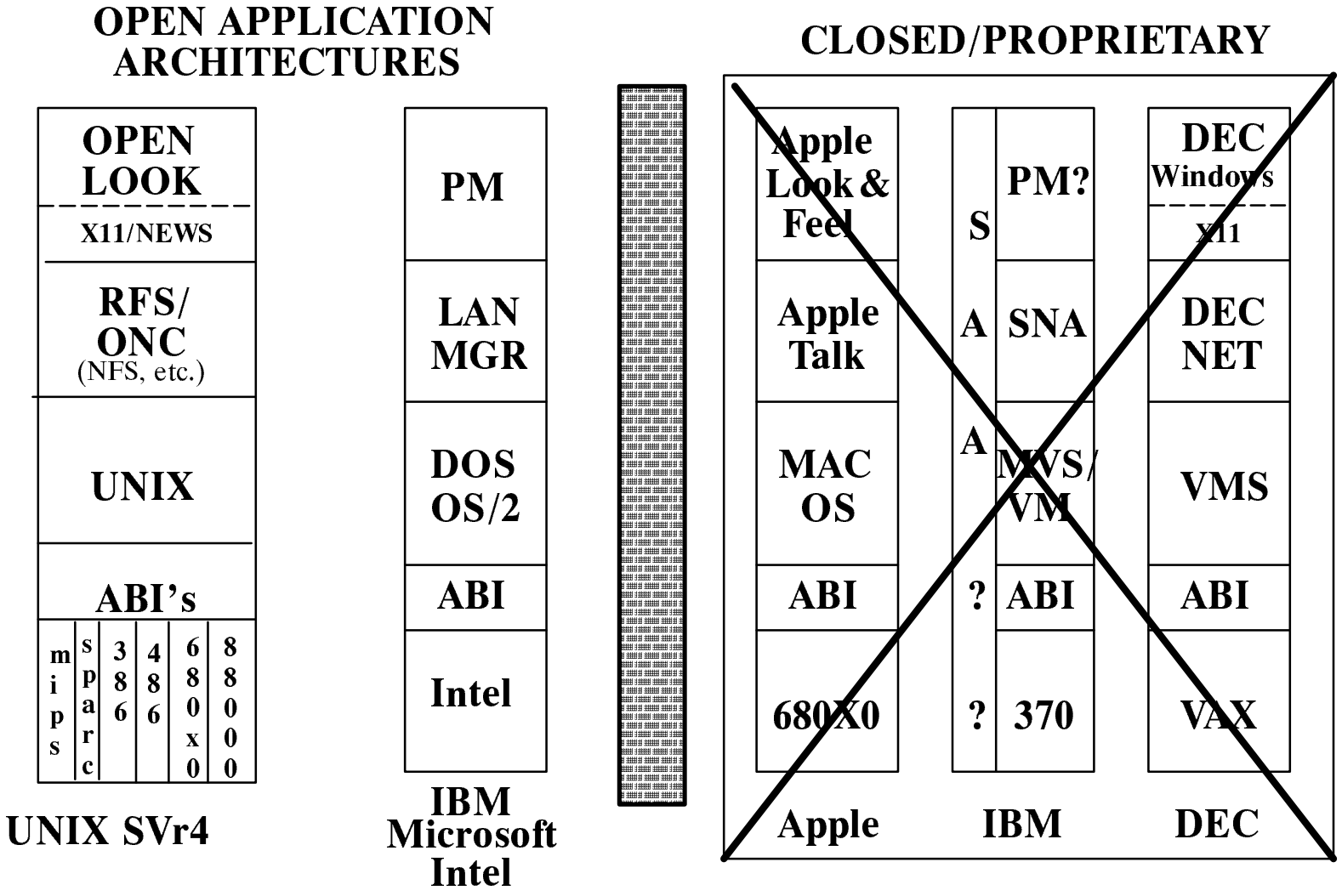
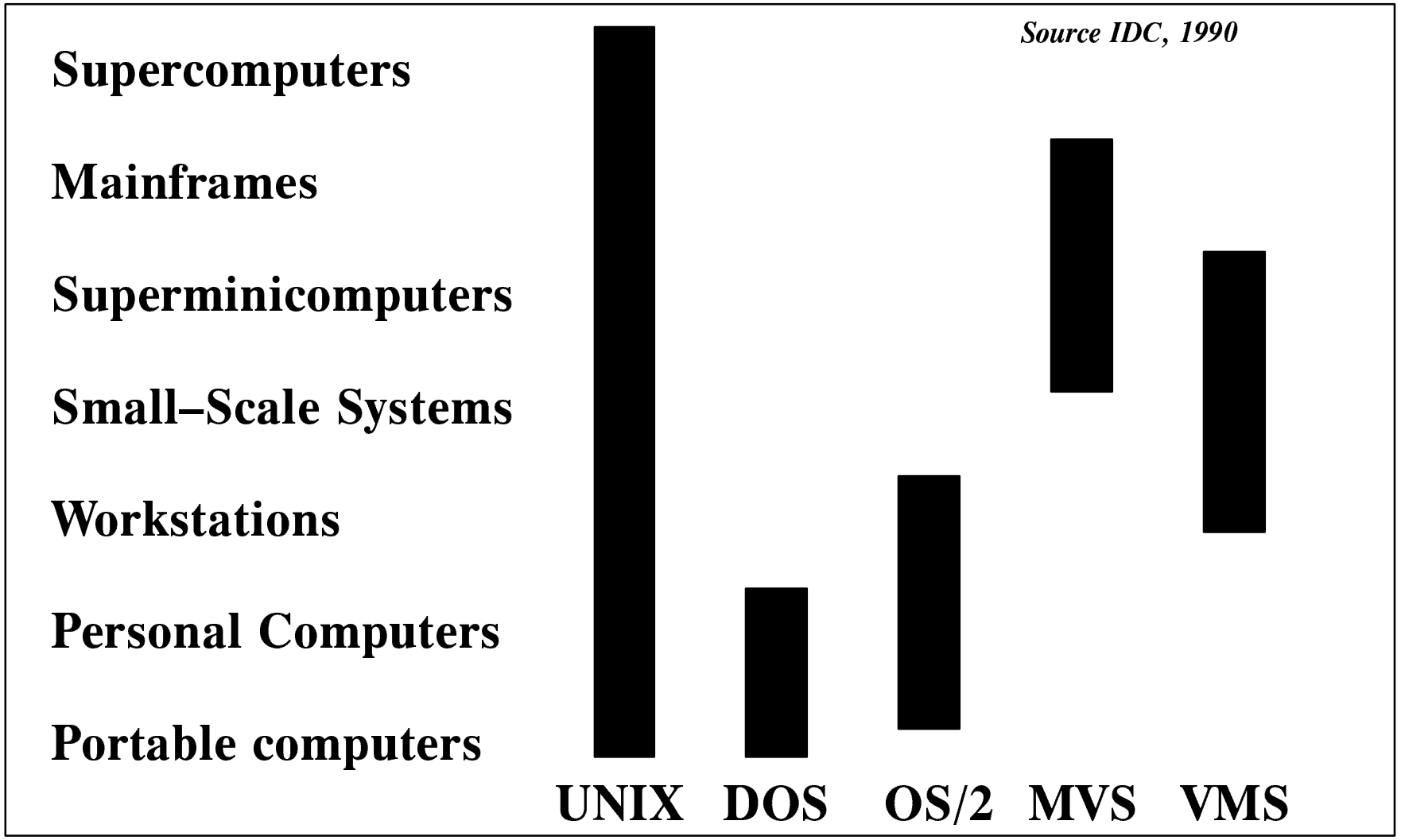
# COMPUTING IN THE 90's

