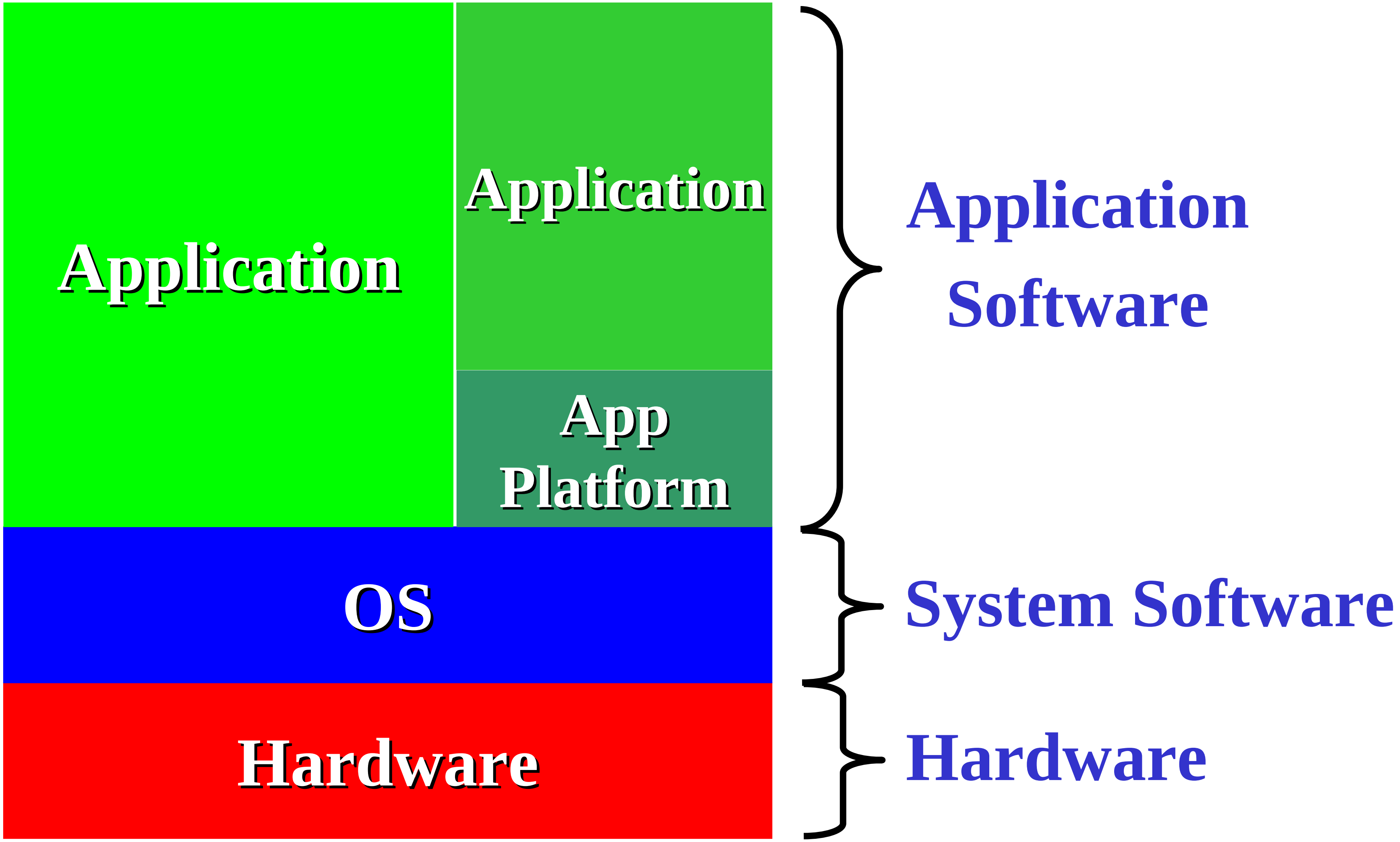
Chapter 1

Introduction

1.1 What is an operating system
1.2 History of operating systems
1.3 The operating system zoo
1.4 Computer hardware review
1.5 Operating system concepts
1.6 System calls
1.7 Operating system structure

Introduction



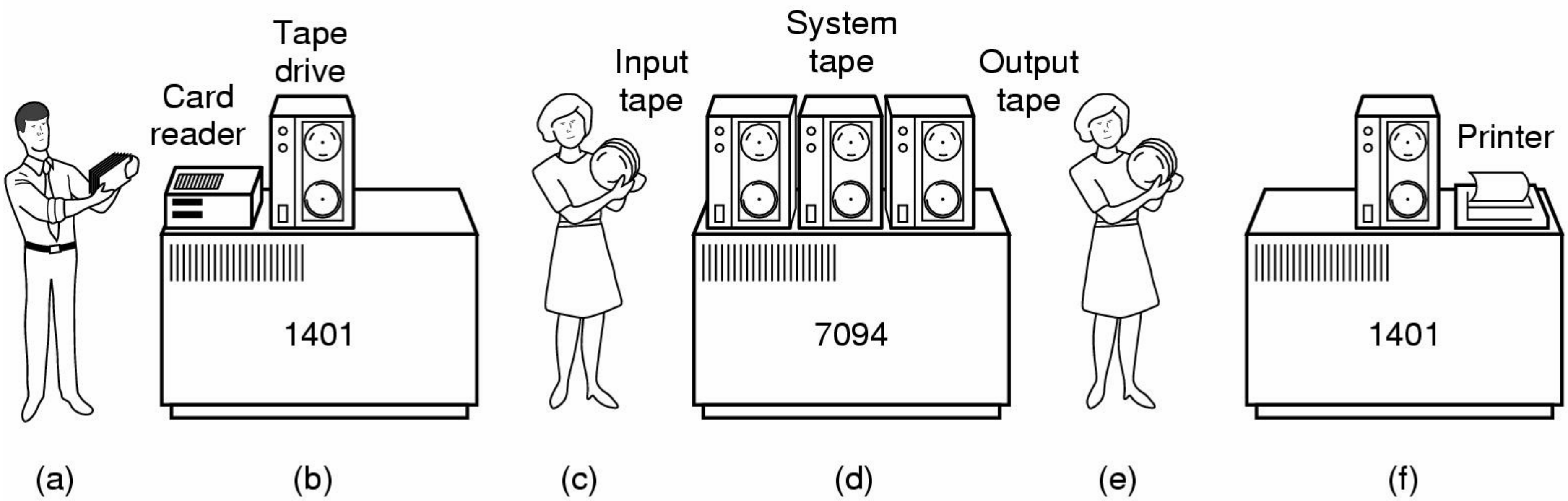
What is an Operating System

- It is an extended machine
  - Hides the messy details which must be performed
  - Presents user with a virtual machine, easier to use
- It is a resource manager
  - Each program gets time with the resource
  - Each program gets space on the resource

History of Operating Systems

- **First generation 1945 - 1955**
  - vacuum tubes, plug boards
- **Second generation 1955 - 1965**
  - transistors, batch systems
- **Third generation 1965 – 1980**
  - ICs and multiprogramming
- **Fourth generation 1980 – present**
  - personal computers

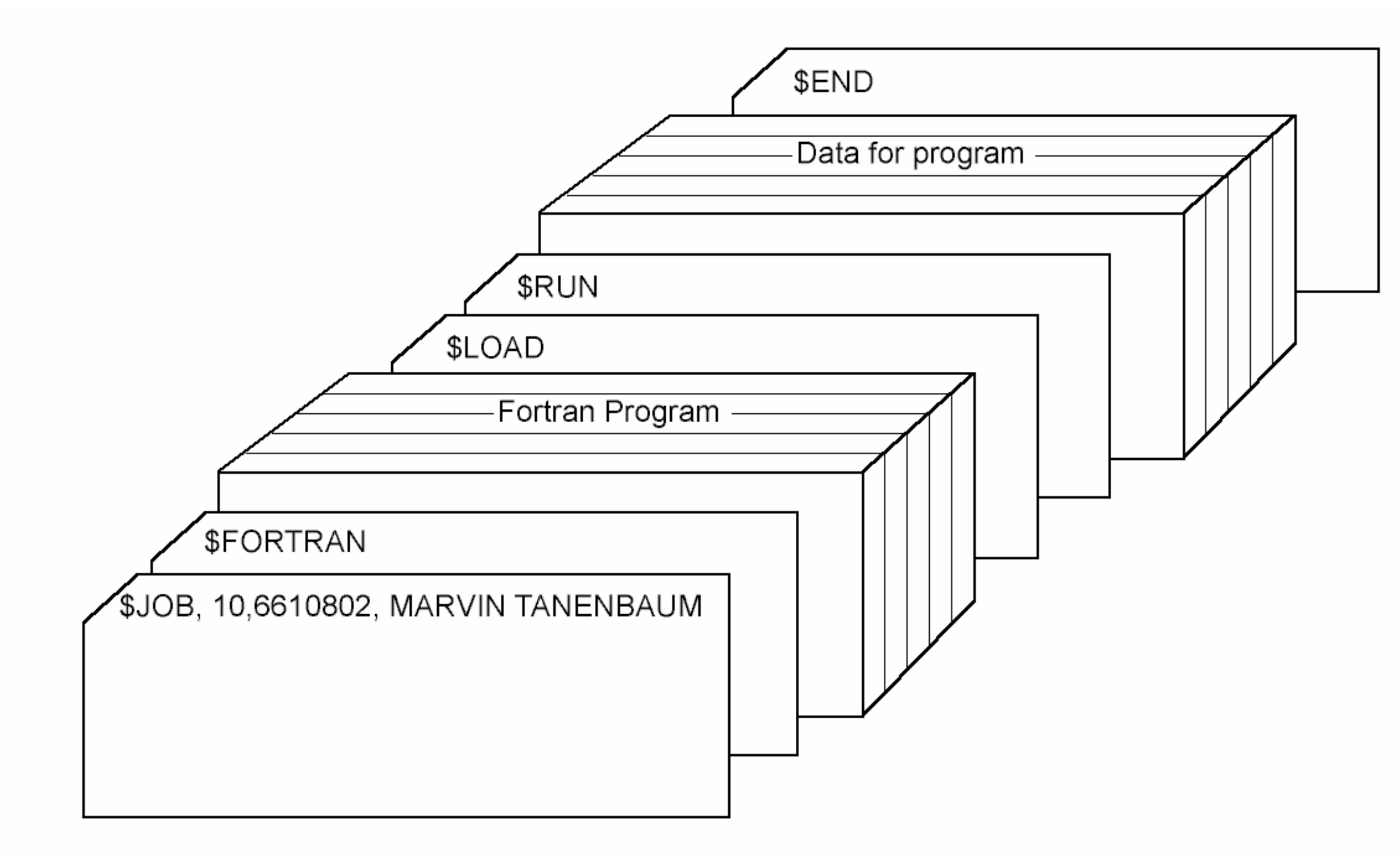
Batch Systems (批处理)



