELOPMENT SCHEDULE
MILESTONES

• PHASE 1  FW 748
  • PROVIDE SECTION DECLARATION
  • SYMDEF/SYMREF ENHANCEMENTS
  • PROVIDE FOR SYSTEM MACRO SUBSTITUTION
  • PRODUCE EXPORT OBJECT UNIT
  • RUN UNDER GCOS III
  • INTERNAL RELEASE ONLY

• PHASE 2  FW 813
  • PRODUCE PROGRAM SCHEMA
  • ENHANCED RELOCATION FLEXIBILITY
  • STILL RUNS UNDER GCOS III
  • STILL PRODUCES EXPORT OBJECT UNIT
  • INTERNAL RELEASE ONLY

• BEGIN NATIVE TEST  FW 826
  • RUNS UNDER CP6
  • NO OBJECT UNIT OUTPUT
  • TEST VEHICLE ONLY

• BEGIN SYSTEM TEST  FW 839
  • MERGE GMAP66 (ADF2V) FEATURES
  • COMMON PROCESSOR WITH GCOS66
  • CAPABLE OF ASSEMBLING SELF
  • PRODUCES CP6 OBJECT UNIT

• SHIP  FW 913
GMAP6
MAINTENANCE STRATEGY

- COMMON PROCESSOR WITH GMAP66
  - SINGLE SOURCE BASE
  - SINGLE PROJECT TEAM

- SPECIALIZED HOST SYSTEM INTERFACE
  - ISOLATES CP-6 INTERFACES

- PROCESSOR PACKAGING
  - INITIALLY DELIVERED AS EXPORT OBJECT UNITS TO BE CONVERTED AND LINKED ON CP-6 FOR SYSTEM TEST
  - FINAL PRODUCT DELIVERED AS CP-6 OBJECT UNITS
CP-6 PRELIMINARY DESIGN REVIEW
PHOENIX SUPPLIED SOFTWARE

• SUMMARY
  • BASED ON EXISTING PRODUCTS
  • MAXIMIZE COMMON SOURCE
  • HIGH DEGREE OF COMPATIBILITY WITH CORRESPONDING GCOS PRODUCTS
  • PERFORMANCE TARGET - PARITY WITH GCOS
I-D-S/II FOR CP-6

- A CODASYL STRUCTURED DATA BASE SYSTEM
- TRANSLATORS CONVERT DDL AND DMCL \rightarrow\ SCHEMA
- TRANSLATORS CONVERT DDL \rightarrow\ SUBSCHEMA
- GCOS III I-D-S/II PLUS
  - INCREASED SECURITY
  - FIXED 1024 WORD PAGE SIZE
  - SHARED DBCS PROCEDURE (ASL)
  - MULTI-LANGUAGE INTERFACE
  - IMPORTANT EDMS FEATURES
  - ON-LINE TRANSLATION OF DDL, DMCL
I-D-S/II

MOTIVATING FACTORS

• NEW HARDWARE - NSA, WORKSPACES, ASL
  - SHARED PROCEDURE IN DBCS

• CP-6 AND GCOS-III SYSTEM DIFFERENCES
  - CONCURRENT ACCESS CONTROL
  - BUFFER MANAGEMENT
  - JOURNALING
  - FILE I/O
  - COMMAND LANGUAGE

• XEROX PARC - NEW (OLD) FEATURES
I-D-S/II

MAJOR EDMS & I-D-S/II DIFFERENCES

• NO IDS INDEXED RECORD SET RELATIONSHIPS
• NO IDS LOGICAL RESTRUCTURING PACKAGE
• DIFFERENT DATA TYPES
I-D-S/II

NEW FEATURES FOR XEROX PARC

• RUN TIME TRACE
• RUN TIME STATISTICS
• MULTI LANGUAGE INTERFACE
• CHECKSUM OF INTEGRATED FILE PAGES
• ENCRYPTION OF INTEGRATED FILE PAGES
• ENCRYPTION OF COMMON JOURNAL
I-D-S/II

RESOLVING DIFFERENCES

NEW MANUALS

I-D-S/II

COBOL-74

GCOS-III

DB

EDMS

ANY LANGUAGE

CP-V

DB

ANY LANGUAGE

CP-6

Conversion

Conversion

JR 9/19/77
I-D-S/II

- PRODUCT DELIVERED TO US AS A GCOS III RELEASE

- STRUCTURE
  - DBCS 16,000 Lines GMAP
  - DBACS 32,000 Lines PL/1
  - UTILITIES 12,000 Lines GMAP

- DROPPING
  - INTERACTIVE I-D-S/II
  - PHYSICAL RESTRUCTURING UTILITY
  - DELETE UTILITY
I-D-S/II PERFORMANCE

- 64 µ SEC FOR EACH DBCS CLIMB
- SOME COMPUTE FOR COBOL-74 OIL SLICK
- EXPECT PERFORMANCE TO BE THE SAME AS GCOS III AND EDMS.
- CP-6 WILL HAVE SHARED DBCS PROCEDURE AND SHARED TRANSLATORS AND UTILITIES
I-D-S/II MAINTENANCE

- EACH NEW GCOS III I-D-S/II RELEASE WILL BE A PROBLEM TO BE CONSIDERED INDIVIDUALLY.

