《操作系统》

概论

主讲教师： 孙卫真

基　本　信　息

1、学习方法和要求

6W

- What is an operating system?
- What will be learning in this course?
- Why must we study it?
- hoW will we learn it?
- Why do we need operating system?
- hoW does the operating system working?

基　本　信　息

2、学习内容和组织

CONTENTS

HOW CAN I LEARN OS WELL?

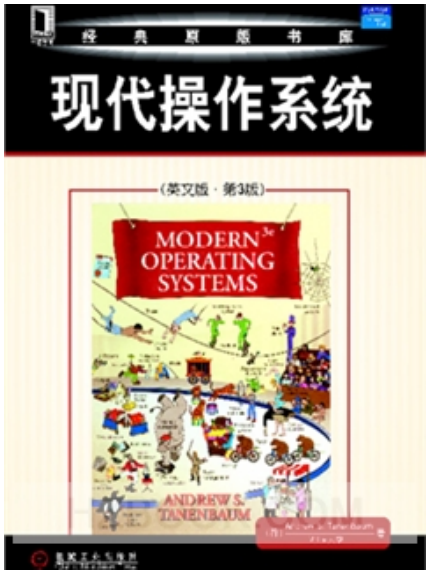
- Take class on time
- Reading reference books after class
- Complete homework/exercise by yourself
- Question