standards
group productivity
distributed networking



# SVr4–OS/2 Scalability

*Source IDC, 1990*



# Windows 3, OS/2 and Unix

**Three Choices**

- **Windows 3**
- **OS/2**
- **Unix**

**Windows 3 gives the DOS user time to make the OS/2 UNIX transition decision.**

**Unix provides much more power, robustness and openness.**

**Many DOS environments will transition DOS, Windows 3, Unix SVr4**

# Windows 3, OS/2 and Unix

**Windows 3** is a powerful concept, introduced by Microsoft, to bring "multi Tasking" capabilities to the DOS desktop. Windows 3 runs on a 386 or 486 and provides a look and feel identical to that of Presentation Manager. Windows 3 provides the first step in the direction of the end user moving from the single user, single tasking DOS environment into the single user multi tasking environment. Windows 3 applications will run under OS/2 in the compatibility mode. Porting Windows 3 applications to OS/2, PM is a major task. In a bid to keep the desktop Microsoft is planning to enhance OS/2 to directly run Windows 3 applications. This is probably some way down the line. The race for the desktop is basically between Unix and OS/2. Perhaps the outcome may be based on this type of functionality. Will Unix vendors provide this functionality and compatibility. It seems like there could be a large market here and the stakes are high. Windows 3 for now gives the end user breathing time to think before making the decision to move to OS/2 or Unix. A picture that seems to occur at this point is "Why go to OS/2" at all. As an end user it seems logical to adopt Windows 3 for now and when the time is right move to Unix where you have the more robust SW environment running across multiple platforms, based on industry standards, more applications, and a much better networking, distributed computing environment built in.This coupled with the fact that SVr4 has the capability of supporting OS/2, DOS and any other file system makes it a safe bet as the OS of the 90's. OS/2 would not have the capability of running SVr4 as an IFS.

# System & Protocol Dependencies

| | SVr4 | OS/2 |
|---|---|---|
| Operating Systems | Multiple Platforms | Intel Only |
| Graphics Interfaces | X11 Protocol | OS/2 Kernel Based |
| | Arch. Independent | Arch. Dependent |
| | Distributed | Not Distributed |
| Networking | Open Standards | IBM LU 6.2 |
| | TLI | |

# System & Protocol Dependencies

**OS**: Unix SVr4 is written in a high level language with data structures developed in such a way that porting this OS to new architectures is straight forward. One example of this is the hardware address translation (HAT) data structures that porting of the VM system straight forward. OS/2 on the other hand is written in assembler specifically for the Intel line of computers. There is talk of OS/2 becoming portable. The first question is when. The second consideration is the robustness of the system when it is done. This will be NEW OS with many new problems to iron out. OS/2 is also planned to be ported to Intel RISC architecture. Yet another upheaval for the customer base to go through. Unix has 20 years of development behind it as opposed to 2.5. Most of these issues have been ironed out. Unix is more robust with more functionality, more applications and the greatest potential to be the OS of the 90,s.
**GUI**: X11 is a communications protocol. It is a protocol that is independent of the underlaying architecture. This is what has made X11 so popular and powerful. In addition it is a distributed graphics interface. Presentation Manager is a kernel based windowing environment and so closely tied to OS/2 that you cannot really consider the one without the other. PM is totally tied to an dependent on OS/2 (on an OS/2 system).
**Networking**: SVr4 supports all the standards necessary for networking in an interconnect or distributed environment. OS/2 is very closely tied to IBM LU6.2 environment.
NOTE: LAN Manager functionality has been adopted by X/OPEN and is known as LM/X. LM/X is gaining market share with large OEM's and vendors such as HP and DEC. DEC is using LM/X to connect it's personal computers to the VAX mainframes.

# Distributed Computing

| Distribution OF: | SVr4 | OS/2 |
|---|---|---|
| CPU | ONC/REX | NO |
| DATA | ONC/NFS | LAN Mgr |
| Graphics Interface | X11/NEWS | NO |
| Applications | ONC/RPC/XDR | LAN Mgr |

# Distributed Computing

When looking at a truly distributed computing environment there are a number of areas of distribution to consider. 1) Distribute your CPU usage. 2) Distribute your data. Distribute your graphics interface. 4) Distribute your application. SunOS/SVr4 provides all of this functionality via the Sun's Open Network Connect (ONC). ONC/REX provides for CPU distribution with the "on" command. ONC/NFS, a widely accepted industry standard, provides for data distribution in a heterogeneous environment. X11/NEWS provides for the distribution of the
graphics interface and ONC/RPC/XDR provides for distribution of your application. Further X11/NEWS and ONC/RPC/XDR are a standard part of the base level release of SVr4 from AT&T.
OS/2 provides LAN Manager which supports distributed computing It appears that LAN Manager is used in an environment such that client A talks to a server and the server talks to client B in order to get communications from client A to Client B. If this server were to be an IBM mainframe the security issue would be solved at that point but LAN Manager does not appear to provide a true distributed processing environment. as per distribution of applications (Eg. ONC/RPC/XDR) Further points of consideration: OS/2 is a single user system so distribution of CPU can not be supported as with ONC/REX. LAN Manager supports distribution of data. Presentation Manager is a kernel based window system so distribution of the graphics interface is not supported as with X11/NEWS.
In general networking seems to be a secondary thought in OS/2. Much of the networking interconnect is left up to third party SW vendors like Banyon etc. SVr4 has much of its networking to support distributed computing in. SVr4, now, like OS2 or DOS conforms to an ABI which will allow for the ISV to develop shrink wrap soft ware. Most of these factors are likely to make SVr4 the OS of the 90's.

