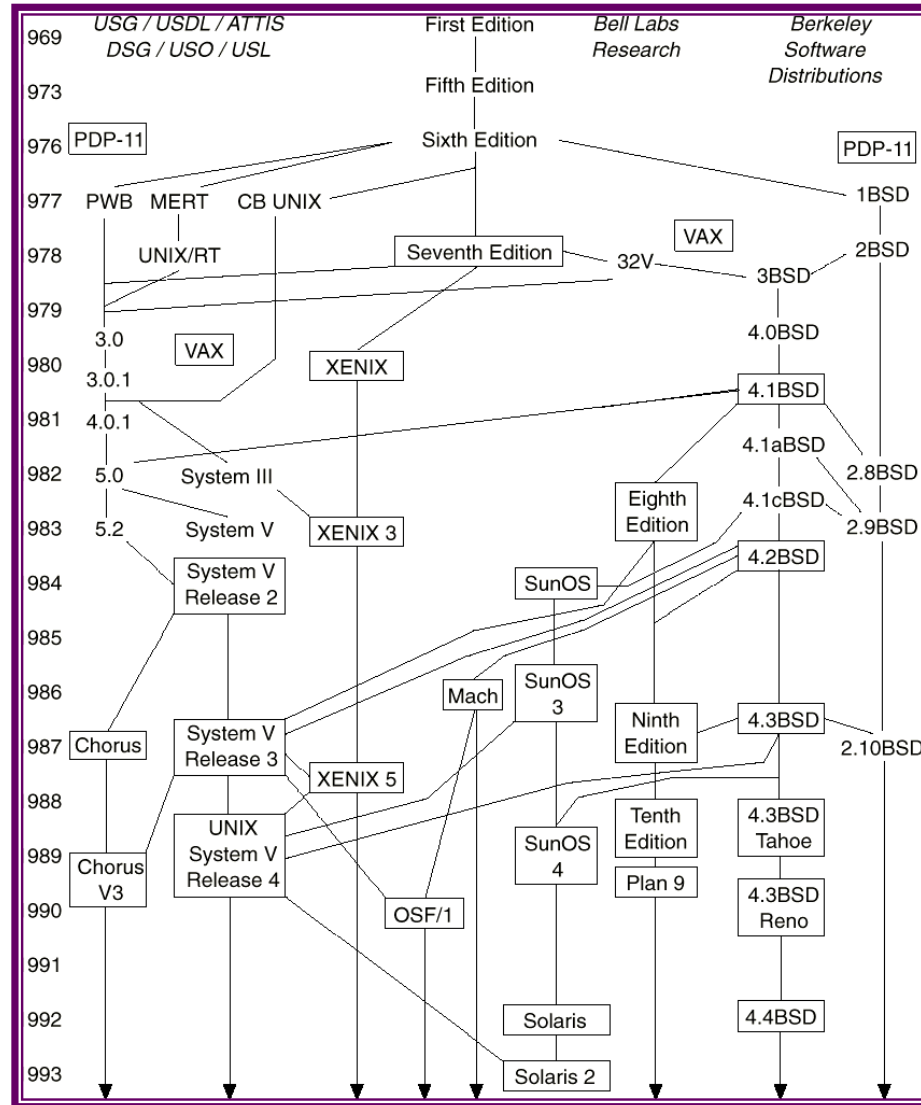


The FreeBSD System


- **History**
- **Design Principles**
- **Programmer Interface**
- **User Interface**
- **Process Management**
- **Memory Management**
- **File System**
- **I/O System**
- **Interprocess Communication**