基　本　信　息

3、教材和参考书

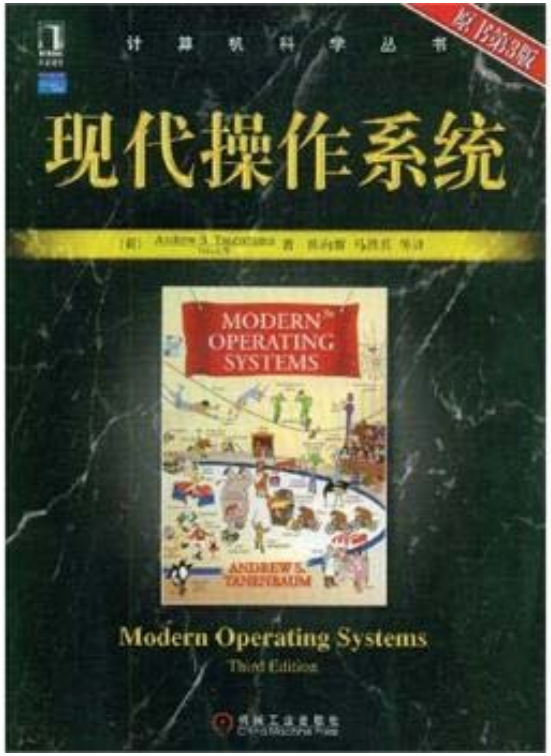
TEACHING MATERIAL

- Andrew S. Tanenbaum
- Modern Operating System 3nd edition



REFERENCE BOOKS

- Modern Operating System 3nd edition
- 中译本
- 陈向群译

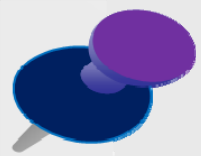


基　本　信　息

4、成绩和学分

ACHIEVEMENT

- Attendance, Performance 20%
- Homework, Exercise 20%
- Examination 60%



一、操作系统概论

1、操作系统的定义

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

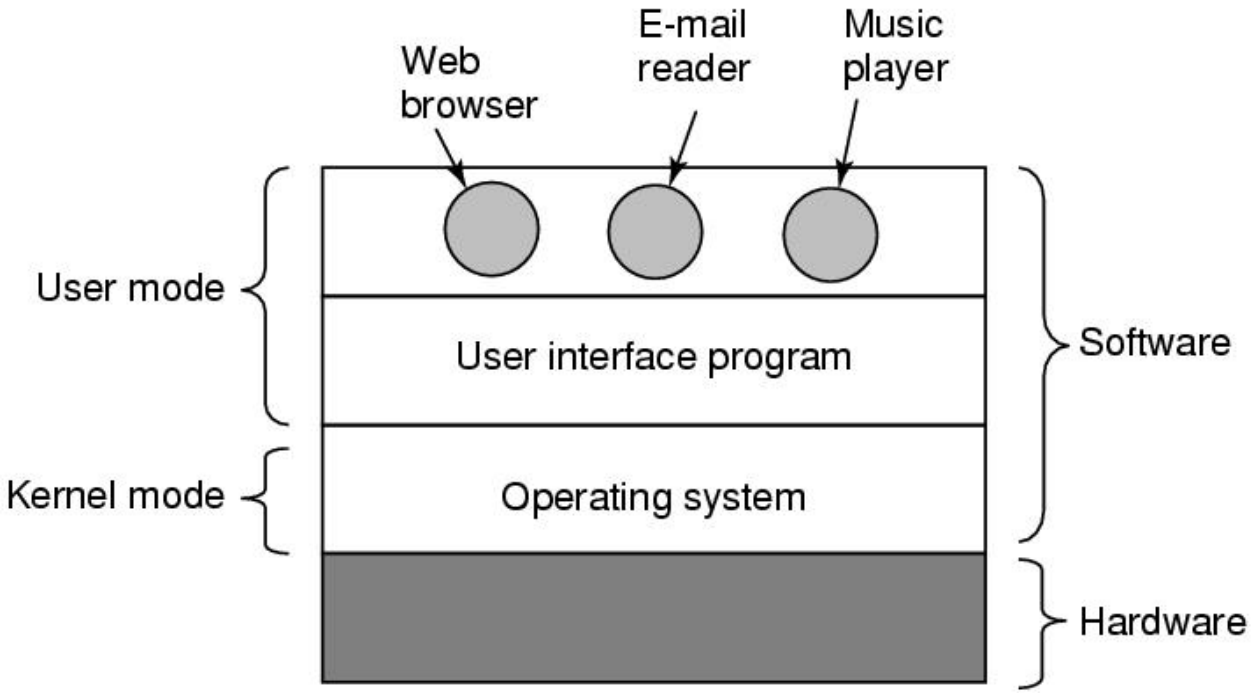
What Is An Operating System (1)

A modern computer consists of:

- One or more processors
- Main memory
- Disks
- Printers
- Various input/output devices

Managing all these components requires a layer of software – the **operating system**

What Is An Operating System (2)



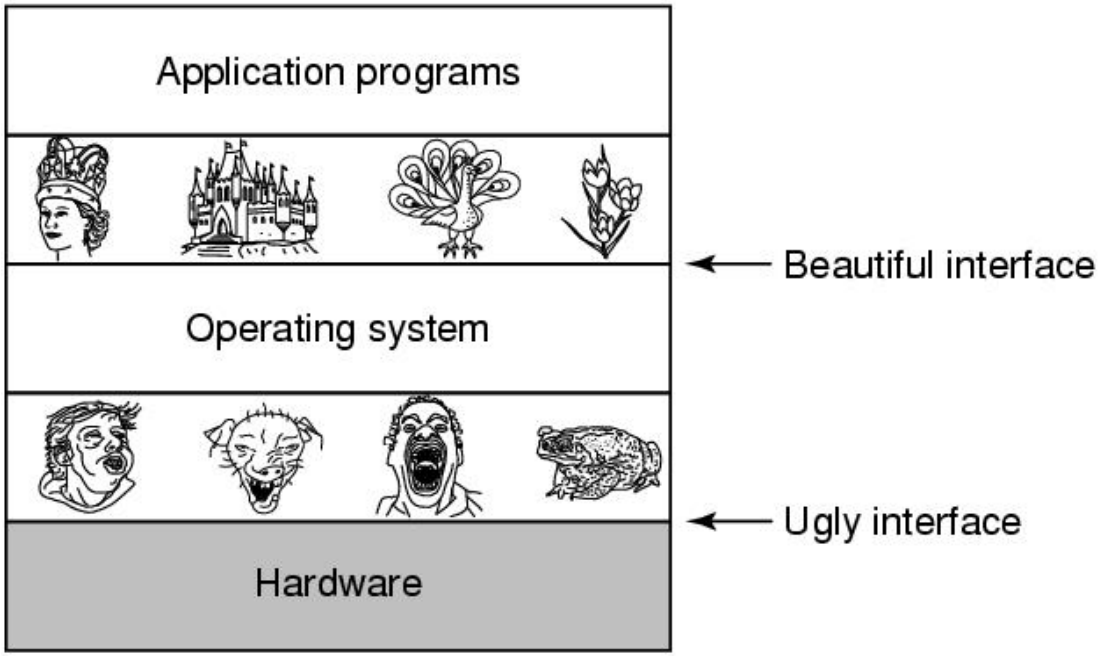
- Where the operating system fits in.