# Internationalization

- Commands – 8th bit cleanup and international code set
- Multi–byte character widths (libraries)
- National conventions (libraries)
- Message handling facility (libraries)

# Internationalization

Both SVr4 and OS/2 provide Internationalization features. SVr4 provides 8bit clean ASCII code set providing support for European countries. The libraries support multi-byte characters for full international support. National conventions provide for the support of different local's, date & time, number formats etc. SVr4 message handling facility also provides a means of translating system messages to a users native language. SunOS/SVr4 (FCS) version will support full internationalization capabilities for Eastern block, Pacific rim, Arabic, etc. OS/2 provides much of the same internationalization as that in SVr4. It would appear that OS/2 is a little ahead of SVr4 with it's internationalization with respect to localization etc.

# Security

**Major flaw in OS/2 for networked environment**

**User and group ID issues**

**Security on SunOS/SVr4**

- **Split Password File**
- **Account Expiration**
- **Password Aging**
- **Disabling Logins**
- **Audit trail**

**Security Ratings**

- **C2, B2 for Unix**

# Security

The fact that OS/2 was built as a personal computer OS no concern was given to the security of the system files based on the premiss that the files on the system were owned by the user. There are no user or group access permissions etc. This situation seems less than optimal in a networked environment. SVr4 has matched C2 security requirements and work is under way at Unix International and AT&T Unix SW Laboratories to have the Unix kernel comply with B2 requirements. SunOS/SVr4 will support such security issues as Split Password File, Account expiration, Password aging, Disabling logins and audit trail features required in C2 specifications.

# Character & Graphics I/O Subsystem

| Functionality | SVr4 | OS/2 |
|---|---|---|
| Character Interface | FMLI<br>ETI | VIO |
| Desktop | FMLI | |
| Graphics Interface | OPEN LOOK<br>Distributed | PM<br>Kernel Based |

# Character & Graphics I/O Subsystem

**Character User Interface**: SVr4 provides the Form & Menu Language Interface (FMLI) and the Extended Terminal Interface (ETI). FMLI is a toolkit that allows developers to write user friendly interfaces with a common look and feel. ETI is a toolkit that allows the developer to write customized character based interfaces. SVr4 also provides the Framed Access Command Environment (FACE) that that provides unsophisticated users on the desktop the ability to easily open files, create directories, without needing to know the Unix commands. FACE is developed with FMLI. **OS/2** too provides a character based interface known as VIO. VIO consists of three subsystems: 1) Keyboard (KBD), Video I/O (VIO where the name comes from) and mouse (MOU). VIO is much like the MS-ROM-BIOS. Like ETI, VIO provides extended functionality in that it allows one to bypass the character mode enabling the developer to talk directly to the frame buffer hard ware for graphics mode applications.

**Graphics User Interfaces (GUI)**: SVr4 supports the MIT X11 based OPEN LOOK GUI. The SVr4 GUI incorporates a merge of X11 and Post Script based NEWS from Sun Microsystems. X11/NEWS is a distributed graphics interface, providing a client server model for distributed applications. X11 and NEWS have been developed using a common window manager. NEWS has the big advantage over X11 in using less network traffic to communicate (Thin Wire0. As applications become more sophisticated and the industry moves to ISDN, multi-media etc, SVr4 is well positioned to support these applications with the user being unaware of the difference in the look and feel between X11 or NEWS. **OS/2** supports Presentation Manager. PM is kernel based in OS/2. It is so tightly coupled that it is difficult to think of OS/2 and PM as different entities. This tight coupling does not make for good distributed applications. Rather OS/2 used LAN Manager to provide multi user capabilities, client server relationships, distributed processing etc. This means extra cost to the user in having to purchase this additional functionality. SVr4 has it all built in.

# Systems & Network Management

| Functionality | SVr4 | OS/2 |
|---|---|---|
| Systems Administration | Backup/Restore<br>Device Management Tools<br>SW Installation | |
| Network Administration | Sun Net Manager | LAN Manager |

# System & Network Management

**SVr4** provides system administration menu's enabling the user to undertake system administration commands without the need to know Unix commands. These commands are based on FMLI discussed earlier. The menu's enable backup & restore, SW installation, creation and deletion of user accounts and configuration if peripherals etc. In addition to this interface Sun specifically is adding much value to SVr4 in the area of ease of use, systems administration etc. All of this added value will be done under the OPEN LOOK interface making for extremely simple administration. SVr4 also provides device management tools that provides information on the states of devices (open, busy etc.) and simple device management commands to add, remove, free, list devices etc. SVr4 also provides SW installation Tools. These tools assist in the creation, registration and removing of SW packages loaded on the system. Sun provides SunNet Manager for network administration. This product with it's open API's will eventually enable systems administration to be undertaken from an administrators seat on any machine on the network. SunNet Manager also provides open interfaces enabling the system to easily report up to enterprise management systems.

**OS/2** provides LAN Manager which is used to provide multi user capabilities via the network. LAN Manager is a distributed architecture providing such services as network management, file and print serving etc. The package consists of a server component and a requestor that enables DOS, OS/2 and Microsoft Windows based applications to get services from the server. LAN Manager does not appear to be tightly integrated into the kernel. It has functionality built in to support network aware applications such as which printer has the shortest queue of jobs etc.

# IPC

| Function | SVr4 | OS/2 |
|---|---|---|
| Signals | Yes | Yes |
| Semaphores | Yes | Yes |
| Anonymous Pipes | Yes | Yes |
| Named Pipes/FIFO's | Yes | Yes |
| Message Queues | Yes | Yes |
| Shared Memory | Yes | Yes |
| Dynamic Data Exchange | Sun active in this area | Yes |

# IPC

In traditional programming modules within a single process can share data by using global variables. In systems running multiple processes the OS must support facilities for these processes to communicate with each other. Both OS/2 and Unix support multiple forms of Inter Process Communication (IPC) to fulfill a variety of needs. Some of the common forms of IPC in both Unix & OS/2 is as follows:

**Signals**: Analogous to a HW interrupt, causing execution to begin at a handler address, usually occurring asynchronously. OS/2 mechanism is very similar to Unix. OS/2 supports 6 signals (^c, ^break, pgm termination and three general purpose signals) Unix supports 32 signals

**Semaphores**: A simple flag or a signal that works in a simple on or off fashion. Generally used for mutual exclusion, locking out a process until it is safe for it to get access to some data point.