- ERRORS AND CUSTOMER PROBLEMS WILL BE SCANNED BEFORE ASKING FOR PHOENIX SUPPORT.

- ALL FIXES FROM PHOENIX WILL REQUIRE REVIEW BEFORE INSTALLATION.
I-D-S/II

TESTING AND VERIFICATION

- PHOENIX 4J TESTS
- CP-6 FUNCTION TESTS
### I-D-S/II

#### DEPENDENCIES

<table>
<thead>
<tr>
<th>Component</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-D-S/II DB3.0</td>
<td>OCT 77</td>
</tr>
<tr>
<td>ASSEMBLER</td>
<td>DEC 77</td>
</tr>
<tr>
<td>CP-6</td>
<td>APR 78</td>
</tr>
<tr>
<td>PL/1 FOR CP-6</td>
<td>SEP 78</td>
</tr>
<tr>
<td>COBOL FOR CP-6</td>
<td>DEC 78</td>
</tr>
<tr>
<td>DELIVERY</td>
<td>APR 79</td>
</tr>
</tbody>
</table>

*Two persons at present, one opening*
IDP FOR CP-6

- INTERACTIVE DATA BASE QUERY LANGUAGE
- TOPOLOGY ANALYSIS
- RETRIEVAL OF SEQUENTIAL DATA FILES
- AUTOMATIC OR SPECIFIED REPORT FORMAT
- REASONABLY SMALL (EST 15,000 LINES OF PL-6)
IDP STRUCTURE

- COMMAND MODULE
- VALIDATION MODULE
- RETRIEVAL MODULE
- DICTIONARY MODULE
- TOPOLOGY MODULE
- REPORT MODULE

RPG-II RUN TIME
IDP FLOW

- USER ENTERS IDP STATEMENTS OR RUNS A FILE OF IDP STATEMENTS FILLING IN BLANKS
- IDP ANALYZES TOPOLOGY
- IDP GENERATES PROGRAM IN MEMORY AND EXECUTES WITH RUN TIME LIBRARY
IDP

- PERFORMANCE - SAME AS IDP IN CP-V
- MAINTENANCE - ENTIRELY AT LADC
- TESTING - CP-V REGRESSION TESTS PLUS NEW TESTS
- DEPENDENCIES - CP-6, I-D-S/II, SORT
- SCHEDULE - 2Q79
- PRESENTLY ONE PERSON, ONE OPENING
RPGII FOR CP-6

- COMMERCIAL DATA PROCESSING
- PROGRAM PHASES
  - COMPILETION
  - EXECUTION
- COMPATIBILITY
LANGUAGE ELEMENTS

- FILE PROCESSING
- FORMS CONTROL
- OUTPUT EDITING
- OPERATIONS CODES
- FIXED SPECIFICATION FORMS
- FIXED PROGRAM LOGIC
1. SELECT RECORD TO BE PROCESSED BY MATCHING FIELD SEQUENCE
2. READ AND IDENTIFY A RECORD
3. RECORD SELECTION
4. IDENTIFY CONTROL FIELDS AND SET CONTROL LEVEL INDICATORS
5. TOTAL CALCULATIONS
6. TOTAL OUTPUT
7. OVERFLOW OUTPUT
8. SET MR INDICATOR TO SHOW MATCHING FIELD STATUS
9. MAKE DATA FROM SELECTED RECORD AVAILABLE
10. DETAIL CALCULATIONS