Early batch system

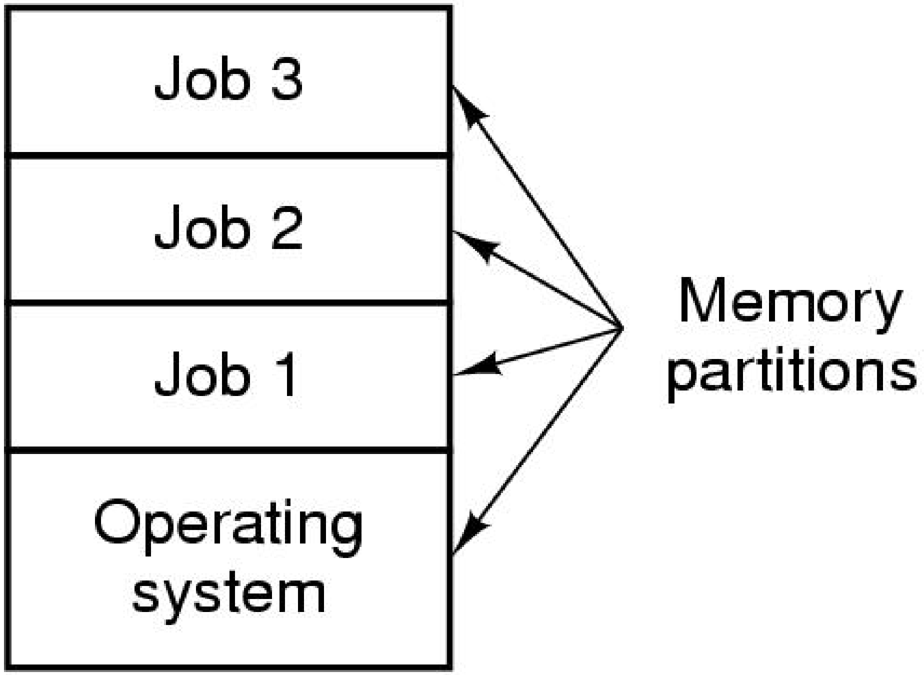
- bring cards to 1401
- read cards to tape
- put tape on 7094 which does computing
- put tape on 1401 which prints output

Cards



- **Structure of a typical FMS job – 2nd generation**

Multiprogramming Systems (多道程序)

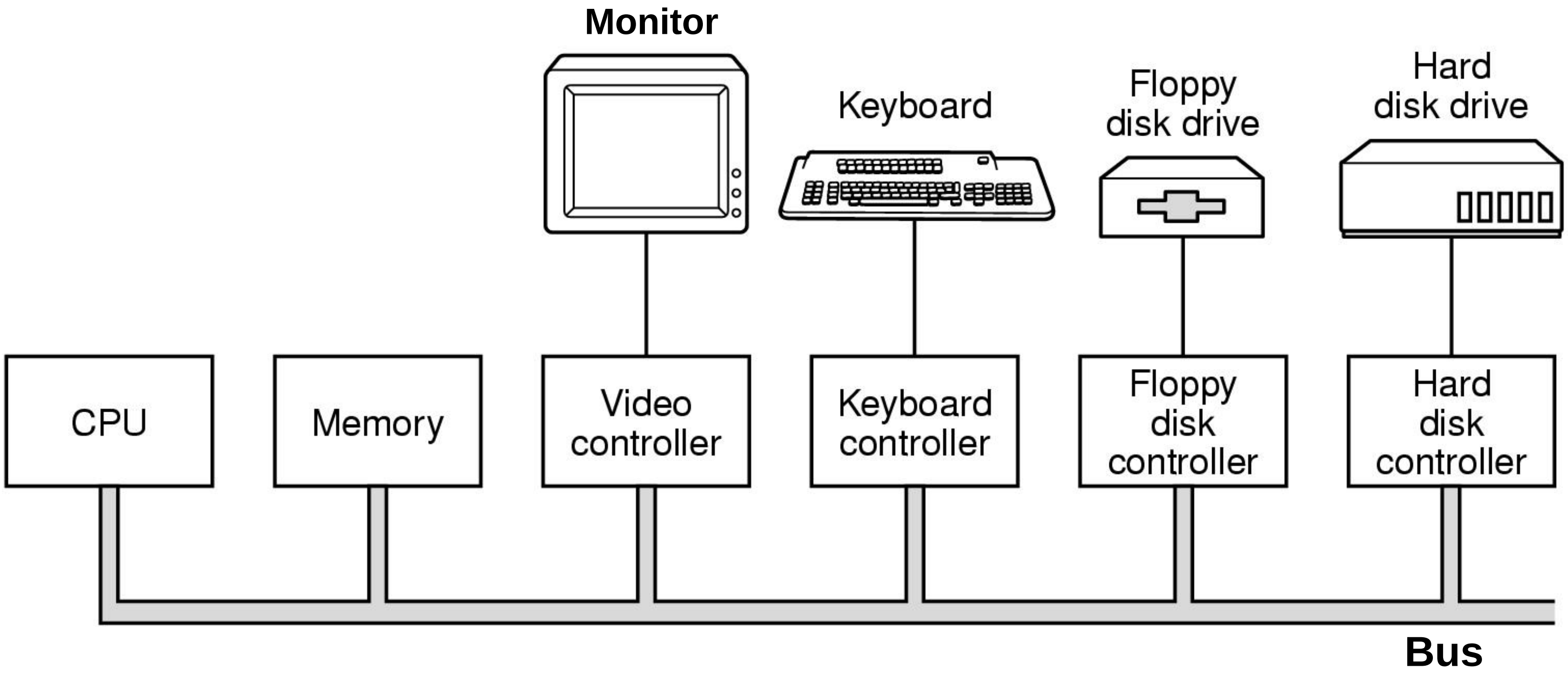


- **Multiprogramming system**
  - **three jobs in memory – 3rd generation**

The Operating System Zoo

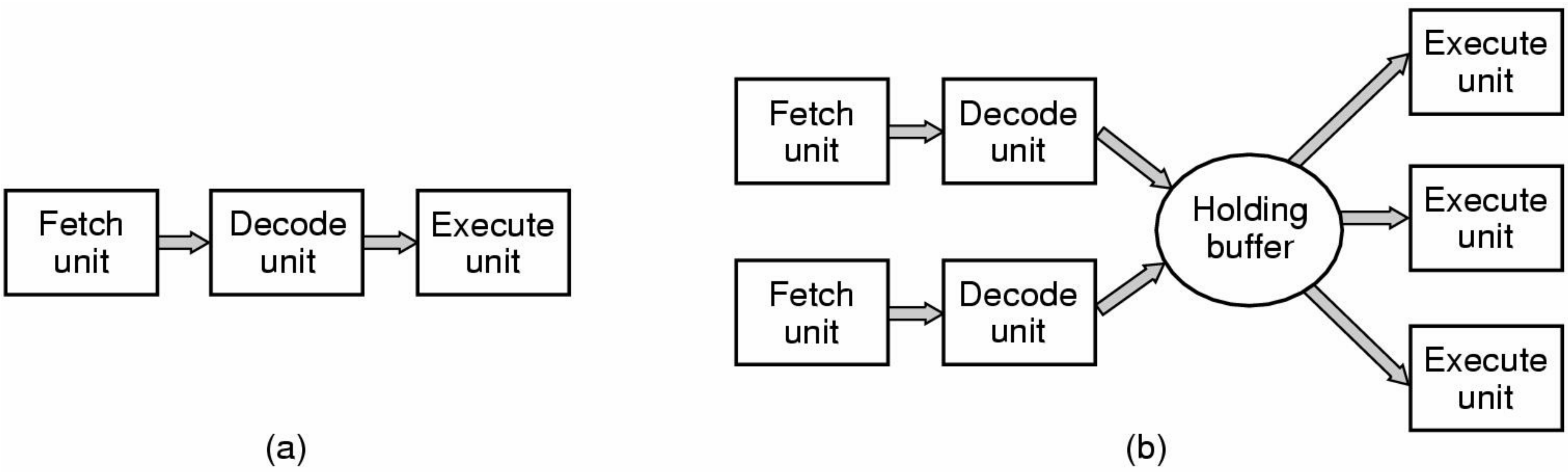
- Mainframe operating systems
- Server operating systems
- Multiprocessor operating systems
- Personal computer operating systems
- Real-time operating systems
- Embedded operating systems
- Smart card operating systems