**Pipes**: A pipe provides a one way flow of traffic between two processes. pipes or anonymous pipes can only act as an IPC between processes that are spawned from the same parent process. Both Unix and OS/2 provide support for anonymous pipes. Named Pipes or FIFO's are similar to pipes but do not need to be related to the same parent. Named pipes therefore work both locally, within a machine, or across a network. Named pipes have been supported in Unix for many years. They were not available in OS/2 Ver.1 but will be available.

**Message Queues:** Allow multiple writes to a single queue, bringing messages to a process from different senders. They resemble named pipes in that they are strictly FIFO. Each queue has a single owner but can have multiple clients.

**Shared Memory**: The most efficient form of IPC as once shared memory is set up no OS calls or copying is done. The processes share the shared memory segment. Semaphores are normally used in conjunction with shared memory in order to provide mutual exclusion as discussed above. Shared memory is supported in both OS/2 and SVr4.

**Dynamic Data Exchange**: Specific to OS/2, DDE is a form of IPC available to applications that use Presentation Manager.PM,s interface is message oriented, enabling applications to define private messages. The power behind this feature is the ability for one process to pass data dynamically to another process. IE: A CAD package may be dynamically linked to a Lotus spread sheet. As the drawing is updated, the spreadsheet could immediately recalculate the additional parts required and the new cost of the project. This is a very powerful feature of OS/2.. Sun is active on products that will provide similar but more powerful functionality, in this area, but the author is not at liberty to reveal further details at this time.

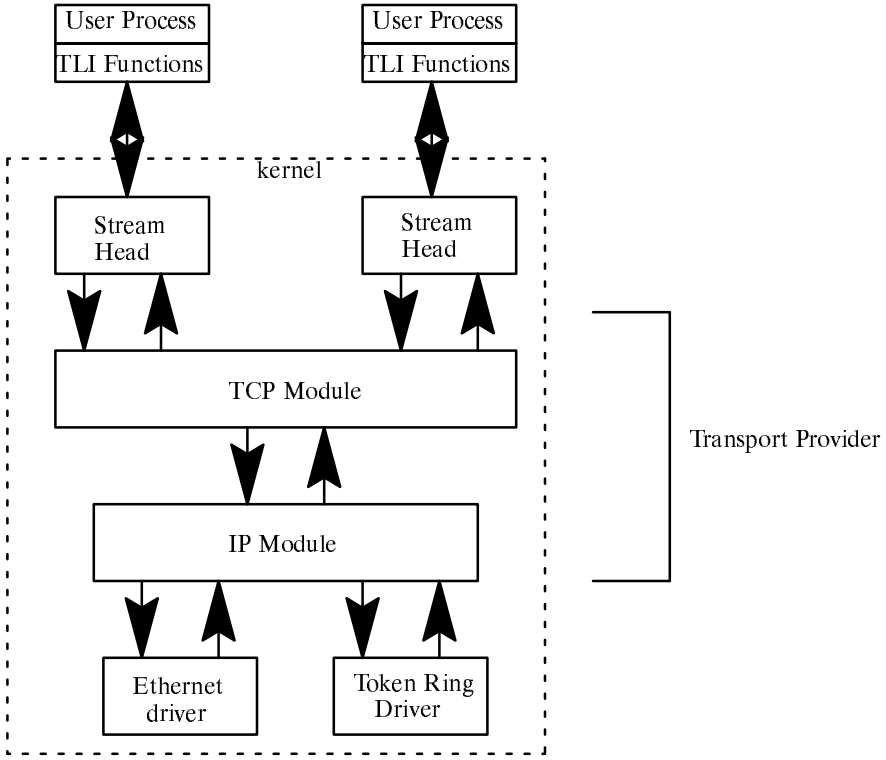
# Networking Subsystem

| Functionality | SVr4 | OS/2 |
|---|---|---|
| | TCP/IP | * APPC (Extended Edition) |
| | RPC/XDR | * IBM Connectivity |
| | Sockets | * SNA |
| | Vnodes | * LAN's |
| | NFS | * X.25 |
| | RFS | |
| | Network Selection | |
| | Streams | |
| | TLI | |
| | OSI | |

* OS/2 relies heavily on third party solutions for networking, as well as the LAN Manager to support multi user. Consider cost, support and future of networking with OS/2

# Network Subsystem

**Network Support**: SVr4 provides much networking built in functionality: 1) TCP/IP–DARPA protocol suite. An industry standard set of protocols that supports heterogeneous networking. 2) RPC/XDR– Sun Remote Procedure Calls & Data Representation provides functionality for a process to call a procedure on a remote machine, using the syntax of a local function call. RPC utilizes XDR to encode data passed between machines in a heterogeneous environment. 3) Sockets– STREAMS is pervasive in SVr4. The Berkley socket interface is implemented on top of STREAMS via libraries. 4) NFS enables file sharing in a heterogeneous environment. NFS is an industry standard developed by Sun Microsystems for file sharing in a heterogeneous environment. RFS is also supported in SVr4. Both NFS and RFS share the same administrative commands. 5) Network Selection–Allows network applications to choose the network over which they wish to run. This takes the burden off the developer and makes for easier systems administration. SVr4 also supports basic network utilities such as Honey Danber UUCP.
OS/2 provides APPC which provides a common interface to IBM machines to SNA, X.25 and LAN's. It is strongly tied to the IBM proprietary LU6.2 protocol
**Streams**: As mentioned earlier STREAMS I/O provides a powerful interface to character devices pseudo devices and networks. With respect to networking STREAMS provides powerful functionality. The device driver becomes the network interface(Ethernet, Token Ring etc.). Modules pushed onto the stream between the head and the device driver implement the network protocol(TCP/IP, XNS, OSI). The modules are termed the transport provider. STREAMS modules pushed onto the STREAM can also implement such functionality as multiplexing. All of this functionality makes for a powerful, flexible, modular interface making a good migration path to new protocols such as OSI, which are bound to become the standard for OPEN systems interconnects during the 90's. The TLI (Transport Library Interface) is a set of library routines that hides the streams interface from the user., making for simpler systems programming. Applications may access the stream I/O directly.
OS/2 provides no such functionality. Instead device drivers need to be written to take of each new functionality that may be required. To a large degree network implementation is an after thought in OS/2/ Much of this functionality is left up to third party vendors. One should keep in mind the affects of support, cost and integration costs with this approach.