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
  - The Operating System as a User Interface
    - Command input (Line,GUI,Script,NUI)
    - System call (Lib,Function,DLL)

- It is a resource manager
  - Each program gets time with the resource
  - Each program gets space on the resource

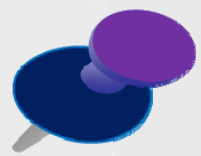
The Operating System as an Extended Machine



- Operating systems turn ugly hardware into beautiful abstractions.

The Operating System as a Resource Manager

- Allow multiple programs to run at the same time
- Manage and protect memory, I/O devices, and other resources
- Includes sharing resources in two different ways:
  - In time
  - In space



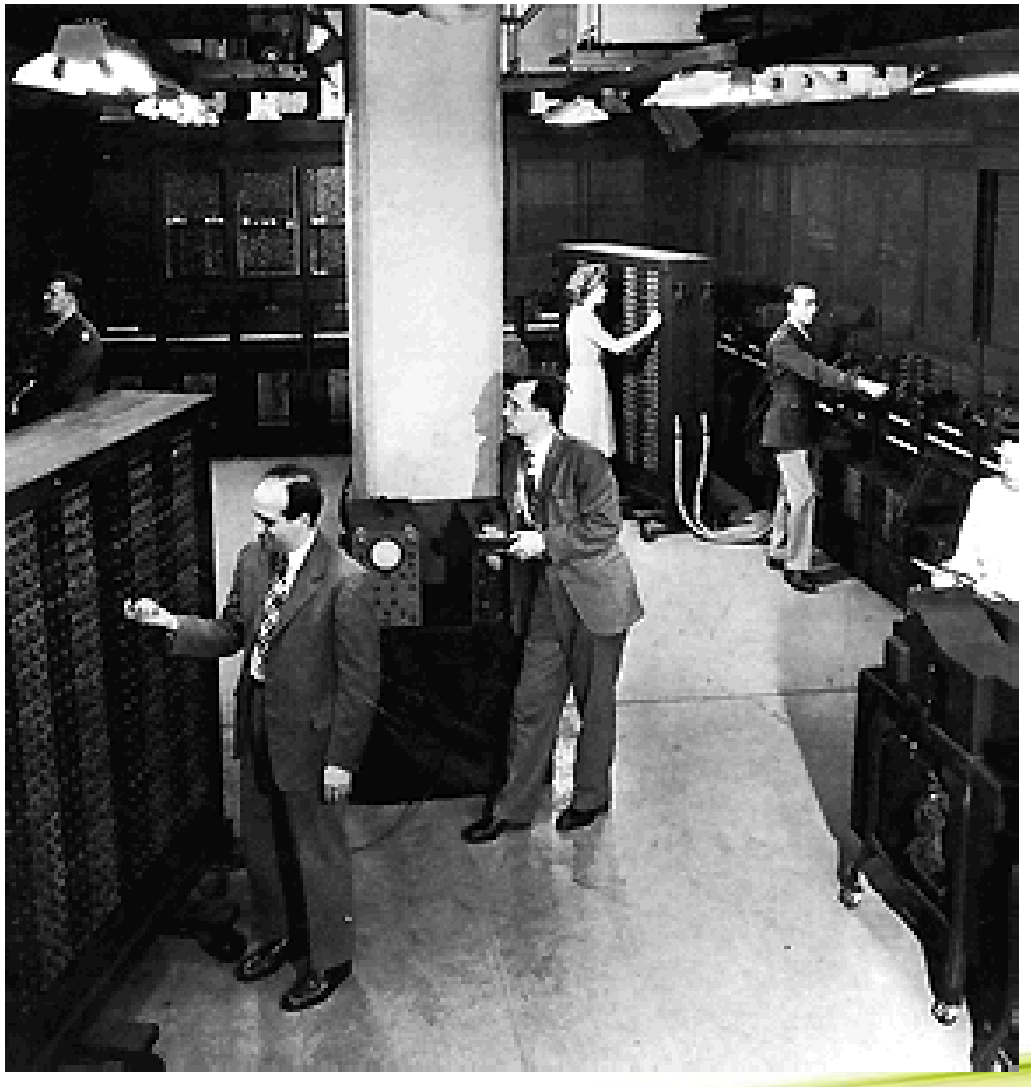
一、操作系统概论

2、操作系统的历史发展

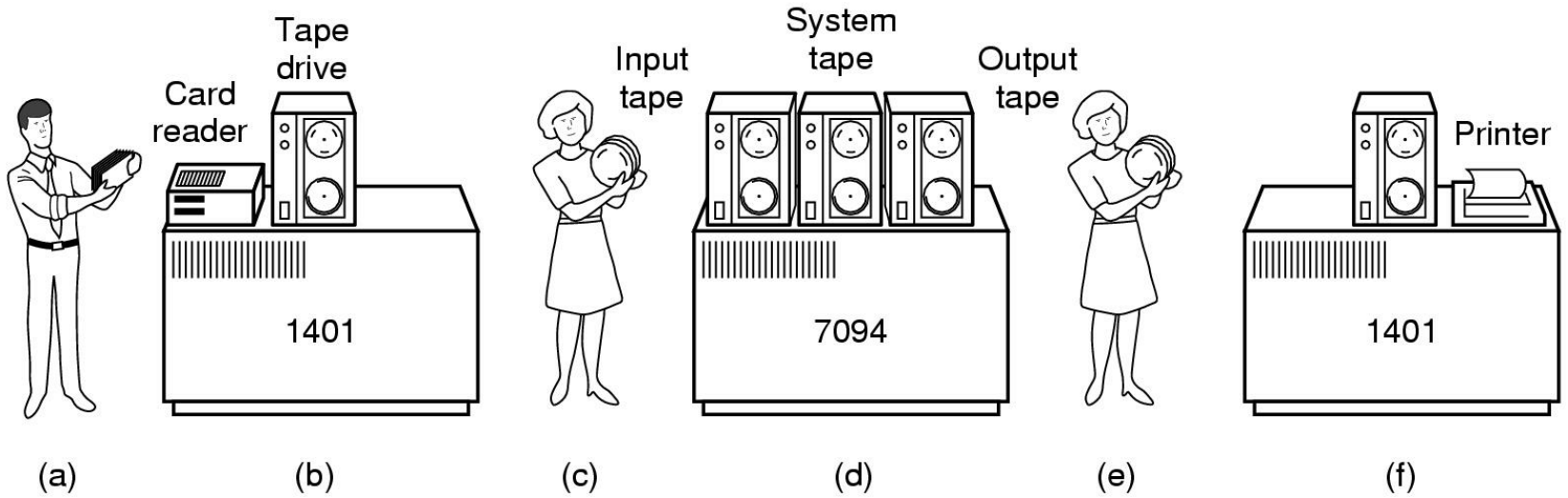
History of Operating Systems

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

History of Operating Systems (1)