Computer Hardware Review



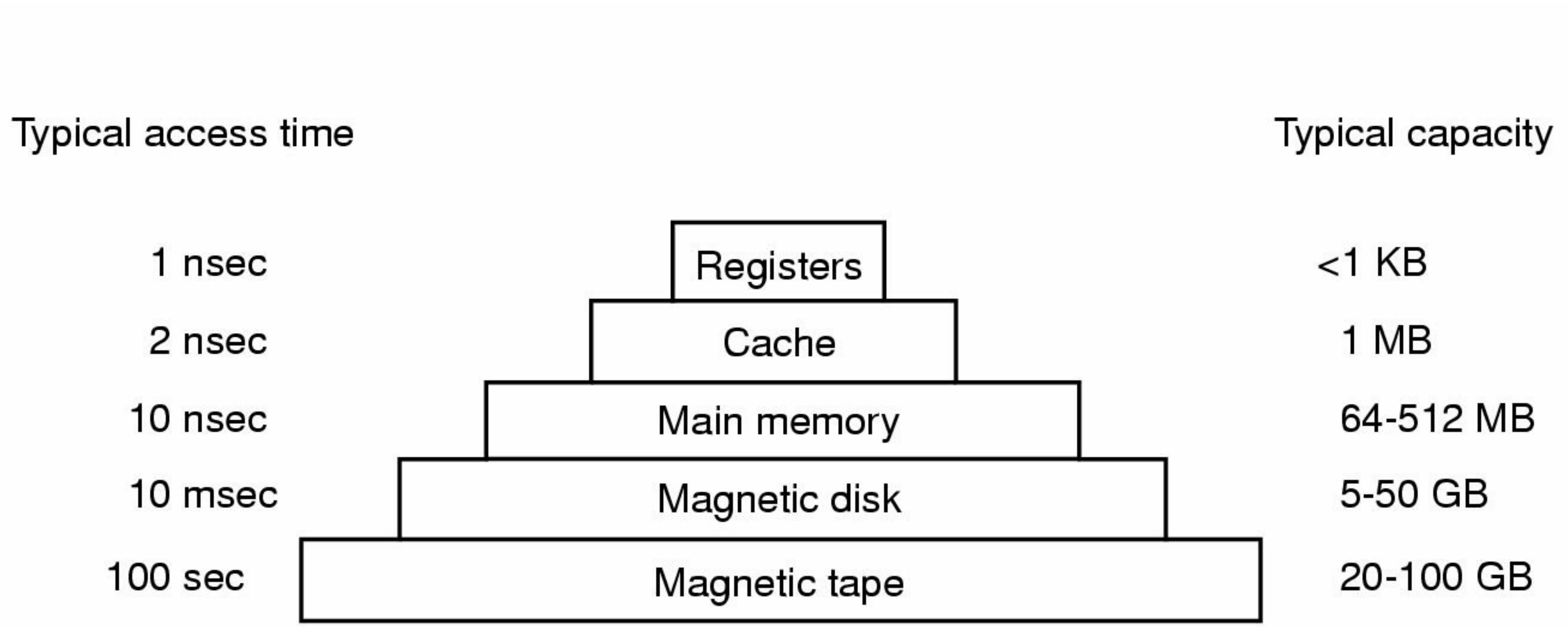
- **Components of a simple personal computer**

How do Instructions Run in CPU



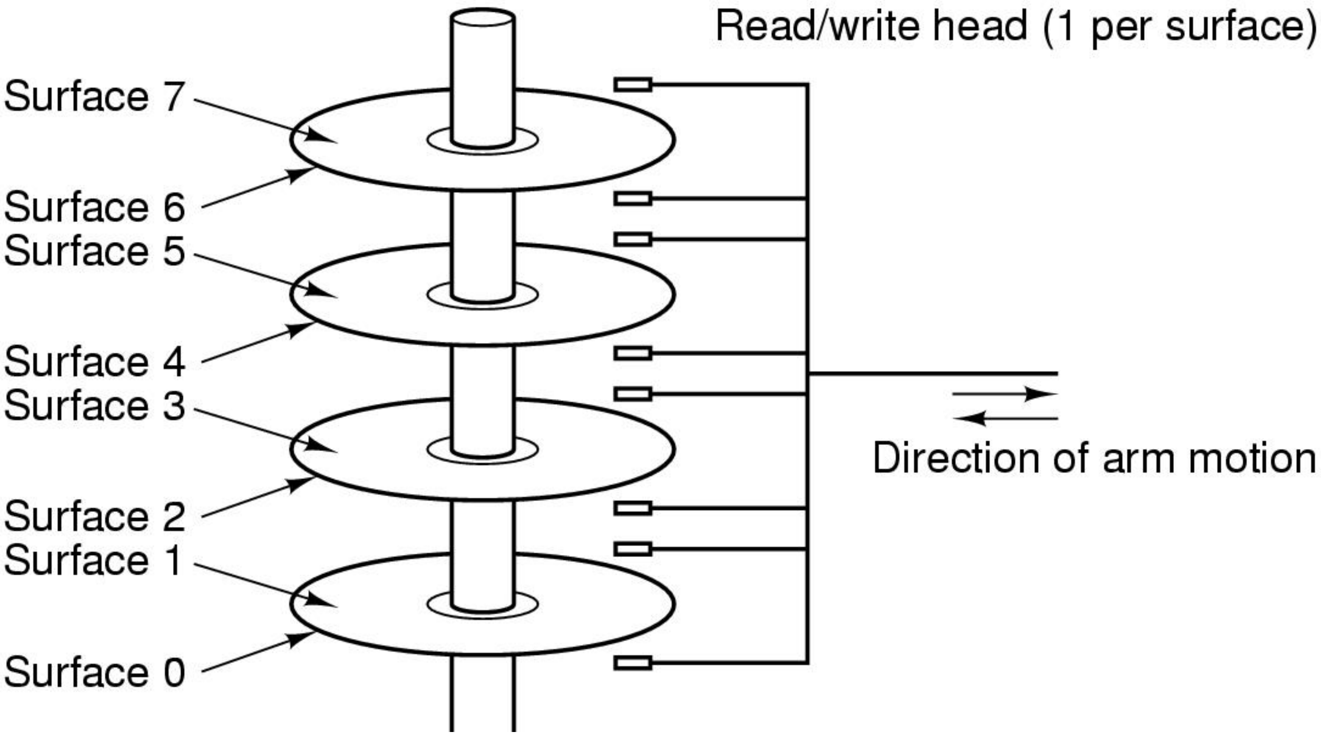
(a) A three-stage pipeline
(b) A superscalar CPU

Memory (存储器)



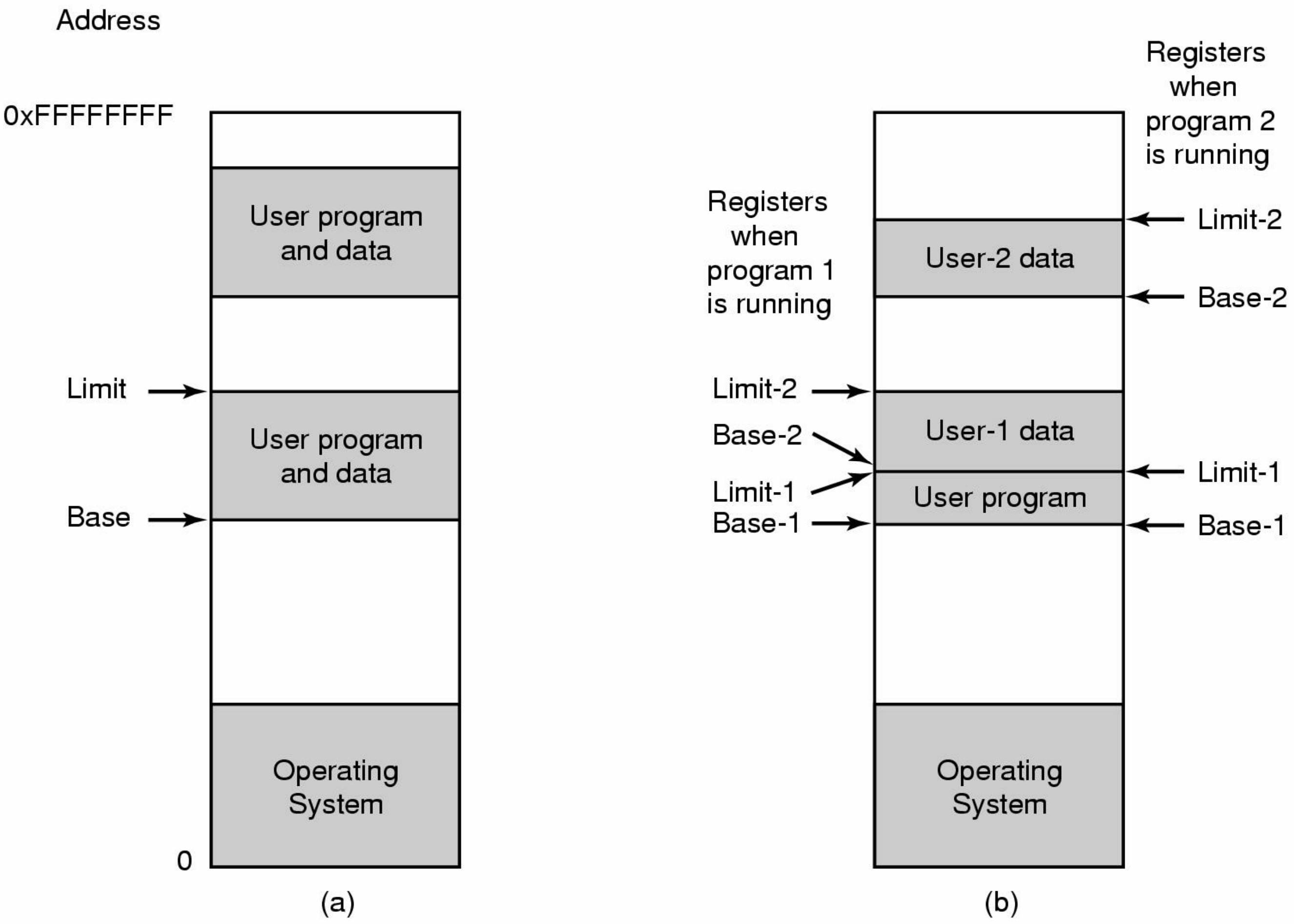
- **Typical memory hierarchy**
  - **numbers shown are rough approximations**