STREAMS Network
Implementation in SVr4
OS/2 provides no such functionality

# File Subsystem

| Function | SVr4 | OS/2 |
|---|---|---|
| File System Name | VFS [Virtual File System] | MS-DOS [FAT] |
| Supported FS Types | s5 [1/2K, 1K, 2K] | |
| | UFS [Unix 8K FS] | |
| | RFS [Remote File System | |
| | NFS [ Network File System | |
| | /proc | |
| | fifofs | |
| | bfs [ Bootable File System | |
| Expendability & Flexibility | VFS [Virtual File System] | IFS [Installable FS] |
| Directory Caching | Yes | |
| Directory Tree | Hierarchical Unix SVr4 | Hierarchical OS/2 |
| Length of File Names | 256 Characters | |

# File Subsystem

**File System Specifics**:The OS/2 File System is identical to the MS–DOS FAT (File Allocation Table) file system. It is not flexible or expandable. It has been proved to be a very slow file system for concurrent processes accessing the same disk. This is a big disadvantage of this implementation in the multi tasking environment of OS/2.. OS/2 Ver. 1.2 implemented the High Performance File System , providing about a 25% improvement in performance. Unix today (May1, 1989) quoted that the Unix "s5" file system out performed OS/2.. SVr4 incorporates the "UFS" (Unix File System – 8K) which once again is much faster.

The SVr4 file system VFS (Virtual File System) is a merge of AT&T File System Switch and Sun Microsystems Vnodes. This architecture provides a modular, cleanly defined interface between the file system and the kernel, allowing several different file systems to exist simultaneously in the system. User programs manipulate the different file systems (local or remote) through a common interface. Different file systems supported: 1) s5: Original AT&T Unix (1/2, 1, 2)K FS. 2)UFS (Unix File System) or Berkely 8K FS. 3)RFS (Remote File System) Remote sharing of files in a System V environment. 4) NFS (Networked File System) – Remote file sharing in a heterogeneous environment. An industry standard. 5) Proc File System: Provides a mechanism for accessing the address space of running processes. Powerful for debugging. 6) FIFO FS: Provides common access to pipe files. 7) specfs: Common–code interface to device or special files. 7) bfs: Supports file system independent booting..

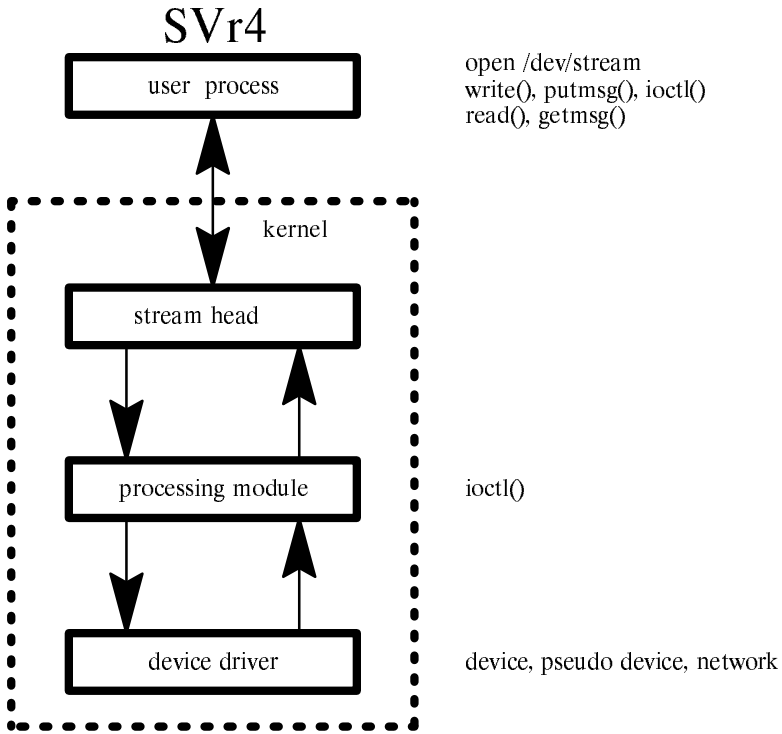
**Expandability**:SVr4 provides the VFS interface to implement additional file systems if so desired.. MS–DOS and OS/2 could be easily implemented in this architecture if deemed necessary. More importantly this architecture looks to the 90's to implement any new file system that may be required to meet new challenges for file systems. The OS/2 architecture was developed with IFS (Installable File System) in mind. An IFS is similar to a device driver, being installed at boot time. With this architecture multiple file systems may be installed. The main difference here between the two systems would appear to be the tight coupling that SVr4 provides for these multiple file systems within the kernel. Time will tell what the difference in performance may be between the two implementations.

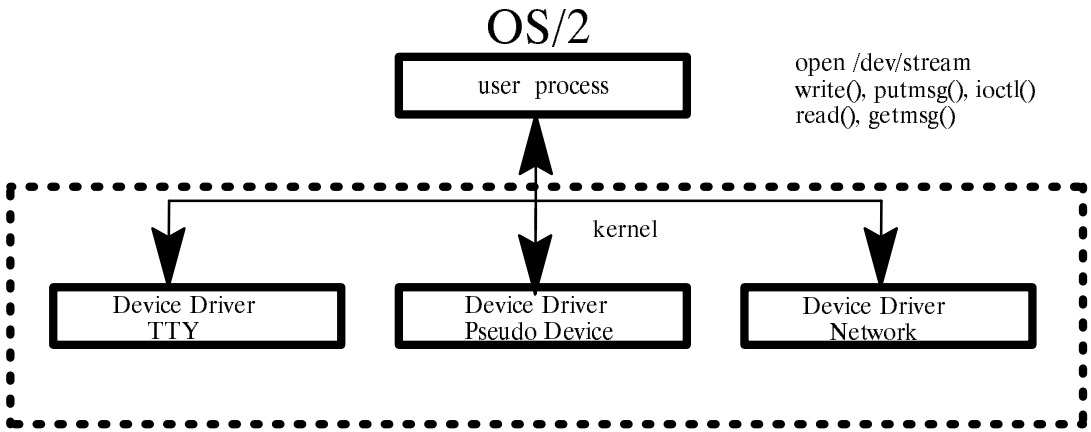
# I/O Subsystem

| Function | SVr4 | OS/2 |
|---|---|---|
| Character I/O | Streams I/O<br>tty<br>pty<br>pipes (named, anonymous)<br>Non-Blocking I/O - BSD<br>Dynamic Allocation - KMA | Monolithic |
| Synchronous I/O | Yes | Yes |
| Asynchronous I/O | IEEE1003.4 | Yes |