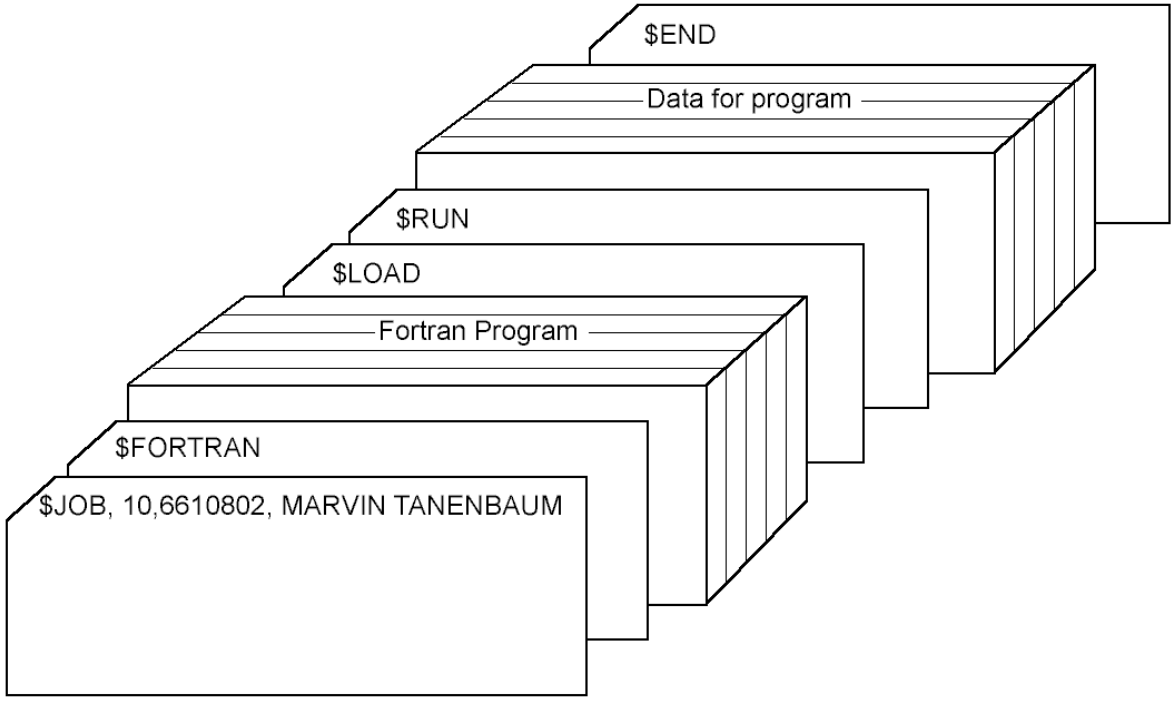
History of Operating Systems (2)