INITIALIZE

DETAIL TIME

TOTAL TIME

OUTPUT DETAIL CALCULATIONS

AVAILABLE SET MR INDICATOR TO SHOW MATCHING FIELD STATUS

READ AND IDENTIFY A RECORD

SELECT RECORD TO BE PROCESSED BY MATCHING FIELD SEQUENCE

IDENTIFY CONTROL FIELDS AND SET CONTROL LEVEL INDICATORS

TOTAL OUTPUT

DETAIL CALCULATIONS

MAKE DATA FROM SELECTED RECORD AVAILABLE

INITIALIZE

HEAD IN DETAIL OUTPUT

DETAIL TIME

TOTAL TIME

OUTPUT DETAIL CALCULATIONS

AVAILABLE SET MR INDICATOR TO SHOW MATCHING FIELD STATUS

READ AND IDENTIFY A RECORD

SELECT RECORD TO BE PROCESSED BY MATCHING FIELD SEQUENCE

IDENTIFY CONTROL FIELDS AND SET CONTROL LEVEL INDICATORS

TOTAL OUTPUT

DETAIL CALCULATIONS

MAKE DATA FROM SELECTED RECORD AVAILABLE

INITIALIZE

HEAD IN DETAIL OUTPUT

DETAIL TIME

TOTAL TIME

OUTPUT DETAIL CALCULATIONS

AVAILABLE SET MR INDICATOR TO SHOW MATCHING FIELD STATUS

READ AND IDENTIFY A RECORD

SELECT RECORD TO BE PROCESSED BY MATCHING FIELD SEQUENCE

IDENTIFY CONTROL FIELDS AND SET CONTROL LEVEL INDICATORS

TOTAL OUTPUT

DETAIL CALCULATIONS

MAKE DATA FROM SELECTED RECORD AVAILABLE

INITIALIZE

HEAD IN DETAIL OUTPUT

DETAIL TIME

TOTAL TIME

OUTPUT DETAIL CALCULATIONS

AVAILABLE SET MR INDICATOR TO SHOW MATCHING FIELD STATUS

READ AND IDENTIFY A RECORD

SELECT RECORD TO BE PROCESSED BY MATCHING FIELD SEQUENCE

IDENTIFY CONTROL FIELDS AND SET CONTROL LEVEL INDICATORS

TOTAL OUTPUT

DETAIL CALCULATIONS

MAKE DATA FROM SELECTED RECORD AVAILABLE

INITIALIZE

HEAD IN DETAIL OUTPUT

DETAIL TIME

TOTAL TIME

OUTPUT DETAIL CALCULATIONS

AVAILABLE SET MR INDICATOR TO SHOW MATCHING FIELD STATUS

READ AND IDENTIFY A RECORD

SELECT RECORD TO BE PROCESSED BY MATCHING FIELD SEQUENCE

IDENTIFY CONTROL FIELDS AND SET CONTROL LEVEL INDICATORS

TOTAL OUTPUT

DETAIL CALCULATIONS

MAKE DATA FROM SELECTED RECORD AVAILABLE

INITIALIZE

HEAD IN DETAIL OUTPUT

DETAIL TIME

TOTAL TIME

OUTPUT DETAIL CALCULATIONS

AVAILABLE SET MR INDICATOR TO SHOW MATCHING FIELD STATUS

READ AND IDENTIFY A RECORD

SELECT RECORD TO BE PROCESSED BY MATCHING FIELD SEQUENCE

IDENTIFY CONTROL FIELDS AND SET CONTROL LEVEL INDICATORS

TOTAL OUTPUT

DETAIL CALCULATIONS

MAKE DATA FROM SELECTED RECORD AVAILABLE

INITIALIZE

HEAD IN DETAIL OUTPUT

DETAIL TIME

TOTAL TIME

OUTPUT DETAIL CALCULATIONS

AVAILABLE SET MR INDICATOR TO SHOW MATCHING FIELD STATUS

READ AND IDENTIFY A RECORD

SELECT RECORD TO BE PROCESSED BY MATCHING FIELD SEQUENCE

IDENTIFY CONTROL FIELDS AND SET CONTROL LEVEL INDICATORS

TOTAL OUTPUT

DETAIL CALCULATIONS

MAKE DATA FROM SELECTED RECORD AVAILABLE

INITIALIZE

HEAD IN DETAIL OUTPUT

DETAIL TIME

TOTAL TIME

OUTPUT DETAIL CALCULATIONS

AVAILABLE SET MR INDICATOR TO SHOW MATCHING FIELD STATUS

READ AND IDENTIFY A RECORD

SELECT RECORD TO BE PROCESSED BY MATCHING FIELD SEQUENCE

IDENTIFY CONTROL FIELDS AND SET CONTROL LEVEL INDICATORS

TOTAL OUTPUT

DETAIL CALCULATIONS

MAKE DATA FROM SELECTED RECORD AVAILABLE

INITIALIZE

HEAD IN DETAIL OUTPUT

DETAIL TIME

TOTAL TIME

OUTPUT DETAIL CALCULATIONS

AVAILABLE SET MR INDICATOR TO SHOW MATCHING FIELD STATUS

READ AND IDENTIFY A RECORD

SELECT RECORD TO BE PROCESSED BY MATCHING FIELD SEQUENCE

IDENTIFY CONTROL FIELDS AND SET CONTROL LEVEL INDICATORS

TOTAL OUTPUT

DETAIL CALCULATIONS

MAKE DATA FROM SELECTED RECORD AVAILABLE

INITIALIZE
CP-V RPG VS. CP-6 RPGII

- COMPLETE SUBSET
- RELATIVE I/O
- LOOK AHEAD
- SPREAD CARD/PLACE
- ARRAYS/TABLES
- OPERATION CODE EXTENSIONS
OVERALL STRUCTURE