# I/O Subsystem

**SVr4 Character I/O interface Streams:** A stream is a full duplex connection between a user–process and a open or pseudo device. STREAMS defines standard modular interfaces for character I/O processing modules that implement Device Drivers, protocol handlers(TCP/IP, XNS, OSI) and line disciplines(tty, pty) etc. User processes can insert modules onto the stream between the "upstream" end( user process) and the "downstream"end (device driver end). These modules are self contained and functionally isolated, communicating through a well defined message passing interface. Opening a STREAMS I/O ( open /dev/stream) creates a stream between the user process and the device driver. Intermediate modules are attached using the ioctl() system call to push new modules onto the stream. STREAMS I/O works on bi–directional messages that flow through the stream. write(), putmsg(), ioctl() take messages from the user process and push them onto the stream. getmsg() and read() get messages off of the stream and return them to the user process. **The importance of STREAMS I/O is that it provides a standard, modularized interface for character I/O processing for TTY's, Pseudo Devices and Networking.**. This functionality becomes very important in networking, implementing new protocol stacks and migrating to OSI etc. More on this issue under Network Subsystem. In SVr4 the terminal subsystem, tty's, pty's are implemented using steams. Named and anonymous pipes have a common streams implementation so STREAMS modules can be mounted on pipes. SVr4 STREAMS provides BSD style non–blocking I/O. STREAMS data blocks and other data blocks are allocated dynamically via the kernel memory allocater(KMA)rather than static pre–allocation thus optimizing memory space.
**OS/2 Character I/O Interface**: OS/2 adopts a monolythic device driver structure. There is no support for layering of functions and modules that makes STREAMS I/O so powerful. IBM provides guidelines for writing device drivers.To add a new feature one needs to write a complete device driver to provide the required functionality using the appropriate system calls. Most communications packages will work this way.
**Synchronous I/O**: SVr4 implements synchronous I/O via the fsync() system call and O–SYNC option. OS/2 supports synchronous I/O with a "Write Through" option.
**Asynchronous I/O**: SVr4 base line release from AT&T does not support Asynchronous I/O. This was left out pending the definition of the IEEE10003.4 specification. Asynchronous I/O will then be supported. Sun Microsystems currently supports Asynchronous I/O on SunOS 4.x. This functionality will be supported in the FCS version of SunOS/SVr4.



# Process Management Subsystem

| Function | SVr4 | OS/2 |
|---|---|---|
| Response Design Goals | Balanced System Throughput | Interactive |
| Multi User | Yes | No |
| Multi Tasking | Yes | Yes |
| Light Weight Processes | Yes | Yes |
| Pri. Based Preemptive Scheduler | Yes | Yes |
| Kernel Preemption | Yes | |
| Swapping | Yes | Yes |
| Paging | Yes | No |
| Memory Locking | Yes | No |
| Multi Scheduler Support | Yes | No |
| Standards [IEEE10003.4] | Yes | |

# Process Management Subsystem

**Response Design Goals**: The OS/2 scheduler optimizes for response rather than system throughput. This is in line with the design goal that OS/2 is designed as a Operating System for Personal Computers, hence prioritizing interactive response. SVr4 optimizes for system throughput and balance for a networked environment while providing high priority to interactive processes.

**Multi User**: OS/2 developed as a PC operating system is single user only. SVr4 is a multi user system. Whilst one may argue that most of the operations on a users system are single user, multi tasking there are many additional benefits that roll out from a multi user system. 1) Security: As a PC OS, OS/2 provides NO user and group access permissions.. This tends to indicate that OS/2 will be unsuitable for Corporate and government environments. One solution would be to use the system as a dataless client on an IBM server where data files may be much more secure, but this further tends towards IBM proprietary systems, away from standards and gives a clear indication that OS2 will not be useful on mainframe computers, especially in a secure data environment. Hence the government's move to Unix. 2) Networking. Multi user is very useful in a networking environment, from the simplest point of rlogin to debugging across the network. 3) Large demand for X Terminals will tend to make Unix multi user systems more useful than OS/2.. The OS/2 approach to multi user is to add Microsoft LAN Manager. This incurs additional overhead and added cost of a network OS as well as all the required HW to support this functionality. SVr4 is fully multi user. If it gets down to the most basic point that multi user is not important to a user there is still the fact that SVr4 provides it for free. It is a big plus and sure can't hurt to have it.

**Multi Tasking:** Both SVr4 and OS/2 are multi tasking Operating Systems

**LWP**: OS/2 makes heavy use of threads. Threads are somewhat like processes running within processes. They have the advantage of using less overhead to switch from one task to another as they all belong to the same process. This makes for faster interactive response for instance but takes away the safety net that multiple processes provide. If one thread crashes the process, all the threads in that process are liable to crash. SVr4 makes heavier use of multi processes and does not support LWP in the base line release. SunOS/SVr4 will provide support for light weight processes.

**Priority Based Preemptive Schedule**r: OS/2 assigns priority to each thread, priority being given to the interactive thread for faster user response. Provision is also made to assign a fixed priority to a time–critical thread, but does not imply that kernel preemption is supported. SVr4 supports multiple schedulers, specifically time sharing and real–time with provision being made for kernel preemption as well as the incorporation of additional schedulers if so required.

**Kernel Preemption**: To the authors knowledge OS/2 does not support kernel preemption. Kernel Preemption is supported in SVr4. Data structures have been clearly developed to easily comply with the IEEE10003.4 specification when that specification is defined.

**Swapping**: Both SVr4 and OS/2 support swapping

**Paging**: OS/2 does not support paging. Entire processes must be swapped out. This would tend to mean a heavier system overhead in swapping processes rather than just pages, rather than doing useful work.

**Memory Locking**: OS/2 provides no mechanism to lock processes into memory. The overhead of swapping on a small PC is expensive. This important functionality has been overlooked. in OS/2. SVr4 provides the ability to lock data and/or text segments into memory via the "plock()" system call. Portions of a process needing fast response can be locked in memory making them immune from swapping or paging..