Disks



Structure of a disk drive

Programs in Main Memory



One base-limit pair and two base-limit pairs

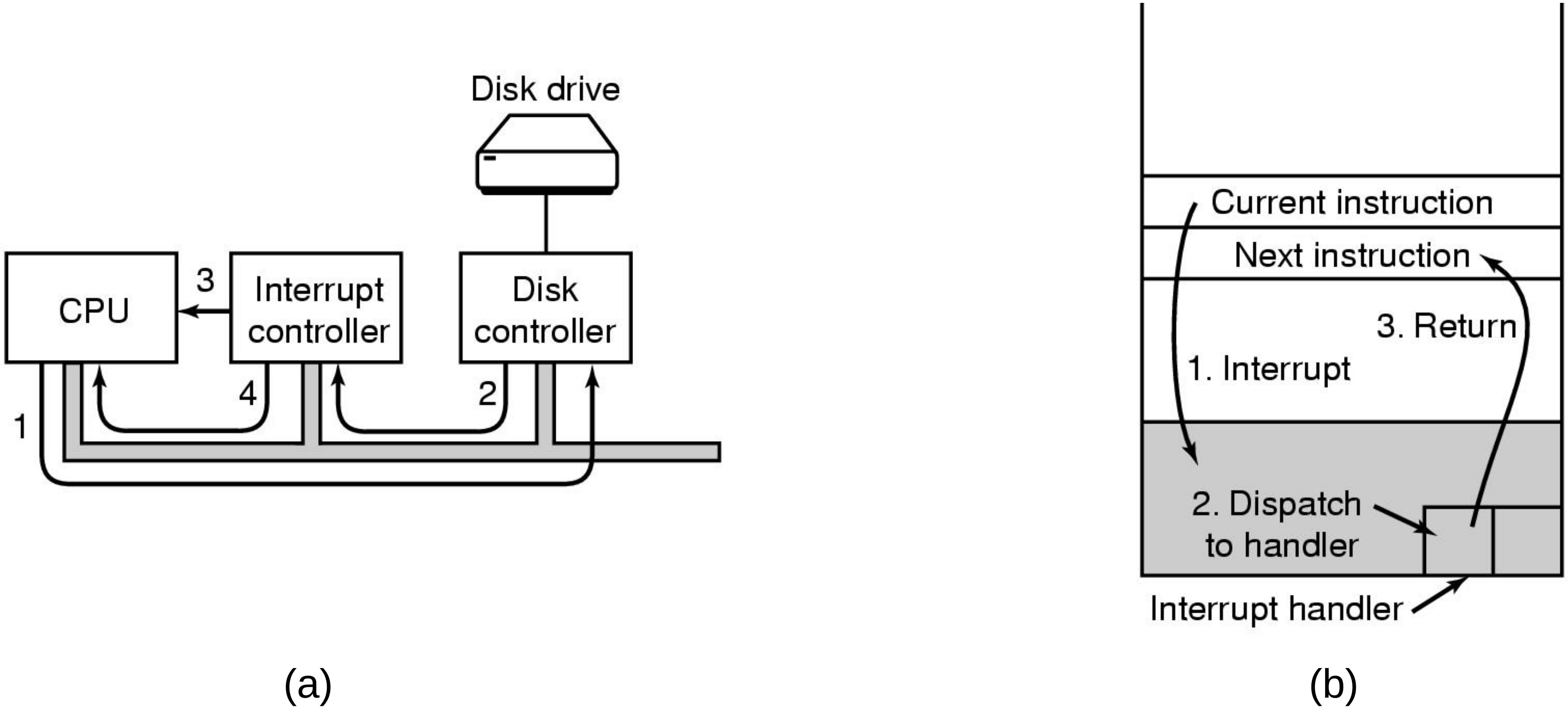
I/O Devices

- **I/O devices generally consist of a *controller* and the *device* itself**
- **OS manages many kinds of I/O devices (controllers)**
- **A kind of device has many models from different manufacturers**
- **So, device drivers!**

Device Drivers

- The software that talks to a controller, giving it commands and accepting responses is called a **device driver**
- Provided by controller manufacturer for each OS
- Works in the OS kernel by
  - Relink and reboot
  - Tell and reboot
  - On-the-fly
- Talks with controller through
  - Busy waiting
  - Interrupt
  - DMA

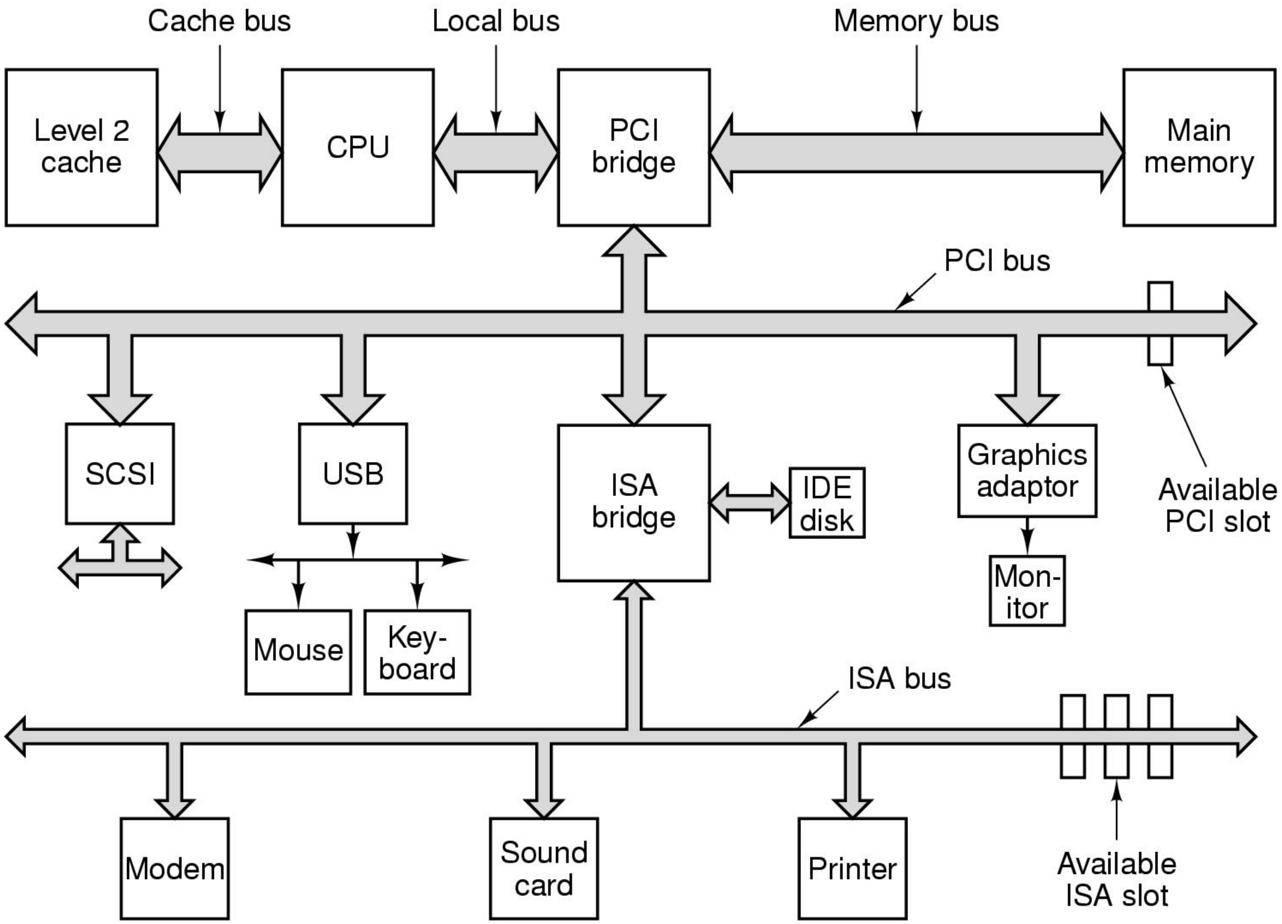
I/O and Interrupt (I/O和中断)



(a) Steps in starting an I/O device and getting interrupt

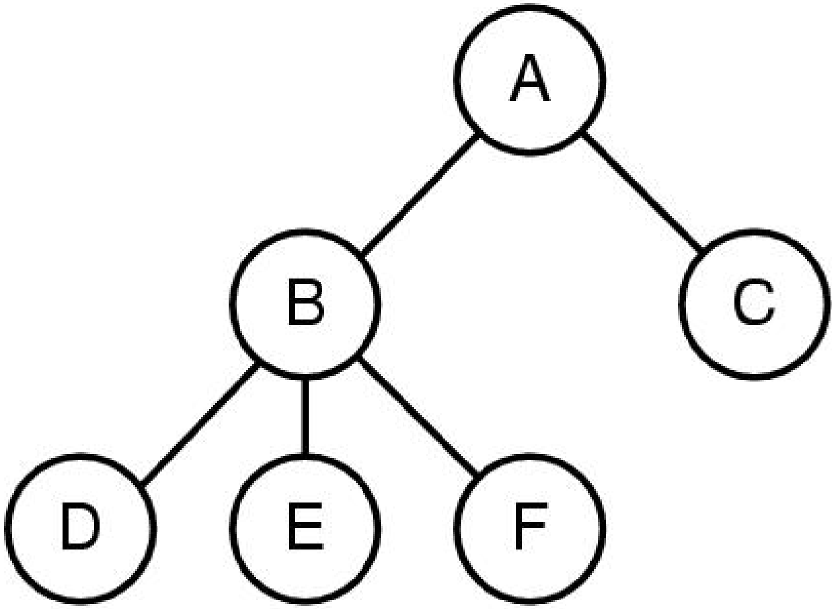
(b) How the CPU is interrupted

Buses (总线)