History of UNIX Versions

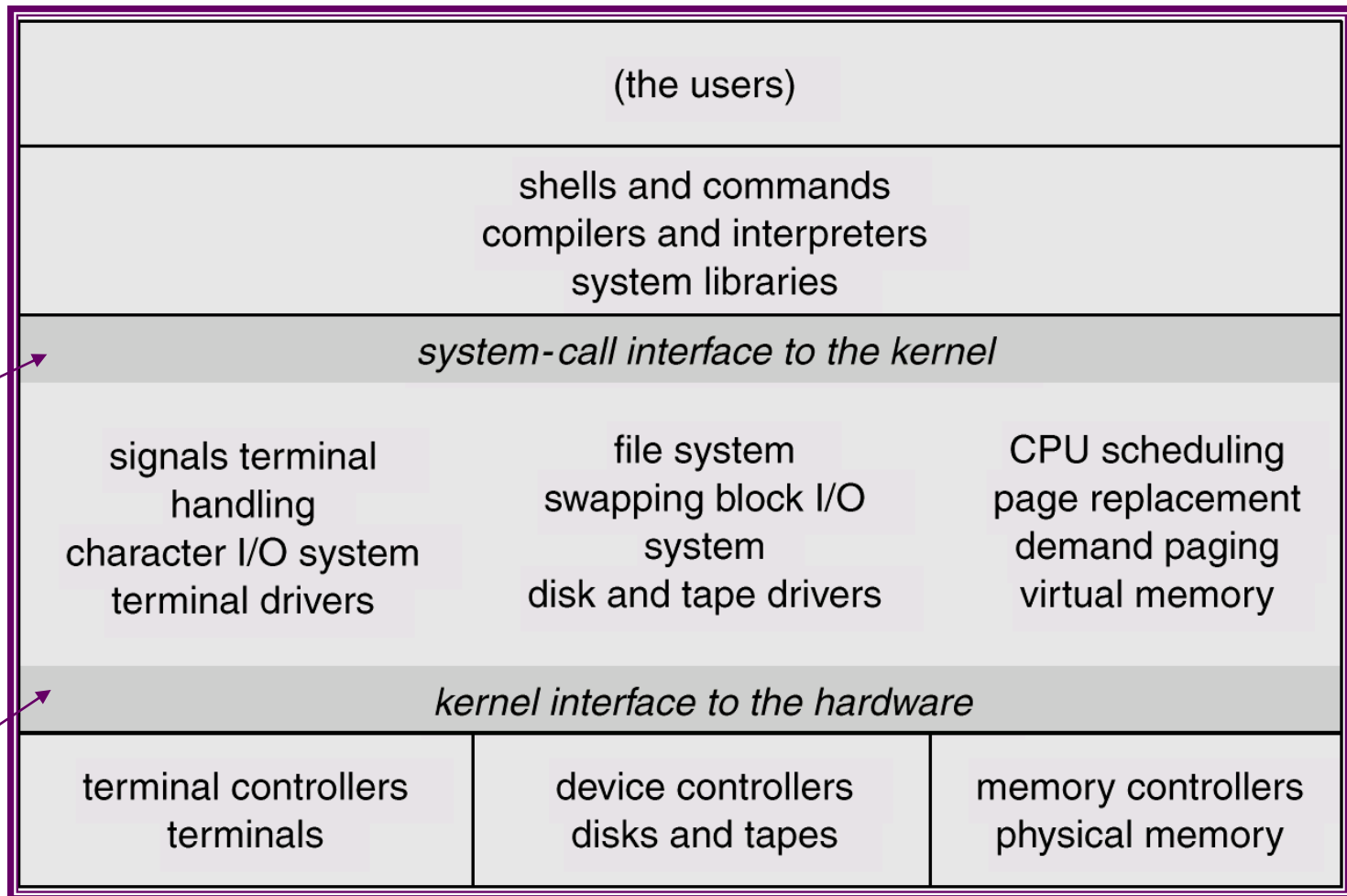


Programmer Interface

Like most computer systems, UNIX consists of two separable parts:

- **1. Kernel:**
- **everything**
 - ☞ **below** the **system-call interface**
 **and**
 - ☞ **above** the **physical hardware.**
- **Provides:**
 - ☞ **file system management**
 - ☞ **process management and CPU scheduling**
 - ☞ **memory management**
 - ☞ **and other OS functions through system calls.**
- **2. Systems programs:**
- **use the kernel-supported system calls**
- **to provide useful functions, such as**
 - ☞ **compilation**
 - ☞ **file manipulation.**

