Current PRISM OS Strategy

- Two system strategy:
  - PRISM ULTRIX. "World-class" implementation of industry-standard system targeted at workstation and technical marketplace
  - Mica. Proprietary operating system that addresses commercial system requirements, VMS compatibility, contemporary computing concepts, and extensibility.

- PRISM operating systems share in the short-term:
  - Languages and layered products
  - AIA and DECwindows

- PRISM operating systems share in the long-term:
  - Service strategy and on-line diagnostics
  - Operating system components
Key Concepts of New Proposal

- Implement OSF and Mica/VMS capabilities with a single set of operating system software.

- Integrate at the program execution level rather than the session level - you run programs in an environment.

- Separate UN*X execution environment, objects, and abstractions from Mica counterparts.

- Structure so that compatibility tradeoffs between UN*X and VMS are not necessary.

- Do not attempt to make UN*X concepts available to Mica programs and vice versa.

- Do not attempt to share everything simultaneously.
Problems with Previous Effort

- Attempted to integrate at the session level - you logged into an environment.

- Attempted to fully integrate UN*X concepts with Mica concepts in a unified way.

- Attempted to totally share all objects and abstractions of the two systems.

- Attempted to make all UN*X concepts available to Mica programs and vice versa.

- Were forced into making tradeoffs of VMS compatibility versus UN*X compatibility.
New System Structure

OSF Environment
- RMS
- ARUS
- Math
- Language Runtime
- DECwindows

Mica Environment
- RMS
- ARUS
- Math
- CMA
- Language Runtime
- DECwindows

Non-Privileged

OSF System Services
- Fork
- TTY
- Streams
- Filesys

Mica System Services

Privileged

Common Executive
- Drivers
- Memory Management
- Object Architecture
- Network

Common Kernel
- SMP
- Scheduling
- Synchronization
Common Executive Components

- System management, system bootstrap, and security.

- Memory management, object architecture, and process structure.

- Device drivers, on line diagnostics, and error logging.

- DECnet and DECwindows.

- High reliability file system that is compatible with both UN*X and VMS.

- SMP and scheduling.
Features

- Absolute OSF compliance without impacting the design of Mica.

- No need to make tradeoffs in favor of either OSF or VMS - both can be accommodated equally.

- Database, enhanced security, availability, and other industrial grade features available to both environments.

- Maximum leverage of software development resources - Digital's products all run in Mica environment, but are freely usable from OSF environment.

- Common DECwindows and AIA support.

- Common language runtime environment (e.g. math library, RMS, ARUS, etc.)

- Mica protected subsystem functionality available to OSF programs as well as Mica programs.
Benefits

- One system with two execution environments versus two systems with shared components.

- OSF not second class citizen - compute performance equal to Mica.

- Layered product development, testing, and certification on a single system.

- Field support personnel only need to be trained on a single system.

- SMP and other Mica capabilities available to OSF environment day one.

- Quality system management and network support.
Open Questions

- What is the OSF operating system standard - XOPEN and POSIX?

- What about VAX ULTRIX compatibility, Berkeley UN*X compatibility, SVID compatibility?

- What should be done about UN*X concepts that compromise security (i.e. set UID and set GID)?

- What about Sun tools - Yellow Pages, RPC, etc.?

- What should we do about multithreading? UN*X doesn’t have it - we could provide it - but why?

- Will an OSF system that is not pure UN*X be saleable in the UN*X market?