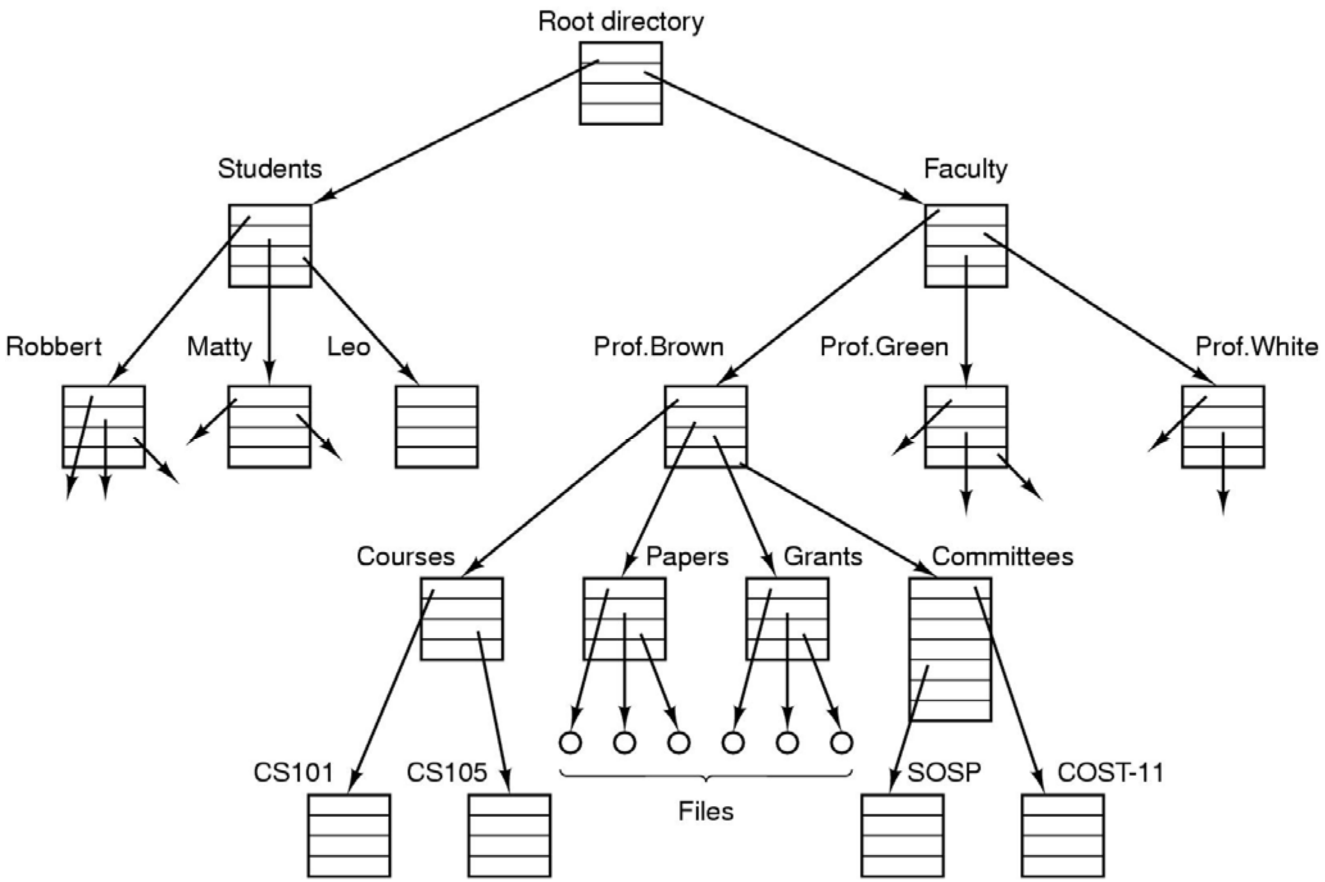
Structure of a large Pentium system

Operating System Concepts Processes (进程)



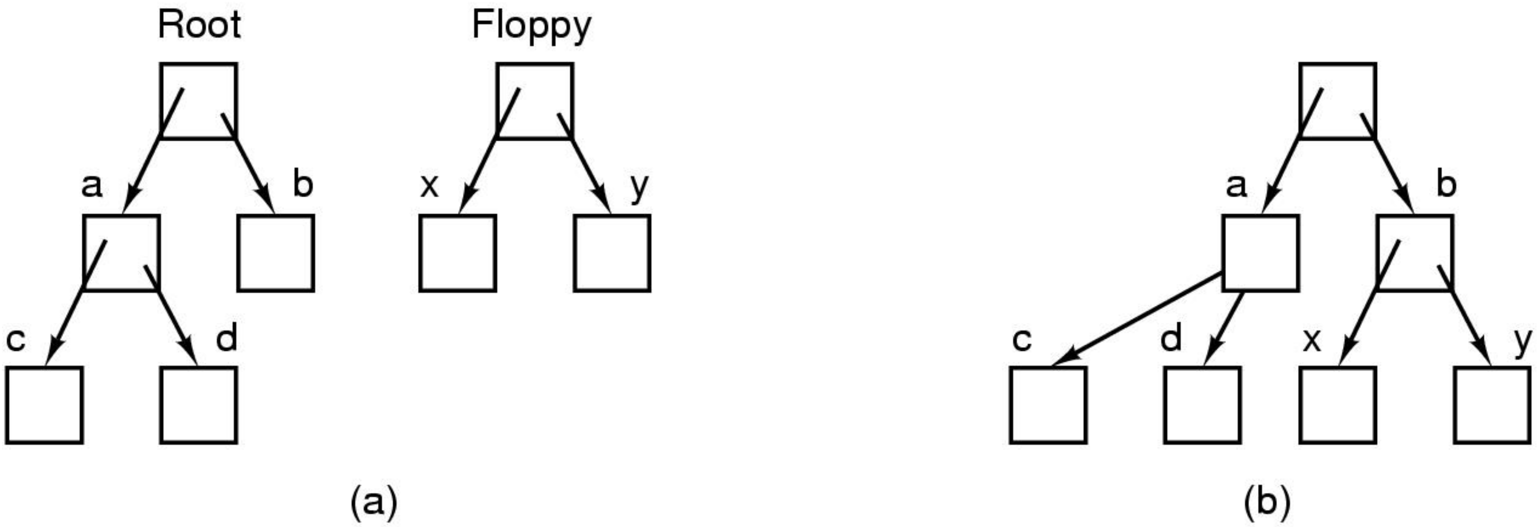
- A process tree
  - A created two child processes, B and C
  - B created three child processes, D, E, and F

Files (文件)



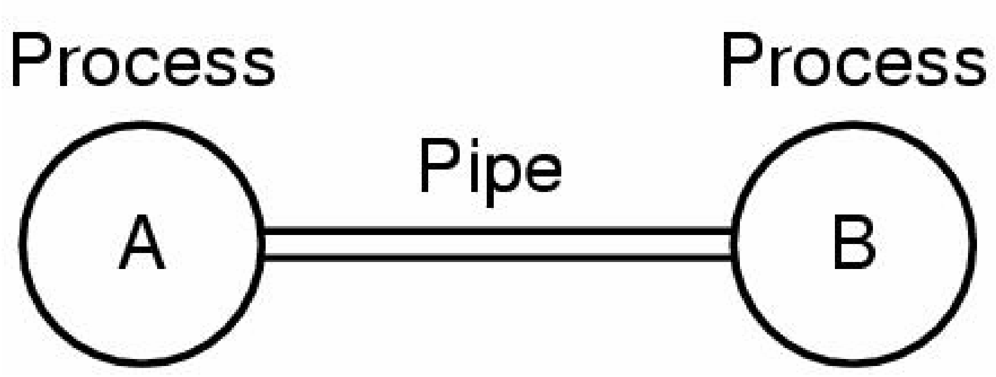
File system for a university department

Mount



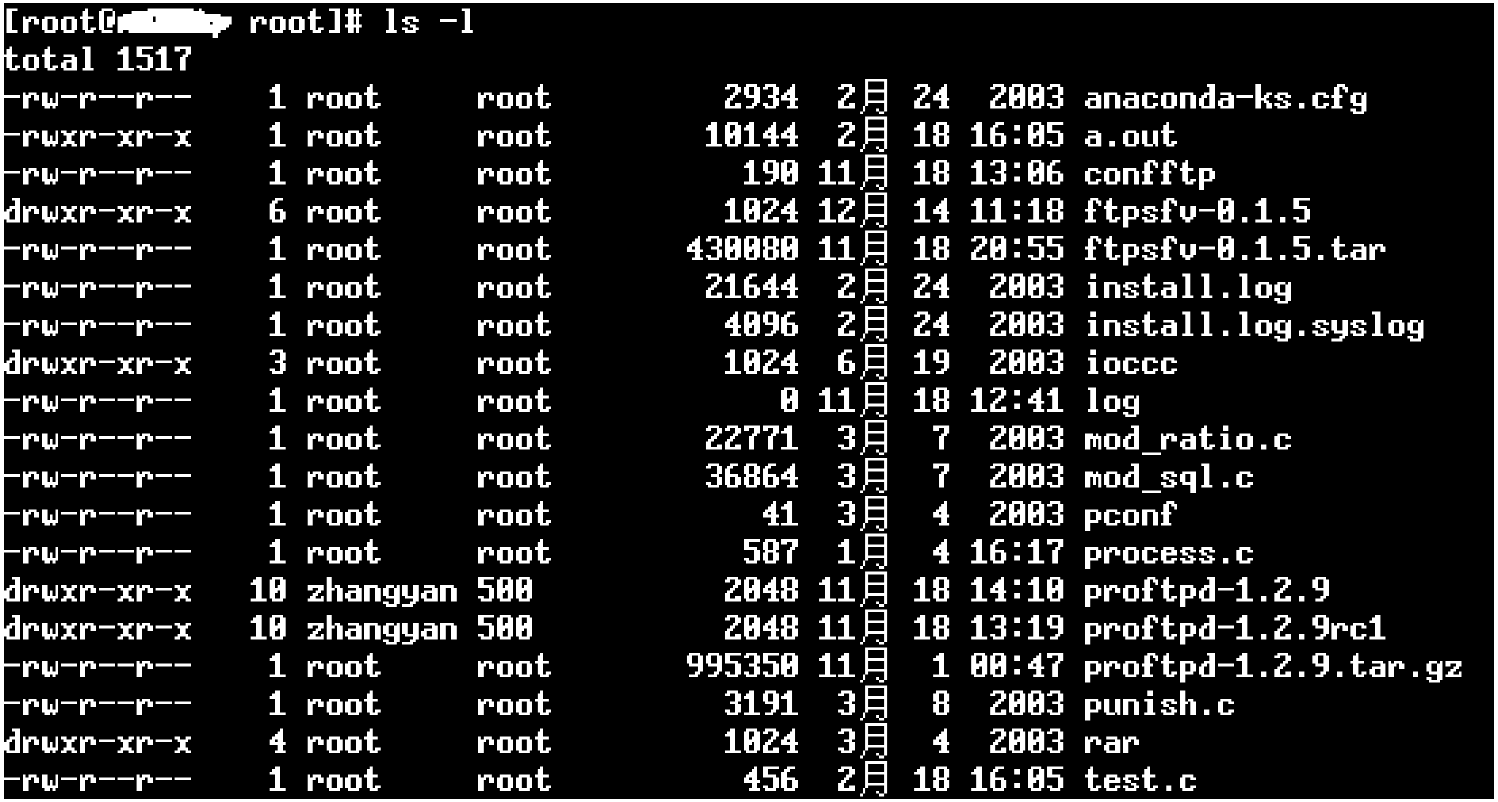
- **Before mounting,**
  - files on floppy are inaccessible
- **After mounting floppy on b,**
  - files on floppy are part of file hierarchy