4.4BSD Layer Structure



Typical UNIX Directory Structure

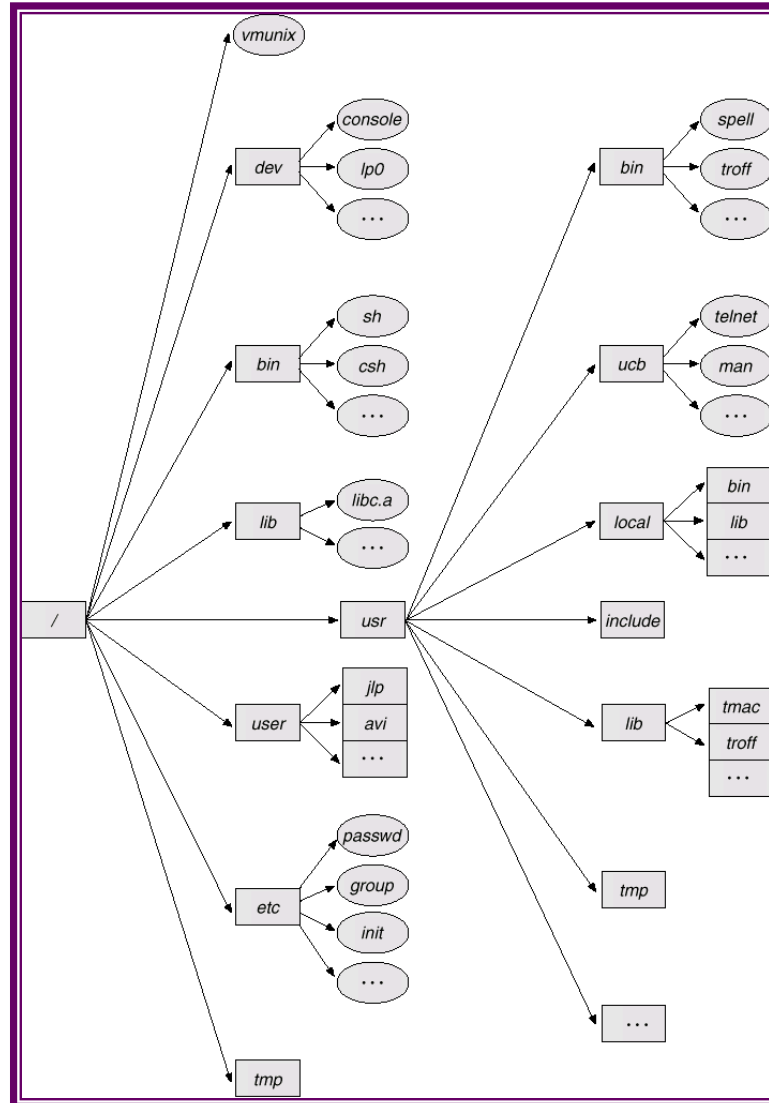
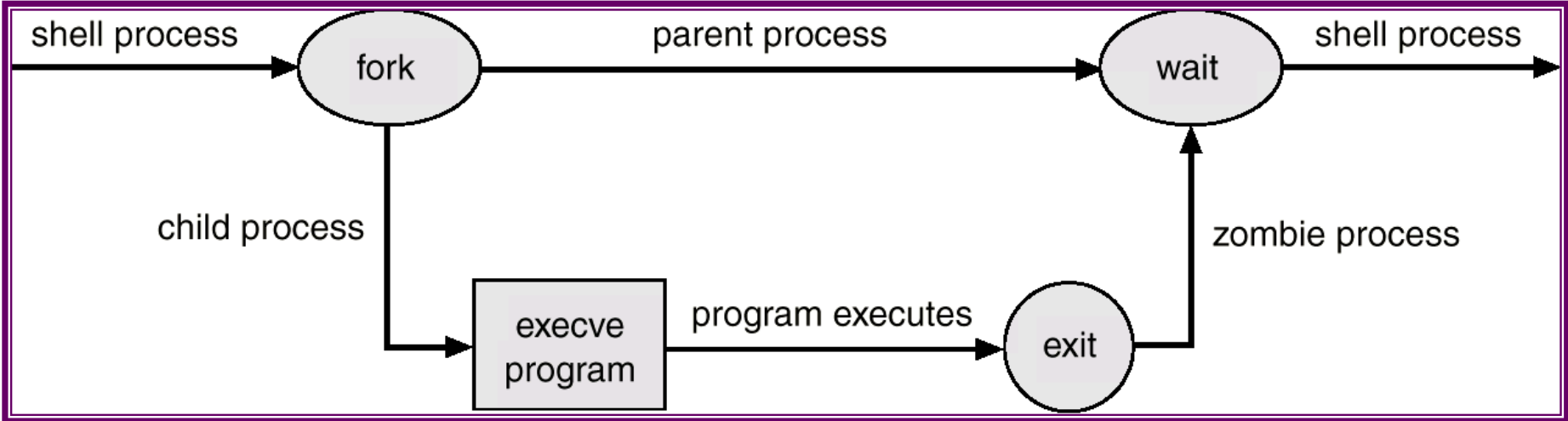


Illustration of Process Control Calls



Shells and Commands (Cont.)