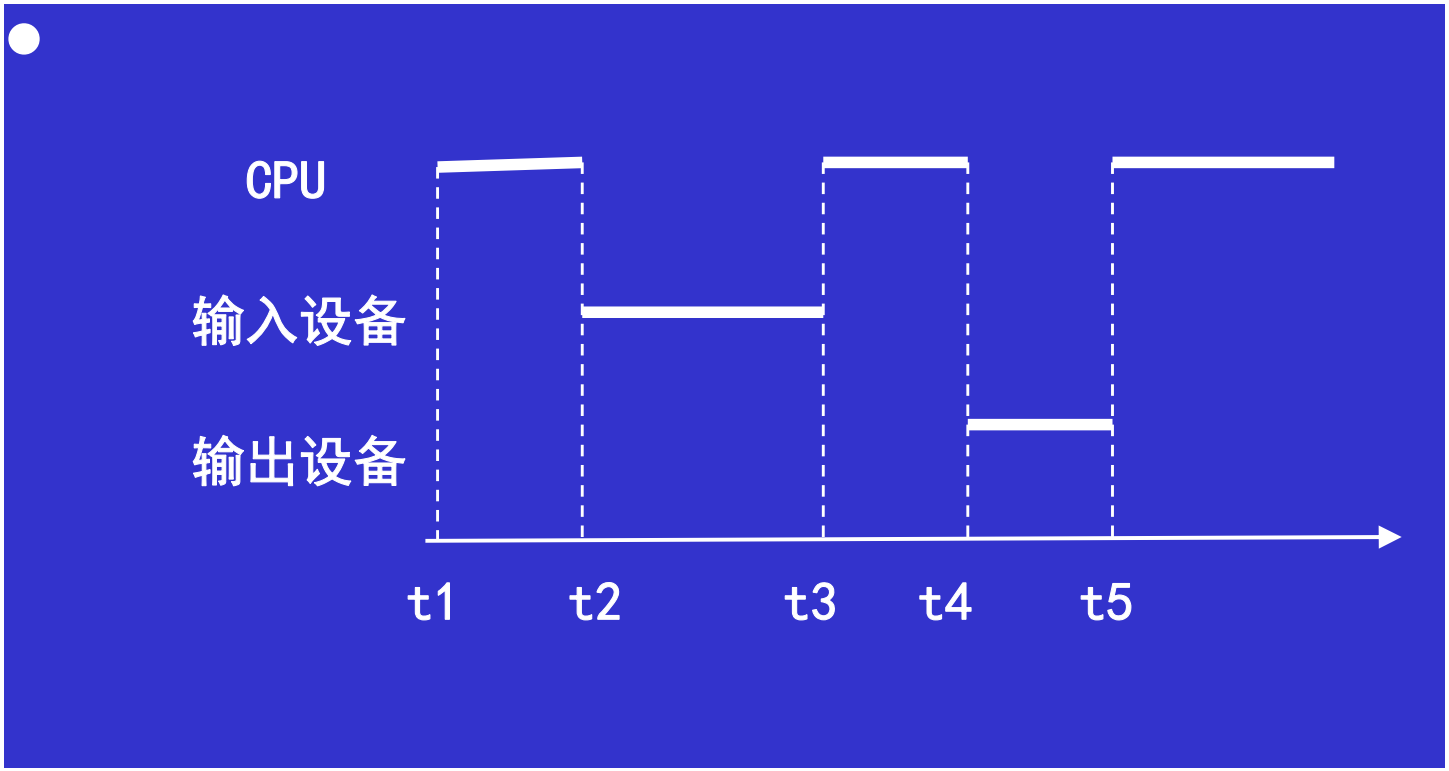
Early batch system

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

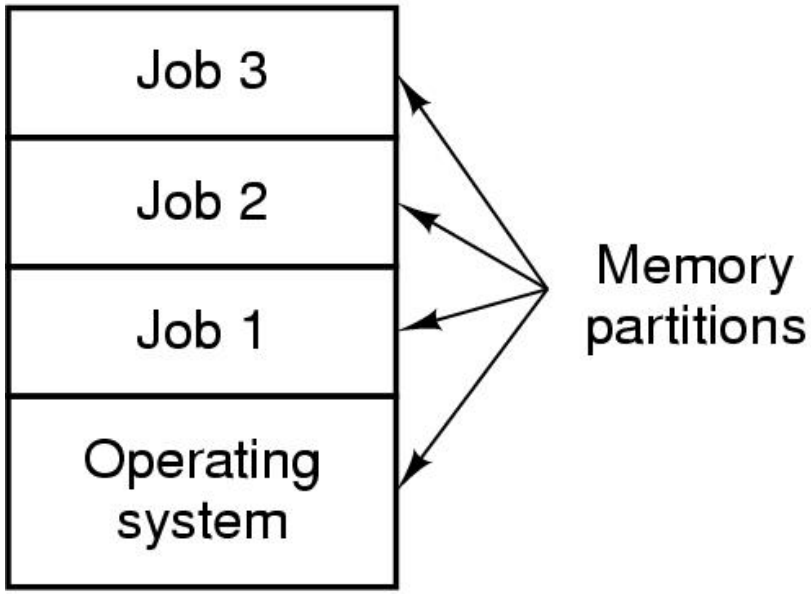