Pipes (管道)



Two processes connected by a pipe

Security (安全)



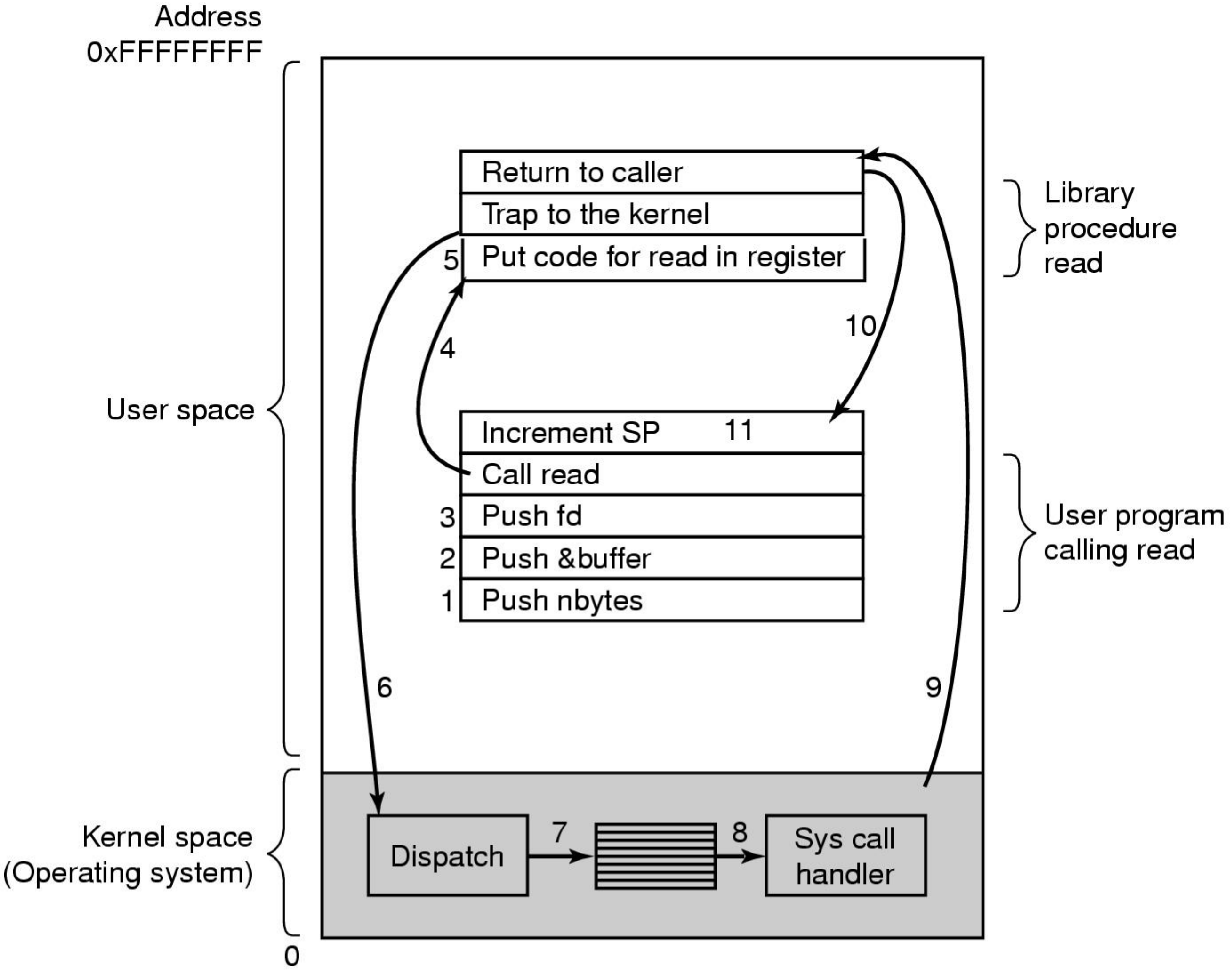
```
[root@[redacted] root]# ls -l
total 1517
-rw-r--r--    1 root     root         2934  2月 24  2003 anaconda-ks.cfg
-rwxr-xr-x    1 root     root        10144  2月 18 16:05 a.out
-rw-r--r--    1 root     root          190 11月 18 13:06 confftp
drwxr-xr-x    6 root     root         1024 12月 14 11:18 ftpsfv-0.1.5
-rw-r--r--    1 root     root       430080 11月 18 20:55 ftpsfv-0.1.5.tar
-rw-r--r--    1 root     root        21644  2月 24  2003 install.log
-rw-r--r--    1 root     root         4096  2月 24  2003 install.log.syslog
drwxr-xr-x    3 root     root         1024  6月 19  2003 ioccc
-rw-r--r--    1 root     root            0 11月 18 12:41 log
-rw-r--r--    1 root     root        22771  3月  7  2003 mod_ratio.c
-rw-r--r--    1 root     root        36864  3月  7  2003 mod_sql.c
-rw-r--r--    1 root     root           41  3月  4  2003 pconf
-rw-r--r--    1 root     root          587  1月  4 16:17 process.c
drwxr-xr-x   10 zhangyan 500          2048 11月 18 14:10 proftpd-1.2.9
drwxr-xr-x   10 zhangyan 500          2048 11月 18 13:19 proftpd-1.2.9rc1
-rw-r--r--    1 root     root       995350 11月  1 00:47 proftpd-1.2.9.tar.gz
-rw-r--r--    1 root     root         3191  3月  8  2003 punish.c
drwxr-xr-x    4 root     root         1024  3月  4  2003 rar
-rw-r--r--    1 root     root          456  2月 18 16:05 test.c
```

- **Read-Write-eXecute**
- **Owner, Group, Others**

The Shell

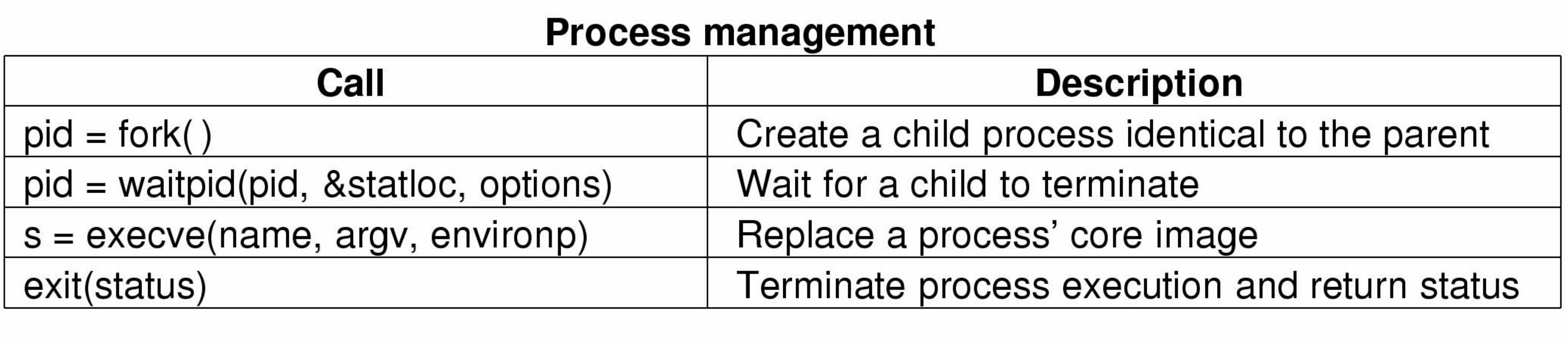
- UNIX command interpreter, called the shell
  - Shell command:
    - date, cat, ls, mv, sort……
- Explorer is the shell in Windows
  - cmd.exe or command.com is also shell

Steps in Making a System Call (系统调用)



read (fd, buffer, nbytes)