- Typical **search path** on a BSD system:

- ☞ .
- ☞ /home/prof/avi/bin
- ☞ /usr/local/bin
- ☞ /usr/ucb
- ☞ /bin
- ☞ /usr/bin

- The **shell**

- ☞ usually suspends its own execution
- ☞ until the command completes.

- **Foreground process**

- **Background process [&]**

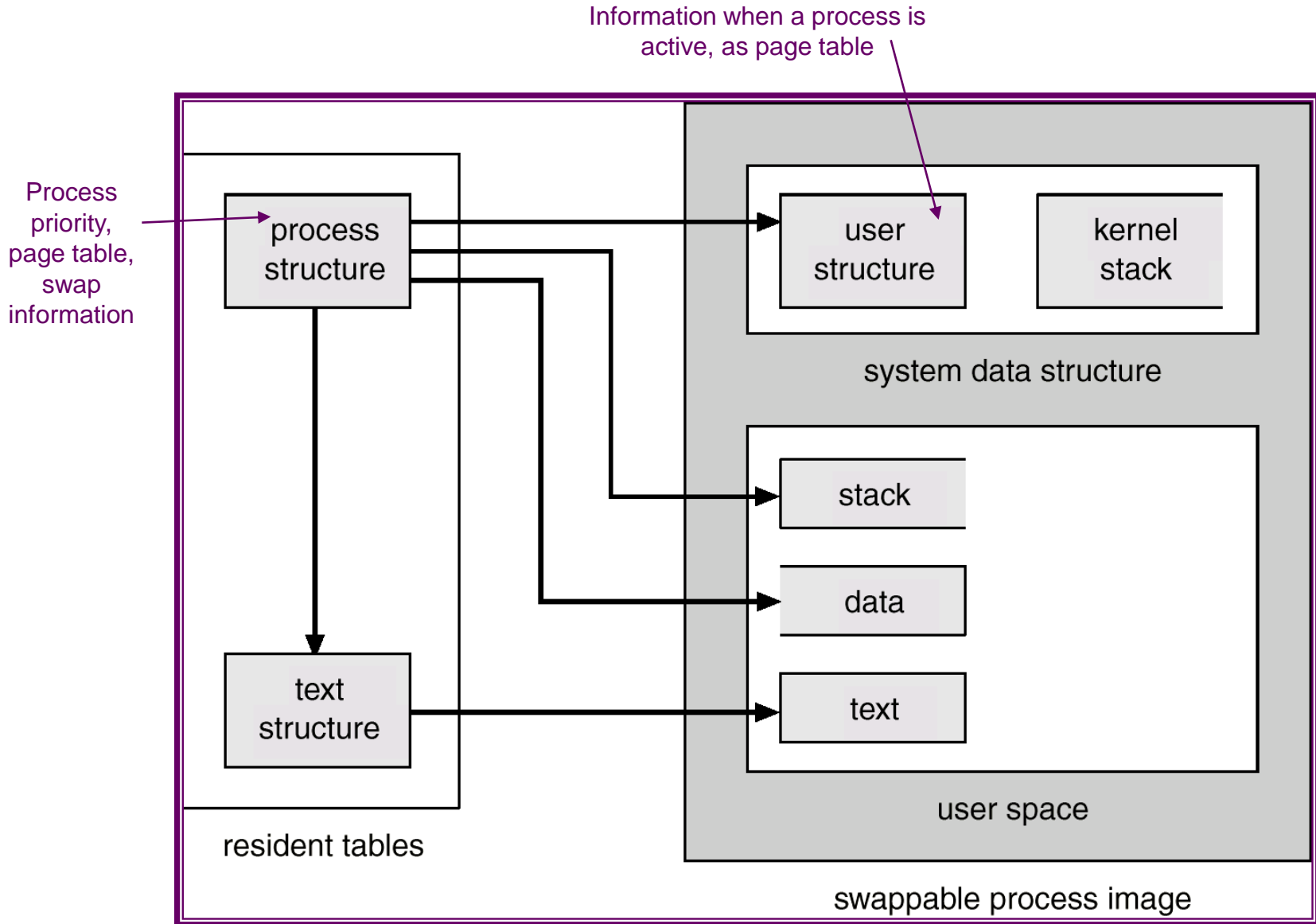
- **IO redirection**

> < 2>n>&m

- **Command pipelining**

\$ ls | pr | lpr

Finding parts of a process using process structure



CPU Scheduling

- **Every process**
 - ☞ has a scheduling priority associated with it;
 - ☞ larger numbers indicate lower priority.
- **Negative feedback in CPU scheduling**
 - ☞ makes it difficult for a single process
 - ☞ to take all the CPU time.
- **Process aging is employed to prevent starvation.**
- **When a process chooses to relinquish the CPU,**
 - ☞ it goes to sleep on an event.
- **When that event occurs,**
 - ☞ the system process that knows about it
 - ☞ calls wakeup with the address corresponding to the event,
 - ☞ and all processes that had done a sleep on the same address
 - ☞ are put in the ready queue to be run.