• 530 DESIGN
  MOM (MACRO OPERATION MACHINE)
  INTERPRET
  RUN-TIME

• COMPIILATION PHASES
  FILE AND EXTENSION PROCESSOR
  INPUT PROCESSOR
  CALCULATIONS PROCESSOR
  OUTPUT PROCESSOR
  FINALIZE PROCESSOR

• PERFORMANCE
  SIZE
  SPEED
TESTING AND VERIFICATION

- EXISTING TESTS
- OUTSIDE VENDOR

DEPENDENCIES AND SCHEDULE

- CP-6
- PL-6 STRING FUNCTIONS
- ASSEMBLER
- RELEASE
- RECRUIT
SOFTWARE TEST AND RELEASE CONTROL

- TESTING, ANALYSIS AND RELEASE DECISION BY ENGINEERING MANAGEMENT

- COMPREHENSIVE TESTING OF FUNCTION, PERFORMANCE, IMPROPER USE

- STABILITY DETERMINED BY EXPOSURE TO LADC USERS

- SPECIFIC INDIVIDUAL ASSIGNED AS RELEASE MANAGER

- RELEASE DECISION BY MANAGER, LADC

9/19/77 IPR/SK
- **EXTERNAL LOAD GENERATOR** (QUESTA SYSTEM)
- **HARDWARE MONITOR**
- **TIME-SHARING USER SIMULATOR**
- PROVIDES SUPPORT TO THE DEVELOPMENT STAFF AND RELEASE MANAGER
  - DEVELOPS TEST TOOLS AND TECHNIQUES
  - PROVIDES MEASUREMENT AND ANALYSIS SUPPORT
  - LIBRARIAN TO PROVIDE ORGANIZED, DOCUMENTED TEST DATA AND HISTORY

- TEST GROUP ESTABLISHED
  - STAFFING IN PROCESS NOW
  - TOOL BUILDING IN PROCESS NOW
  - ACTIVE MEASUREMENT BEGINS IN EARLY 1978, 1 YEAR BEFORE RELEASE
CP-V TO CP-6 CONVERSION

• PRINCIPAL ORIENTATION TO CONVERSION WHILE CP-V SYSTEM IN PLACE, BUT CAN BE DONE ON CP-6

• WILL NOT CONVERT MACHINE LANGUAGE PROGRAMS

• WILL CONVERT BULK OF:
  -- HIGH LEVEL SOURCE PROGRAMS
  -- DATA FILES WITH KNOWN DATA TYPES

• TAPE IS PRIMARY MEDIUM OF CONVERSION

• SPECIFIC CONVERTERS DEVELOPED BY EACH DEVELOPMENT GROUP
HIGH LEVEL SOURCE PROGRAMS

- CP-6 ANS FORTRAN COMPATIBLE WITH CP-V ANS FORTRAN
- COBOL USES CAPS AND PROGRAMS REQUIRE SOME MANUAL ASSISTANCE
- APL CONVERTER GENERATES APL SOURCE FROM WORKSPACES
- BASIC CONVERTER CONVERTS ALL STATEMENTS WHICH HAVE SAME SEMANTICS IN ANS BASIC
- EDMS SCHEMA CONVERSION TO IDS DATA DEFINITION LANGUAGE
FILE CONVERSION METHODS:

- ALL CHARACTER FILES TRANSPORTABLE VIA ANSI TAPE (UNLESS KEYED FILES)
  -- CP-V CAN WRITE ASCII, CP-6 CAN READ EBCDIC

- OTHER FILES USE ONE OF TWO METHODS:
  -- DATA CONVERSION PACKAGE, CP-6 TAPE WRITE PACKAGE ON CP-V
  -- CP-V TAPE READ PACKAGE, DATA CONVERSION PACKAGE ON CP-6
CP-6 MAJOR MILEPOSTS