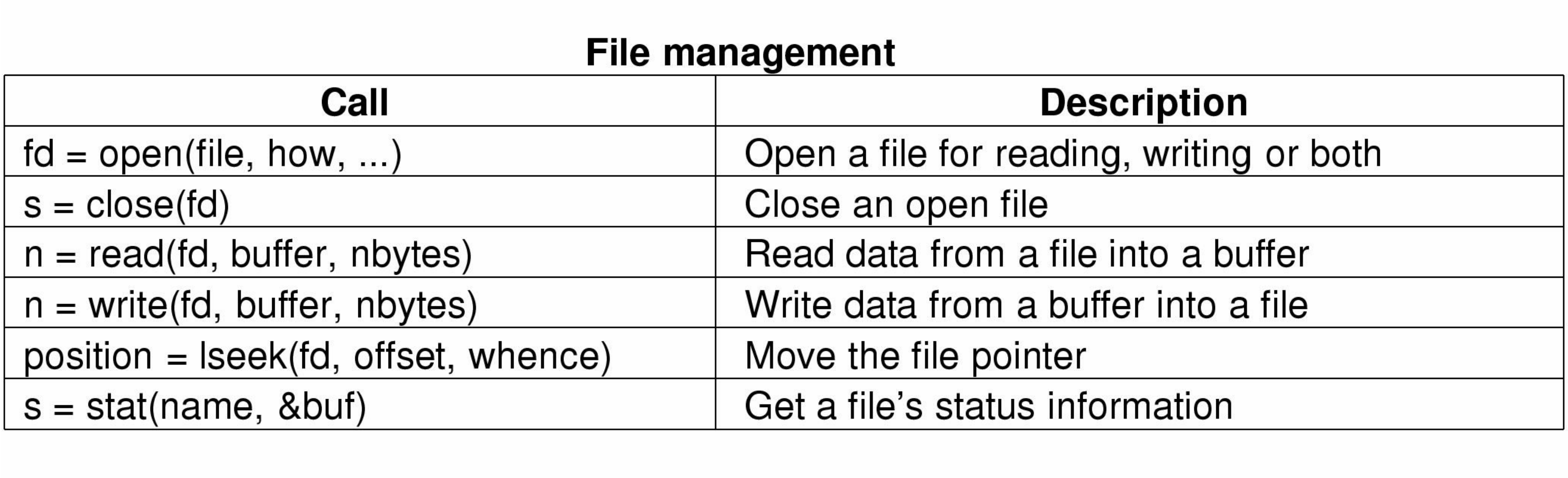
Some System Calls For Process Management



Process management

Call	Description
pid = fork( )	Create a child process identical to the parent
pid = waitpid(pid, &statloc, options)	Wait for a child to terminate
s = execve(name, argv, environp)	Replace a process’ core image
exit(status)	Terminate process execution and return status

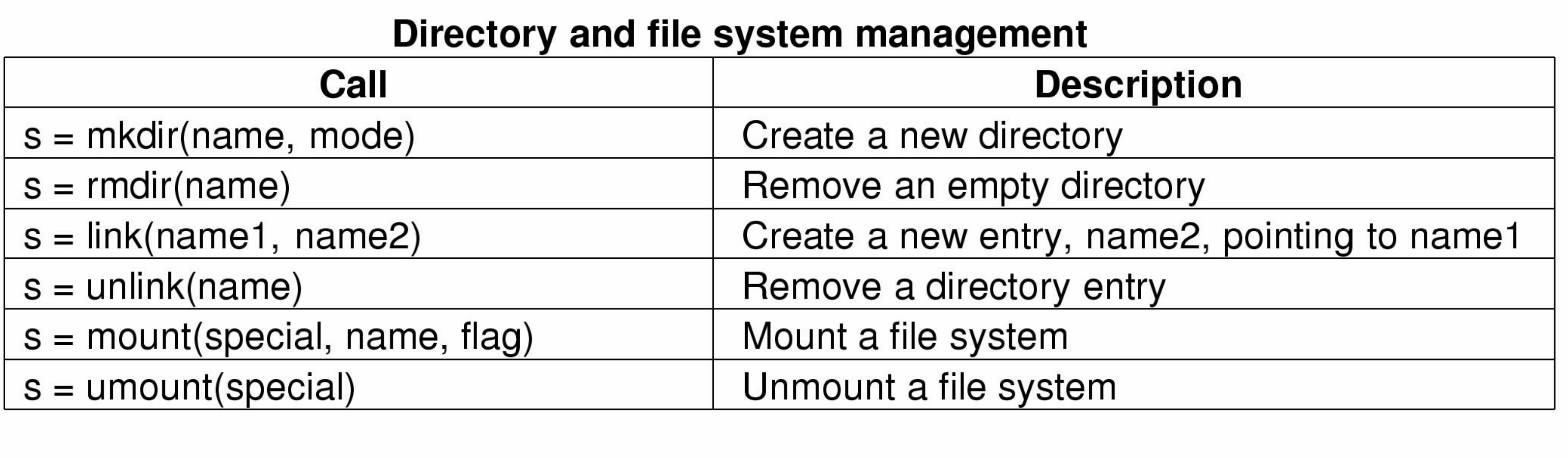
Some System Calls For File Management



File management

Call	Description
fd = open(file, how, ...)	Open a file for reading, writing or both
s = close(fd)	Close an open file
n = read(fd, buffer, nbytes)	Read data from a file into a buffer
n = write(fd, buffer, nbytes)	Write data from a buffer into a file
position = lseek(fd, offset, whence)	Move the file pointer
s = stat(name, &buf)	Get a file's status information

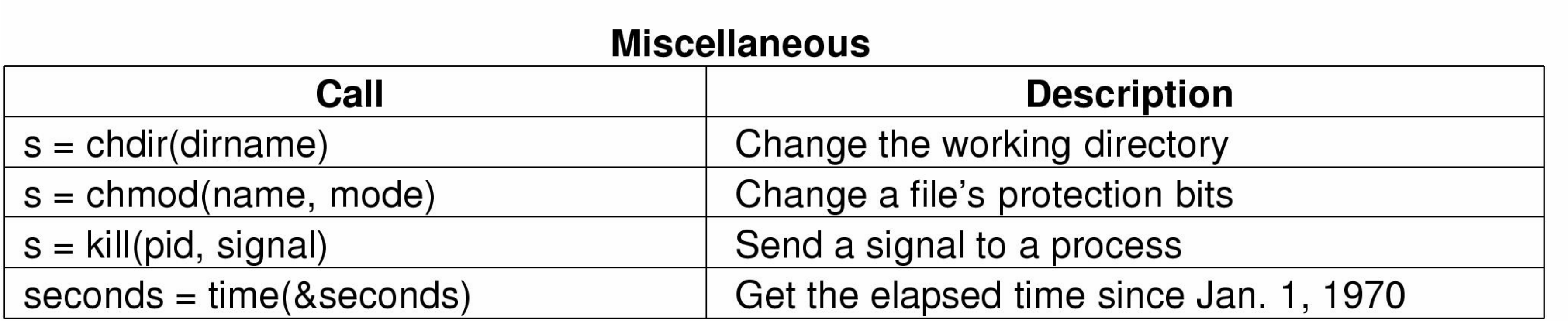
Some System Calls For Directory Management



Directory and file system management

Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system

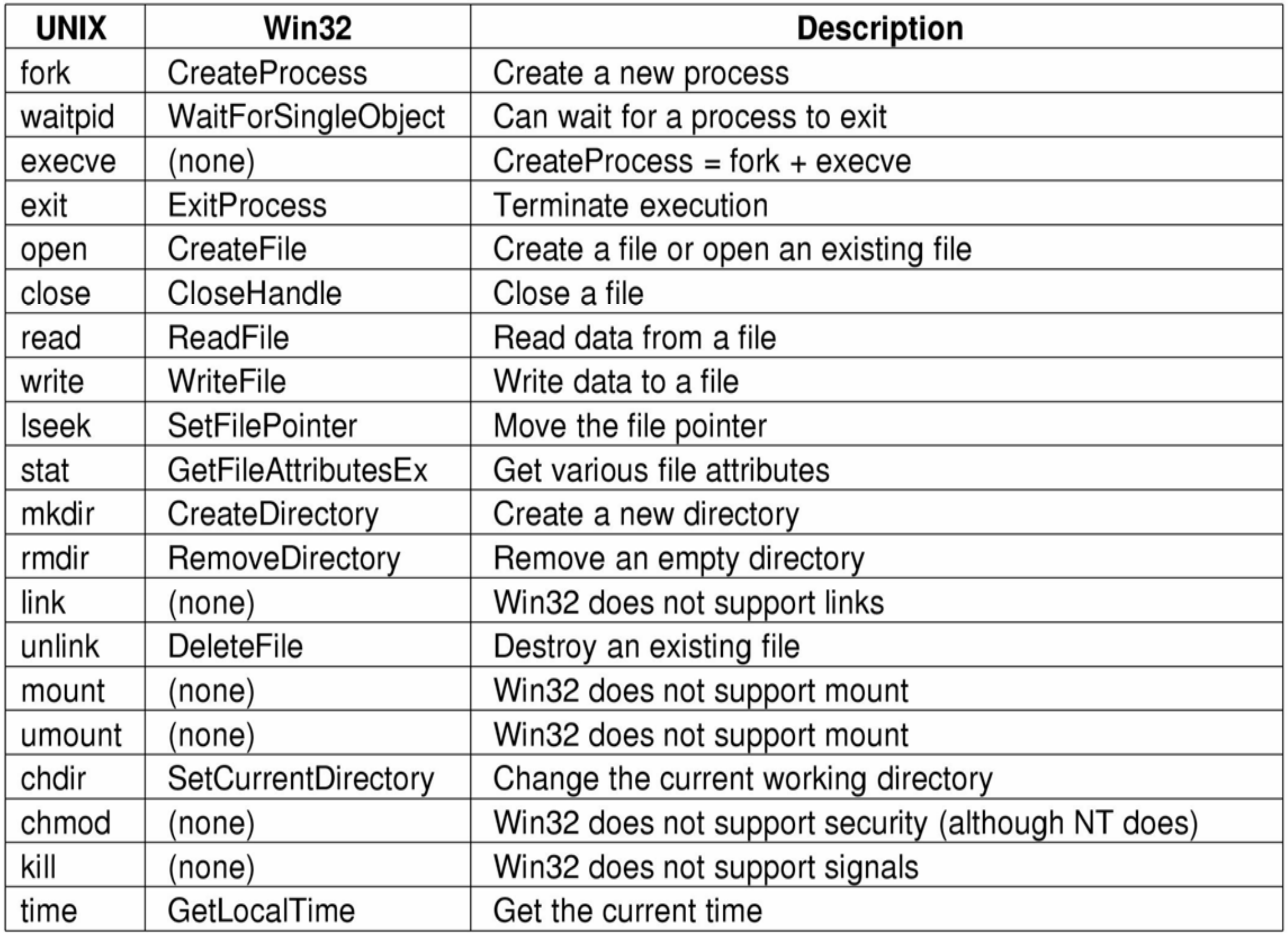
Some System Calls For Miscellaneous Tasks