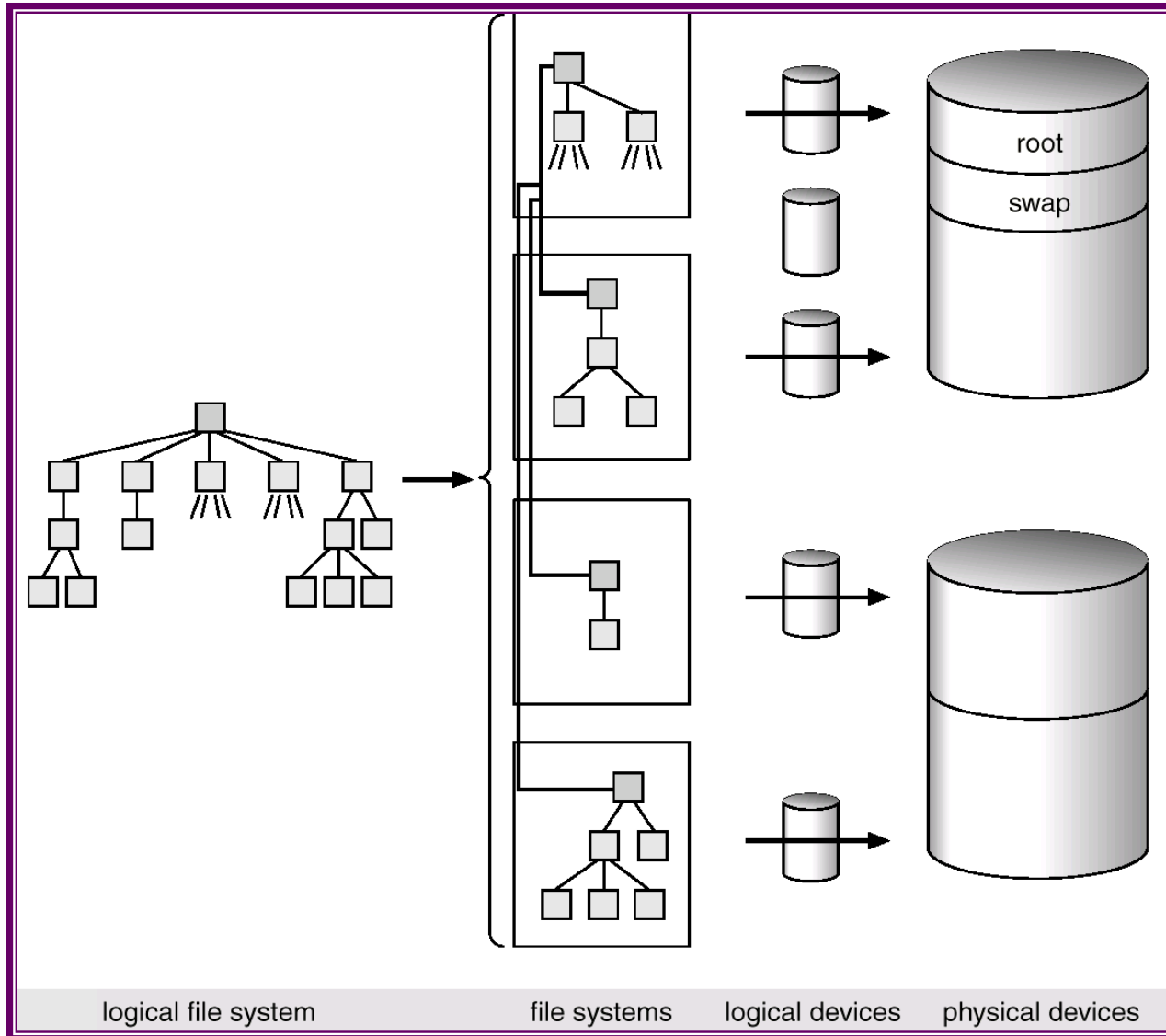
Memory Management

- **The initial memory management schemes**
 - ☞ were constrained in size
 - ☞ by the relatively small memory resources
 - ☞ of the PDP machines
 - ☞ on which UNIX was developed.
- **Pre 3BSD system**
 - ☞ use swapping exclusively
 - ☞ to handle memory contention among processes:
 - ☞ If there is too much contention,
 - ☞ processes are swapped out
 - ☞ until enough memory is available.
- **Allocation of both**
 - ☞ main memory and swap space
 - ☞ is done first-fit.

Paging-Virtual memory

- **Berkeley UNIX systems**
- **depend primarily on paging**
 - ☞ for memory-contention management,
 - ☞ and
- **depend only secondarily on swapping.**
- **Demand paging** –
 - ☞ when a process needs a page and
 - ☞ the page is not there,
 - ☞ a page fault to the kernel occurs,
 - ☞ a frame of main memory is allocated, and
 - ☞ the proper disk page is read into the frame.
- **A pagedaemon process uses**
 - ☞ a modified second-chance page-replacement algorithm
 - ☞ to keep enough free frames
 - ☞ to support the executing processes.
- **If the scheduler decides**
 - ☞ that the paging system is overloaded,
 - ☞ processes will be swapped out whole
 - ☞ until the overload is relieved.