**Multi Scheduler Support**: As indicated above OS/2 provides a single scheduler. SVr4 data structures are cleanly implemented in such a way that two different schedulers are supported now. More important, as we go into the 90's, we see in SVr4 the functionality to easily integrate additional schedulers as the need arises.

**Standards[IEEE10003.4]**: The author is unaware of any development goals of OS/2 to meet the real–time standard when that solidifies. SVr4 has the data structures cleanly implemented to incorporate that standard when it is finalized. SVr4 will be IEEE10003.4 compliant.

# SVr4 Scheduler

The SVr4 scheduler provides a modular interface to the kernel and allows several schedulers to reside on the system simultaneously. SVr4 provides flexible scheduling options by currently supporting two schedulers, namely a *standard time sharing* and "user controlled" *settable fixed priority preemptive* scheduler. System developers may easily develop and implement new schedulers for SVr4. A process can switch from its own scheduler to another whilst the system is running and several different schedulers can be running on the system simultaneously.. SVr4 time sharing mode is developed for maximum system throughput and balance, which is important in a networked environment, while favouring interactive processes. The data structures that support these classes are cleanly developed to easily conform with the IEEE10003.4 specification when finalized.

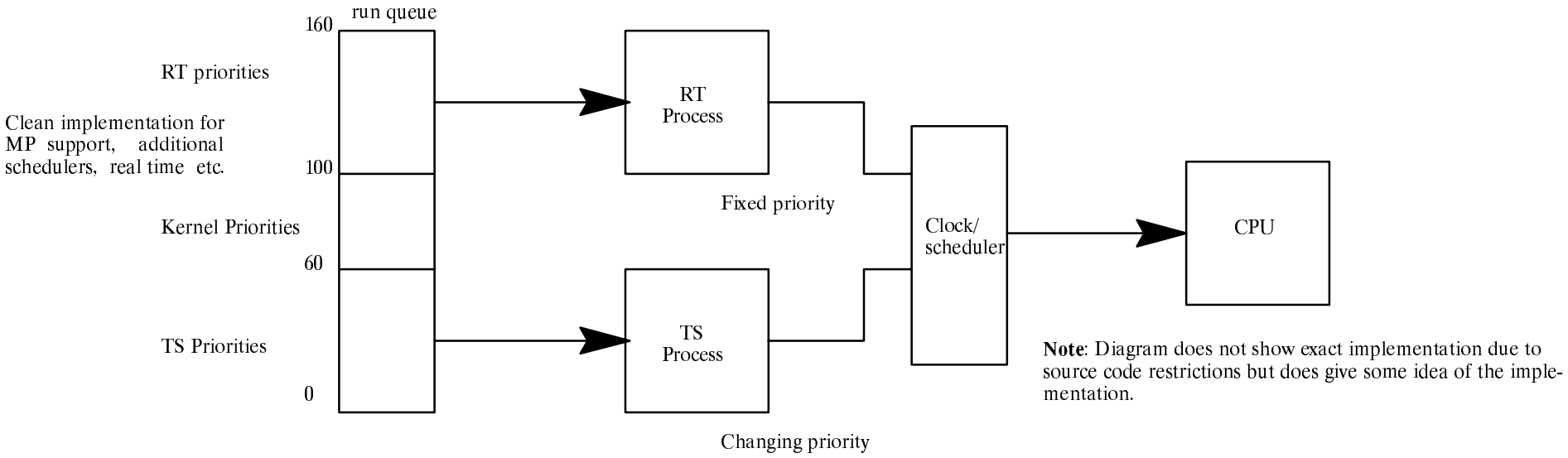
In addition SVr4 provides additional support for real-time processes:

**High resolution timing services**. Processes have access to timing services ranging from Sec. to nSec. Alarms, TOD, interval timers etc.

**Memory Locking**: Processes are able to lock their text and/or data segments into memory using the plock() system call. This facility locks the pages into memory thus ensuring that critical parts of the program are always resident in memory, thus supplying fast response for time critical applications.

**Kernel Pre-emption**: Provides kernel preemption points in order that a time critical process may preempt a lower priority process running, **even while in kernel mode**

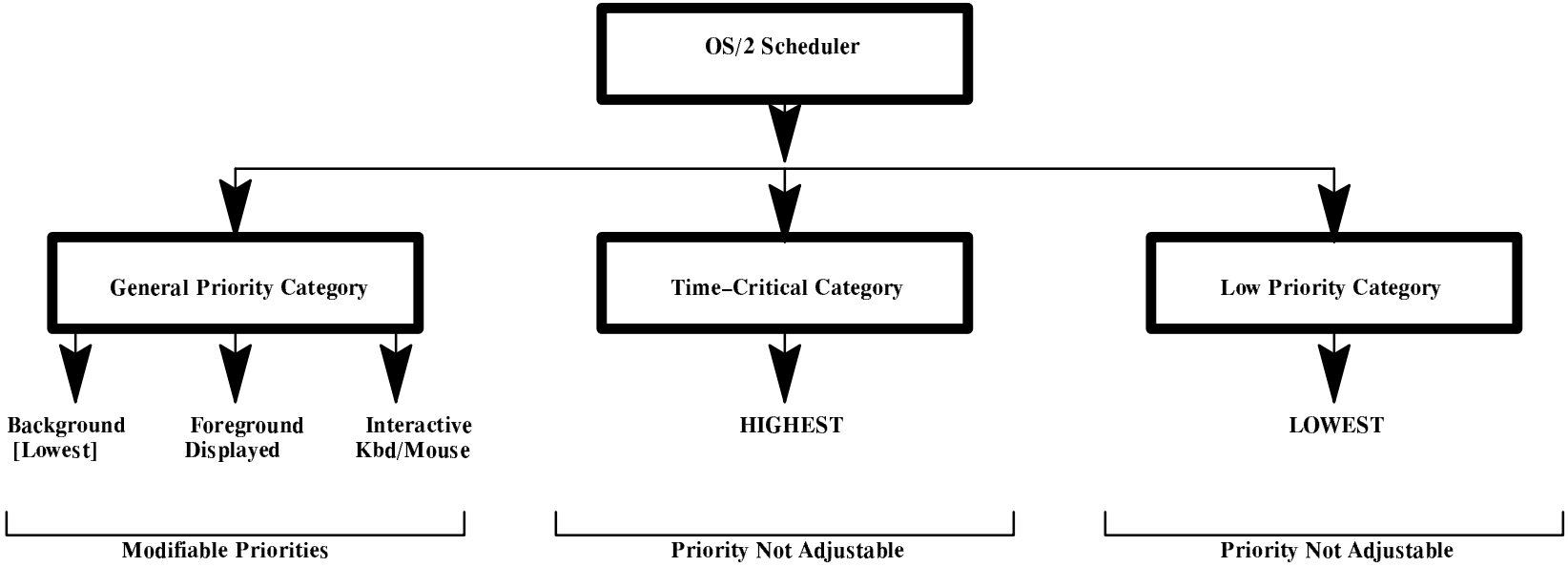
**User Controlled**: The "priority()" system call allows users to simply enter real time mode. [Must be SU]




**Note**: Diagram does not show exact implementation due to source code restrictions but does give some idea of the implementation.