History of Operating Systems (2)



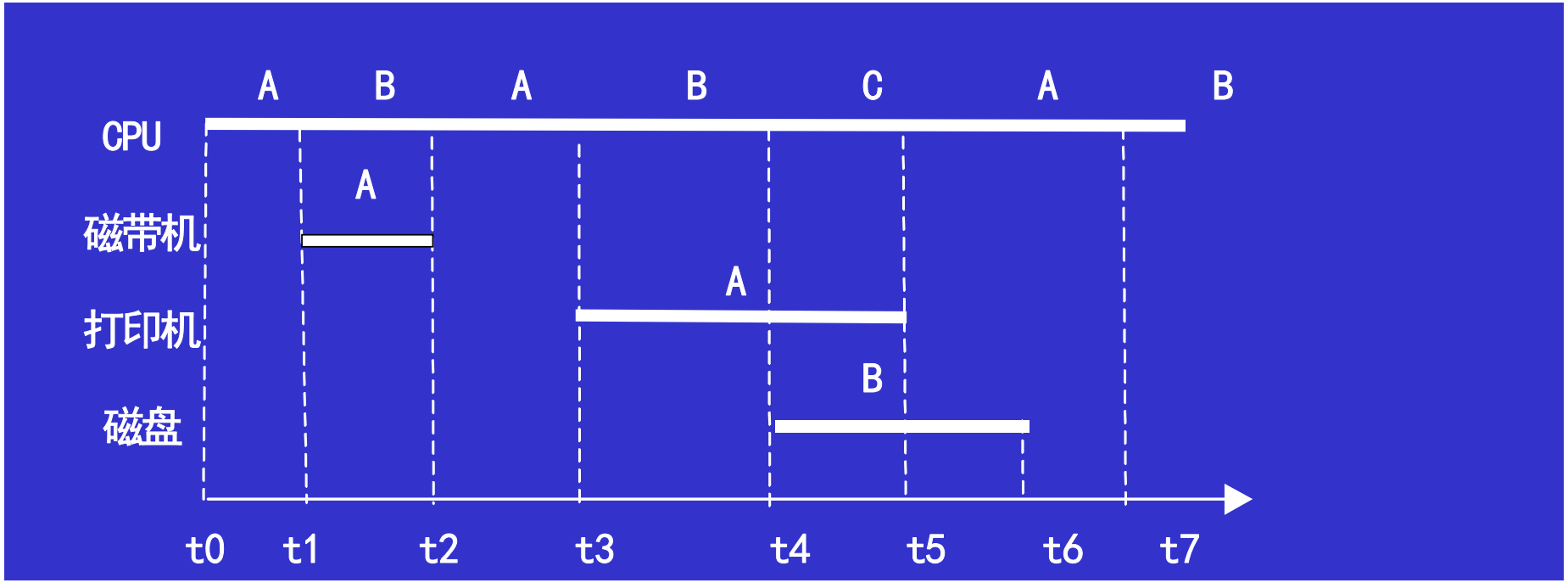
- Structure of a typical FMS job – 2nd generation