• Transition to Honeywell
  July 1976

• Begin CP-6 Architecture Phase
  July 1976

• Level 66A Installation
  November 1976

• Begin CP-6 Detail Design and Implementation Phase
  January 1977

• A-System First L66 Code
  February 1977

• B-System Software Factory
  May 1977

• C-System PL-6
  July 1977

• Level 66B Installation
  September 1977

• Complete CP-6 Architecture Phase
  September 1977

• F-System - File Management
  November 1977

• H-System - Time Sharing
  February 1978

8/31/77  R.J.L.
MAJOR MILEPOSTS (continued)

- Four Million Words on L66B  
  February 1978
- Second CP L66B  
  March 1978
- K-System - Batch  
  June 1978
- Language Processors  
  July 1978
- Demonstration (K-System)  
  August 1978
- First Release Implementation Complete  
  October 1978
- In-House ALPHA Test  
  November 1978
- In-House Beta Test  
  January 1979
- Customer Benchmark  
  March 1979
- Final System Build (S-System)  
  May 1979
- Controlled Field Release  
  June 1979
CP-6 PROGRESS TO DATE

DESIGN REVIEW -

- 40 Documents
- Over 2100 pages

SOURCE

- In Checkout
  16,000 (PL-6) 6,000 (Assembler) D-System

SOURCE

- Awaiting Checkout (Estimate)
  26,500 (PL-6) 12,000 (Assembler)

8/31/77  RJL
A-SYSTEM

Checkout Bootstrap Environment
  Mini I/O
  Mini DELTA
  Mini Boot

Used GCOS GMAP - ABS LOAD
Used NSA (WSQO)

B-SYSTEM

Software Factory Test
  BMAP
  LINK
  DEF

Established Monitor Environment (WSQ1)

C-SYSTEM

PL-6 Test
Unit Test
  Memory Management
  CPU Scheduler
  Service Decoder
  Fault Handler

Stabilize C-System
Basis for Parallel Checkout and Integration

D-SYSTEM

8/31/77 RJL
<table>
<thead>
<tr>
<th>AAARDVARK</th>
<th>A BISON</th>
<th>A COYOTE</th>
<th>A DINGO</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINI-DELTA</td>
<td>FACTORY LINK</td>
<td>MM SCHILD</td>
<td>TEST DRIVERS</td>
</tr>
<tr>
<td>MINI-110</td>
<td>FACTORY DEF</td>
<td>SERVICE DESIGNER</td>
<td>STABLE SYSTEM</td>
</tr>
<tr>
<td></td>
<td>INITIAL</td>
<td>FAULT HANDLER</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLC MODULES</td>
<td></td>
</tr>
</tbody>
</table>

8/31/77    R.J.L.
COMMENT ON PL-6

- PL-6 to Date
  - No (serious) Code Generator Bugs
- CP-6 Code in PL-6
  - Straight Forward Checkout
    - L66 Hands-on
    - L66 Simulator
- Good Trend
- Too Soon To Tell
NSA PROBLEM RESOLUTION

- CP-6 - Dependent upon NSA
- CP-6 / NSA Problem Definition
  - L. Krasny (LADC)/R. Mynatt (Phx) - Change Request
- Two Sets of NSA Boards
  - In Hardware Engineering for LADC
- Approved Change Request (C. O.)
  - Install on System P1 (redwire).
  - Install on LADC Boards (Betty McCulley-Phx).
- FED (R. Fawcett) Tests and Sends Boards to LADC (P. Germain).
- NSA T&D (P. Drown)/Phx Sends Tape to LADC (H. Geshwind).
- LADC FED (E. Steinhauer) Coordinates with PHX FED (R. Fawcett).

8/31/77 R.J.L
What's Missing?

- Single Control Point
  - Hardware Changes
  - T & D

- Installation and Checkout Expertise For New Hardware at LADC

SOLUTION

- New Product Test Engineering
  - Coordination
  - Installation

8/31/77  R JL
CP-6 PROBLEMS AND ISSUES

- Test and Diagnostics
  - TOLTS Interface to CP-6
  - HEALS FED Errorlog Spec
- CP-V Drain on Management and Programmer Resources
- Software Factory - Dual 560
  - Production for CP-6
  - Test for CP-V

8/31/77    RJL
CP-V   F01      October 1977

- MPC Disks
- Sigma 5 MAP Support (F00)
- Multi-Account Private Packs (F00)
- Multi-Processing Performance Improvement
- 300 SIDRs Closed (CK-97s)

CP-V   December 1977

- Dual Sigma 6/7
- Sigma 6/7 to 256K (with MOS)

CP-V   F00      March 1978

- MPC Tape
- MOS Memory (Single bit error logging)
- 300 SIDRs