Miscellaneous

Call	Description
s = chdir(dirname)	Change the working directory
s = chmod(name, mode)	Change a file’s protection bits
s = kill(pid, signal)	Send a signal to a process
seconds = time(&seconds)	Get the elapsed time since Jan. 1, 1970

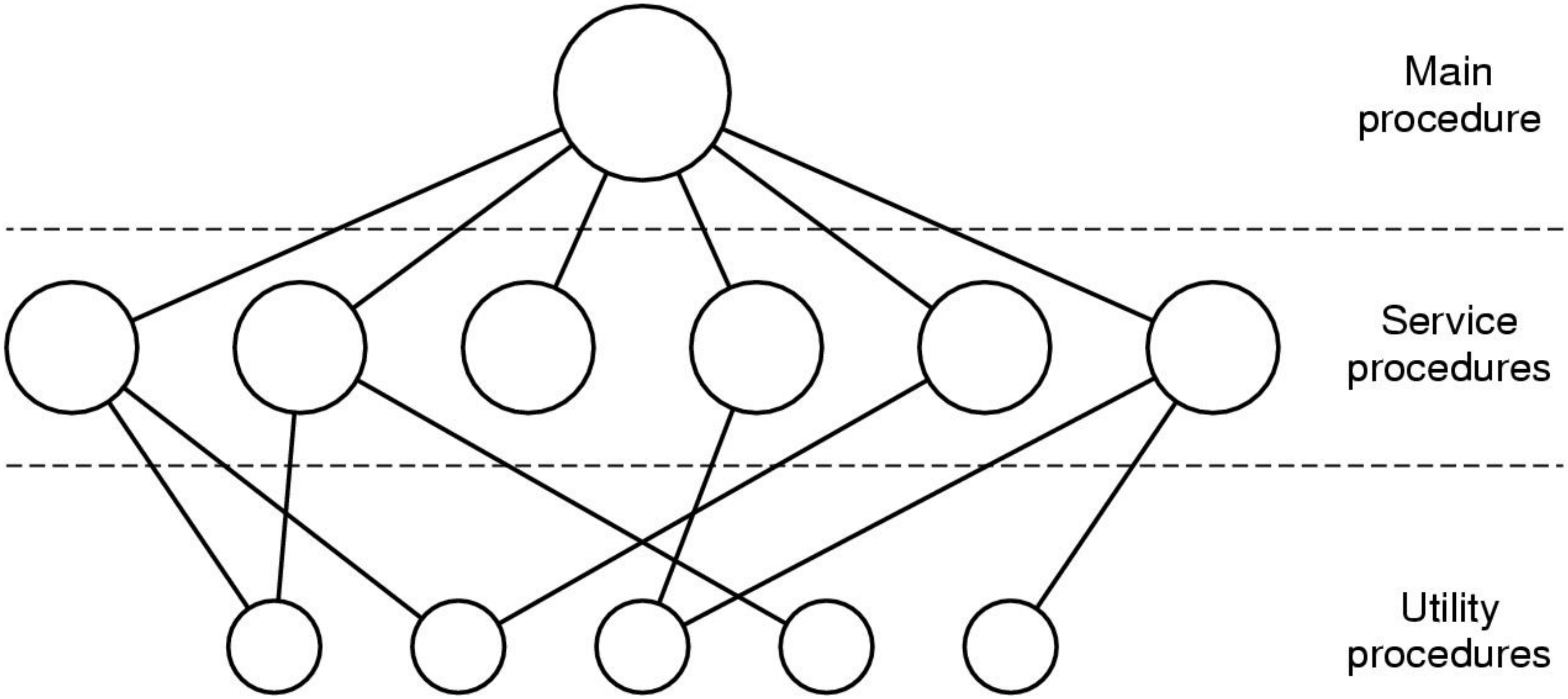
System Calls



UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time

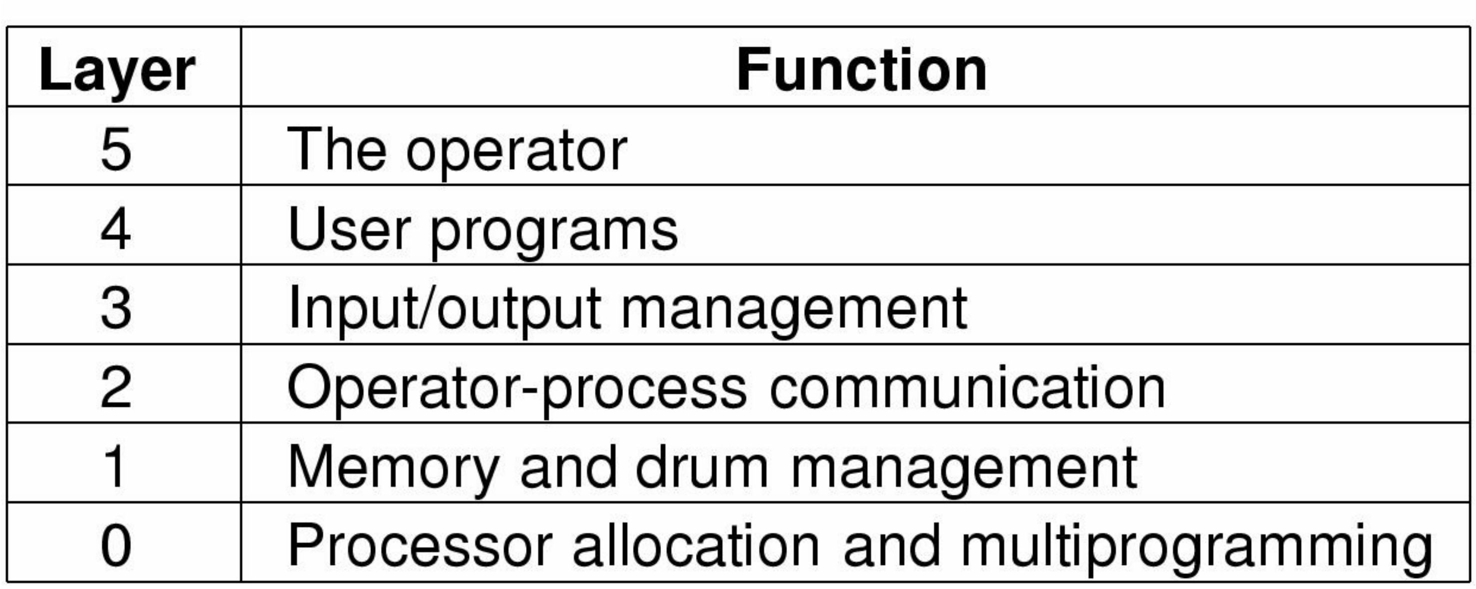
Some Win32 API calls

Operating System Structure



Simple structuring model for a monolithic system

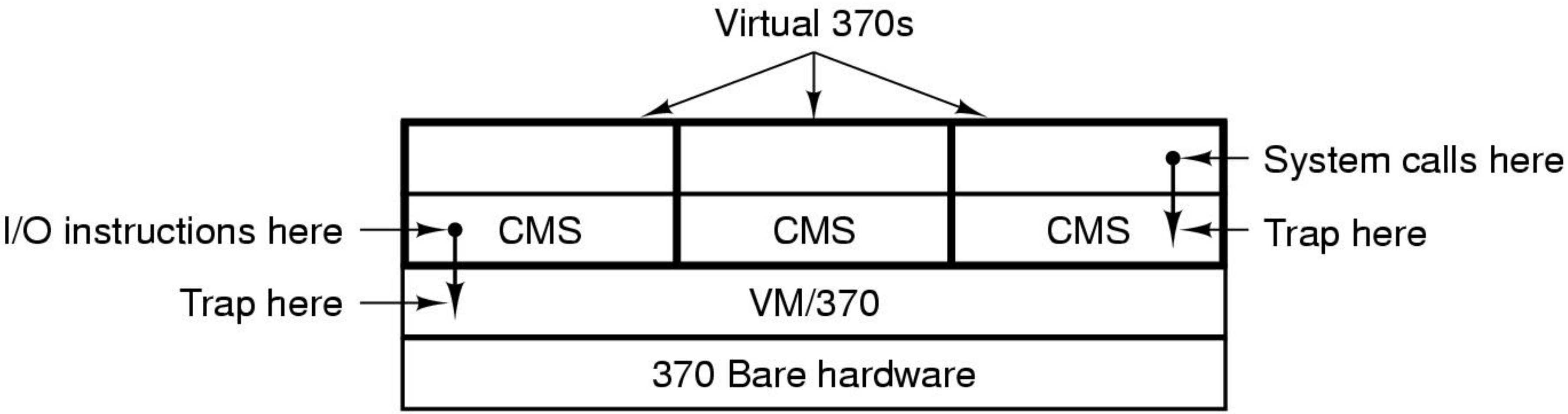
Layered Systems



Layer	Function
5	The operator
4	User programs
3	Input/output management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

Structure of the THE operating system

Virtual Machines



Structure of VM/370 with CMS