CPU
输入设备
输出设备
t1
t2
t3
t4
t5

History of Operating Systems (3)

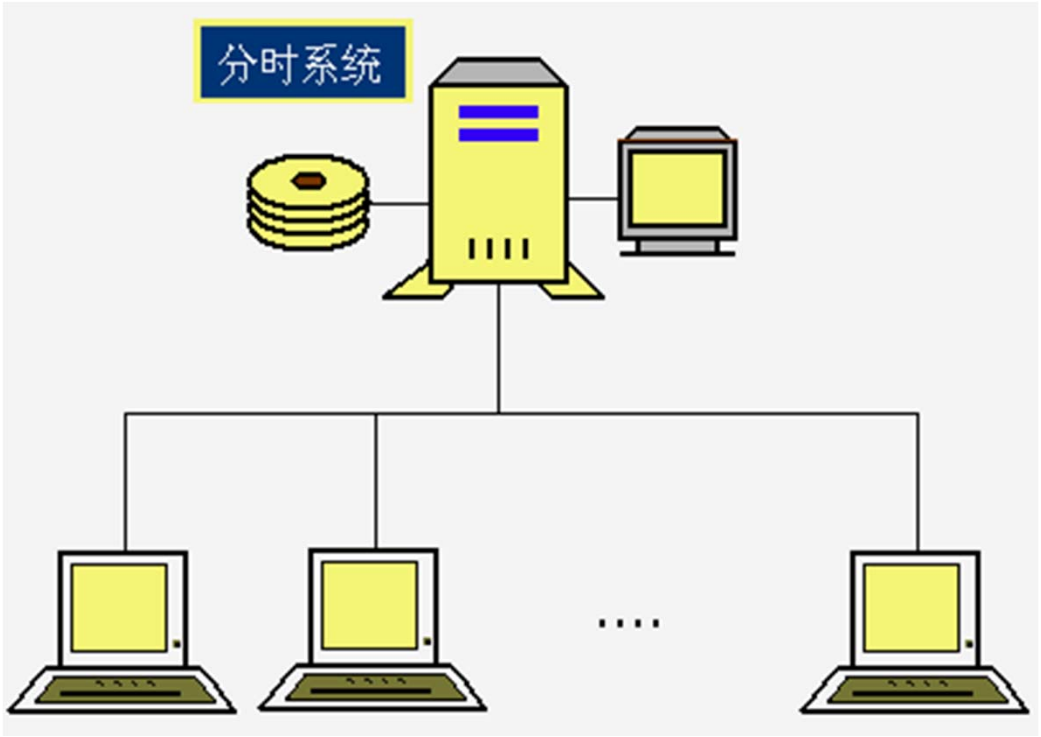


- Multiprogramming system
  - three jobs in memory – 3rd generation


A
B
A
B
C
A
B
CPU
A
磁带机
A
打印机
B
磁盘
t0
t1
t2
t3
t4
t5
t6
t7

History of Operating Systems (3)

- Spooling (Simultaneous Peripheral Operation On Line)
- Timesharing System
  - CTSS
- UNIX