Mapping File System to Physical Devices



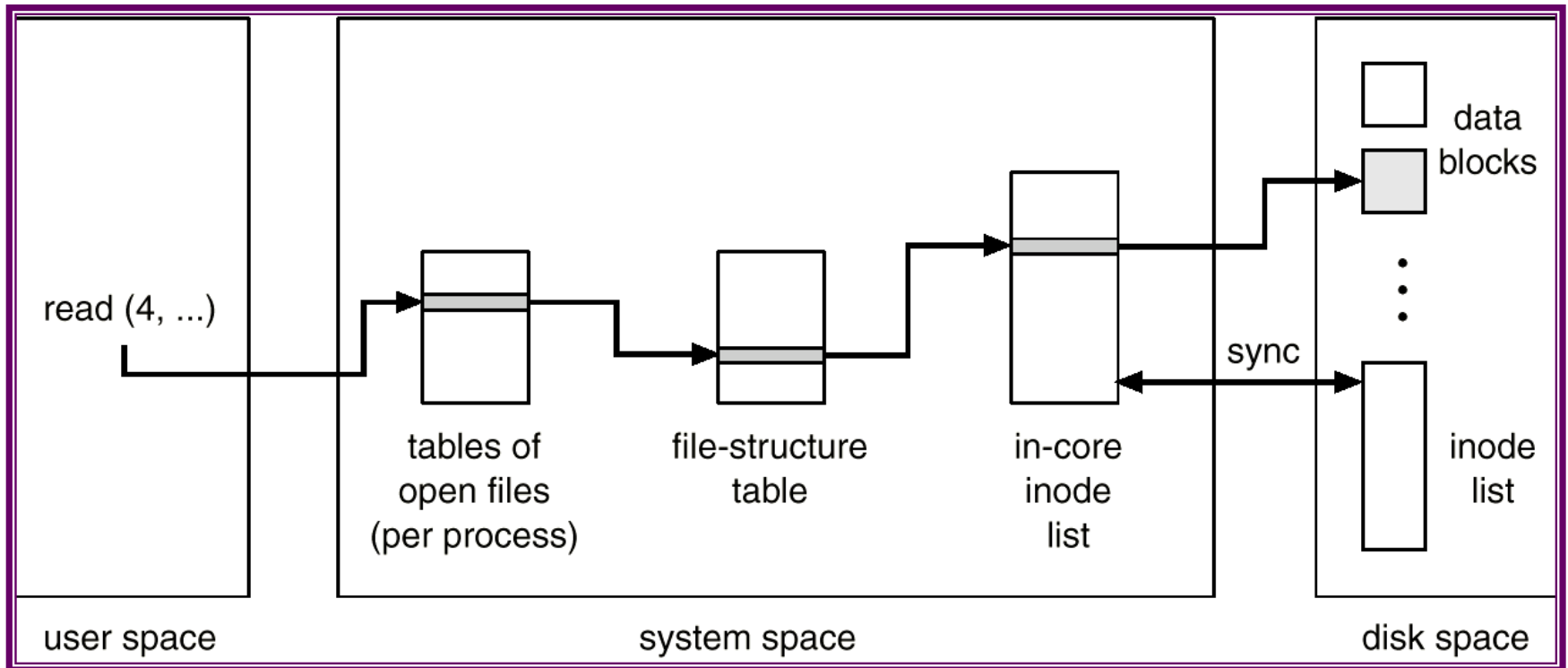
File Systems: Blocks and Fragments

- Most of the file system is taken up
 - ☞ by data blocks.
- 4.2BSD uses two block sized methods for small files
 - ☞ All the blocks of a file
 - ☞ are of a large block size (such as 8K),
 - ☞ except the last.
 - ☞ The last block is
 - ☞ an appropriate multiple of a smaller fragment size
 - ☞ (i.e., 1024) to fill out the file.
 - ☞ Thus, a file of size 18,000 bytes would have
 - ☞ two 8K blocks
 - ☞ and
 - ☞ one 2K fragment (which would not be filled completely).
- The maximum **block-to-fragment ratio** is 8 : 1;
 - ☞ the minimum block size is 4K
 - ☞ typical choices are
 - ☞ 4096 : 512
 - ☞ and
 - ☞ 8192 : 1024.

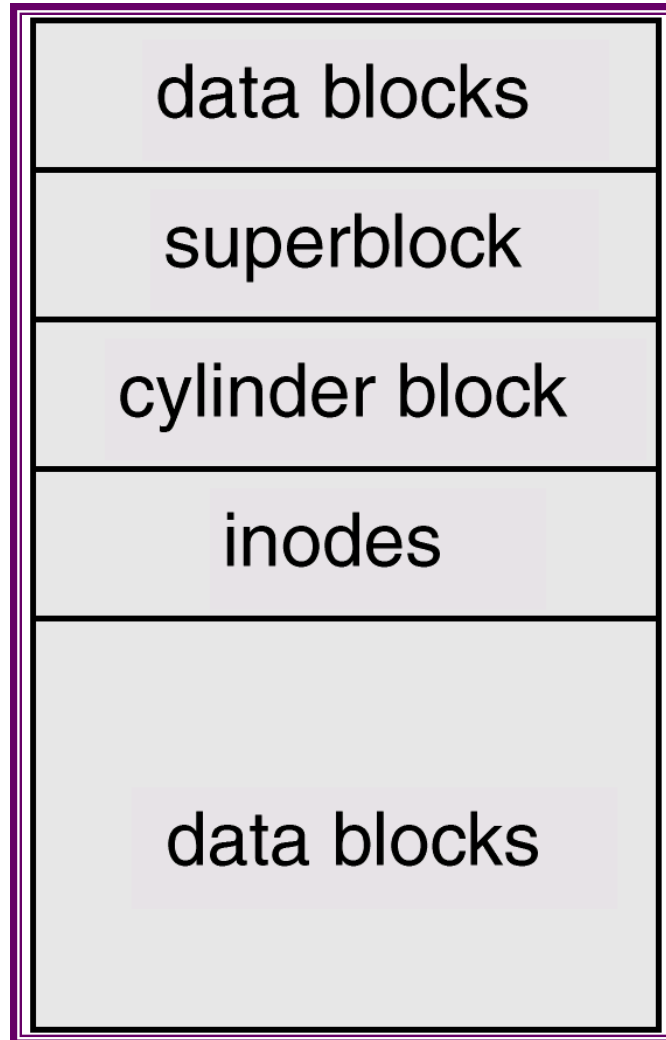
Inodes

- A file is represented by an inode —
 - ☞ a record that stores information
 - ☞ about a specific file on the disk.
- The inode also contains 15 pointer
 - ☞ to the disk blocks
 - ☞ containing the file's data contents.
- First 12 point to direct blocks.
- Next three point to indirect blocks
 - ☞ First is a indirect block pointer
 - 📄 is the address of a single indirect block —
 - 📄 an index block containing the addresses of blocks
 - 📄 that do contain data.
 - ☞ Second is a double-indirect-block pointer,
 - 📄 the address of a block
 - 📄 that contains the addresses of blocks
 - 📄 that contain pointer to the actual data blocks.
 - ☞ A **triple indirect pointer is not needed;**
 - 📄 files with as many as 2^{32} bytes
 - 📄 will use only double indirection.