# OS/2 Scheduler

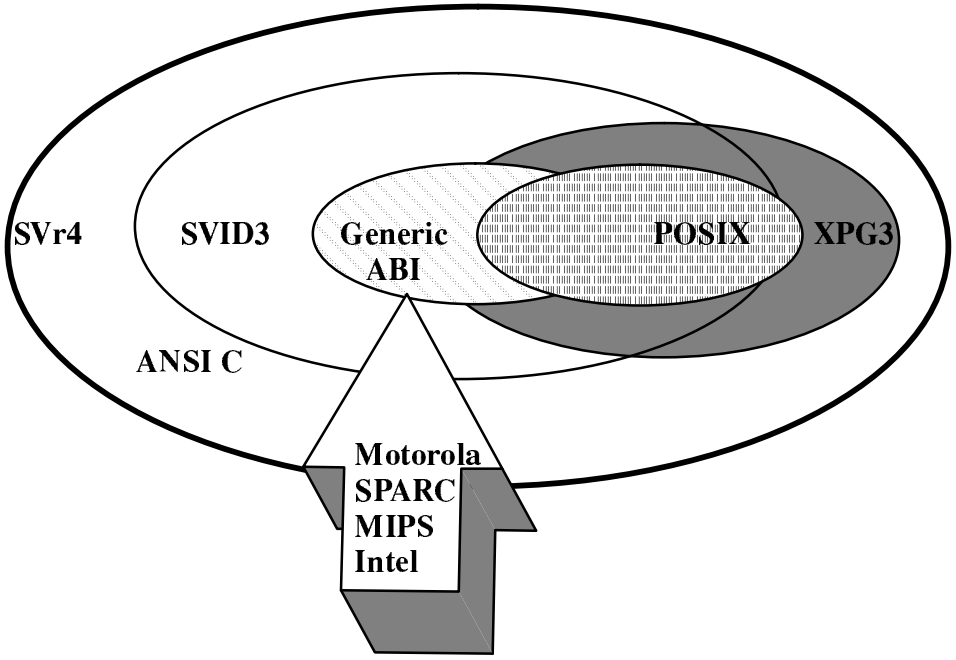
**OS/2** provides conventional multitasking as well as making extensive use of threads. The scheduler is designed to provide maximum user responsiveness as opposed to system throughput. Multitasking is implemented via a mechanism called *priority–based preemptive scheduler.* The highest priority thread runs till completion. If a higher priority thread becomes runable the kernel will be preempted. Threads run in three classes: *general priority, time–critical priority and low priority.* The general class is the most generally used Threads running in this mode belong to one of three sub–classes: *Background, foreground and interactive.* A task in the foreground (program on front of screen) has higher priority priority than background process.Background threads get CPU only when foreground threads are idle. All current active processes are members of the foreground subcategory. The process that is currently interacting with the keyboard or mouse is promoted to the interactive category.Non interactive threads and background threads may get any CPU time for long periods. *Time critical* threads are not subject to adjustment. Not all threads within a process need to be of the same priority. The *low priority* class threads get the CPU only when NO other threads need it. This is not a very useful or popular class but was a added as it came free as a mirror image of the time–critical class. OS/2 provides NO page locking mechanism. Provides for process swapping but NO paging



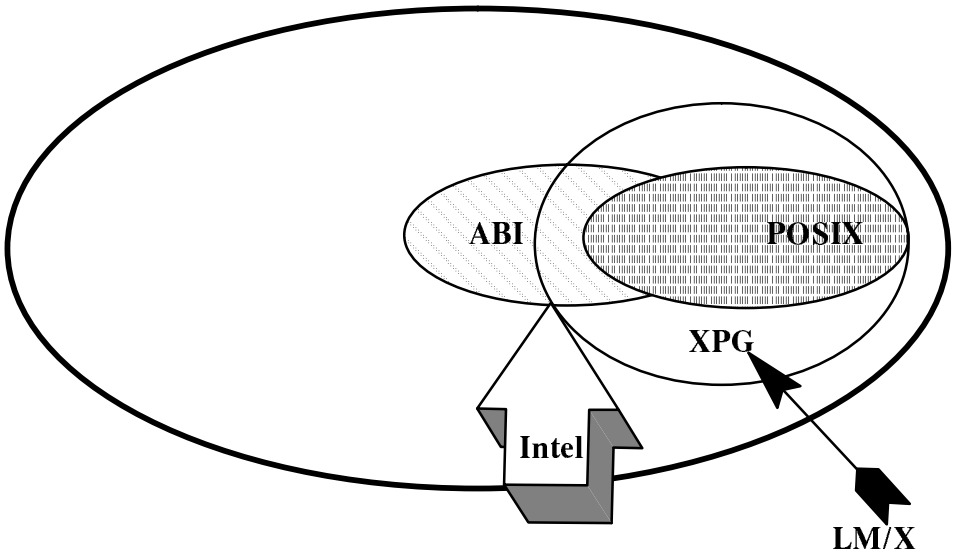

*Note: OS/2 scheduler is optimized for user response as opposed to system throughput. This may NOT be a good solution for Networking responsiveness etc.*

# Conformance with Source Standards

## SVr4




## OS/2




SCHEMATIC ONLY: NOT TO SCALE

# UNIX/SVr4 & OS/2 Comparison

# Sun Microsystems, Inc.

# UNIX/SVr4 & OS/2 Technical Comparison

**Norman Eaglestone**
**Area Technology Manager**
**North East Area Technical Staff**

*Systems for Open Computing*

**For: tlau**
**Printed on: Tue, Jul 14, 1992 10:46:14**

**Document: 6853**
**Last saved on: Tue, Jul 14, 1992 10:46:01**

pl2ps 4.1.0  Copyright 1989 Interleaf, Inc.