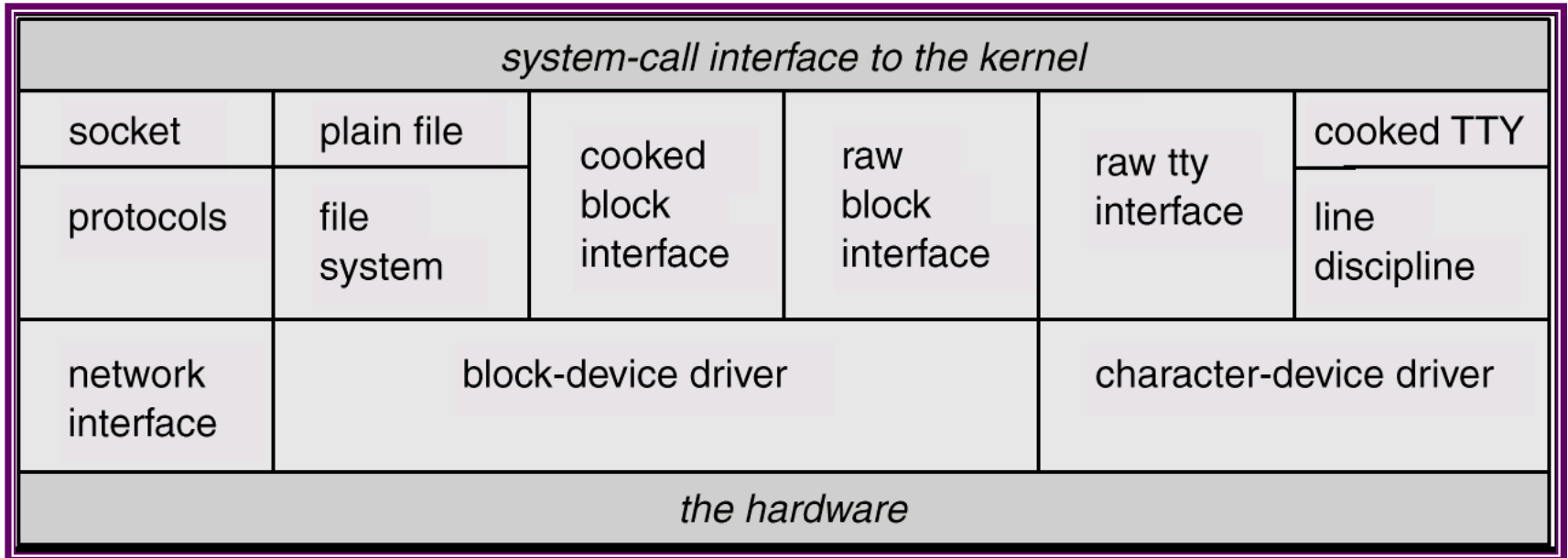
File-System Control Blocks







4.3BSD Cylinder Group



I/O Structure



Interprocess Communication

- The **pipe** is the **IPC** mechanism most characteristic of **UNIX**.
 - ☞ Permits a **reliable unidirectional byte stream**
 -  **between two processes (using a file)**
 - ☞ A **benefit of pipes small size**
 -  **is that pipe data are seldom written to disk;**
 -  **they usually are kept in memory**
 -  **by the normal block buffer cache.**
- In **4.3BSD**, pipes are implemented
- as a **special case of the socket mechanism**
 - ☞ which provides a **general interface**
 - ☞ not only to facilities such as pipes,
 - ☞ which are local to one machine,
 - ☞ but also to networking facilities.
- The **socket mechanism**
 - ☞ can be used
 - ☞ by unrelated processes.

Network Support

- **Networking support**
- **is one of the most important features in 4.3BSD.**
- **The **socket concept** provides**
 - ☞ **the programming mechanism**
 - ☞ **to access other processes,**
 - ☞ **even across a network.**
- **Sockets provide an interface to several sets of protocols.**
- **Almost all current UNIX systems support UUCP.**
- **4.3BSD supports the DARPA Internet protocols**
 - ☞ **UDP, TCP, IP, and ICMP**
 - ☞ **on a wide range of Ethernet, token-ring, and ARPANET interfaces.**
- **The 4.3BSD networking implementation,**
 - ☞ **is more oriented**
 - ☞ **toward the ARPANET Reference Model (ARM)**