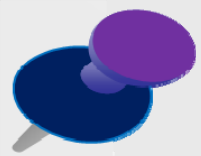

分时系统

History of Operating Systems (4)

- MS-DOS
- GUI
  - MAC OS;Windows; X Windows
- Network Operating System

  Distributed Operating System

  Embedded Operating System



一、操作系统概论

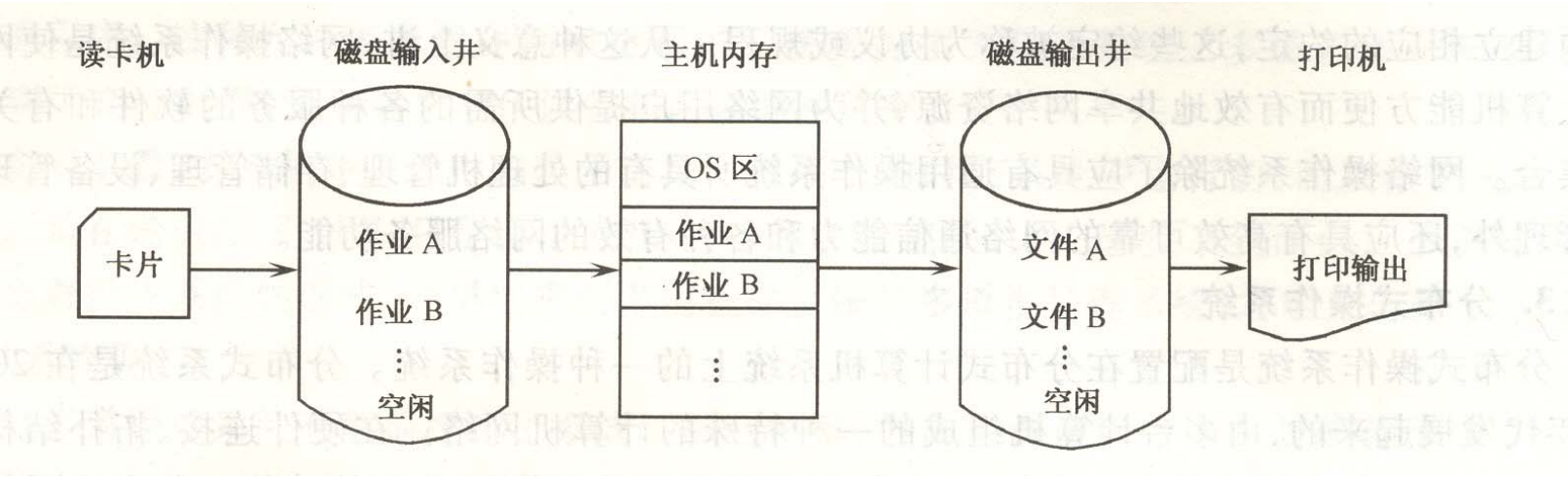
3、操作系统的分类

The Operating System Zoo

- Mainframe Operating Systems(OS/390; UNIX)
- Server Operating Systems(LINUX;WIN200x)
- Multiprocessor Operating Systems
- Personal Computer Operating Systems
- Handheld Computer Operating Systems(PDA)
- Embedded Operating Systems
- Sensor Node Operating Systems
- Real-Time Operating Systems
- Smart Card Operating Systems

The Operating System Zoo

- batch
- timesharing
- Real-Time


读卡机
磁盘输入井
主机内存
磁盘输出井
打印机
卡片
作业 A
作业 B
空闲
OS 区
作业 A
作业 B
文件 A
文件 B
空闲
打印输出

多道批处理系统

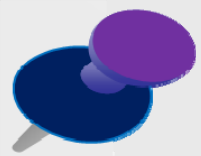
- 优点：
  - 资源利用率高：CPU和内存利用率较高
  - 作业吞吐量大：单位时间内完成的工作总量大
- 缺点：
  - 用户交互性差：整个作业完成后或中间出错时，才与用户交互，不利于调试和修改
  - 作业平均周转时间长：短作业的周转时间显著增长

分时系统

- 人机交互性好：在调试和运行程序时由用户自己操作
- 共享主机：多个用户同时使用
- 用户独立性：对每个用户而言好象独占主机

实时系统

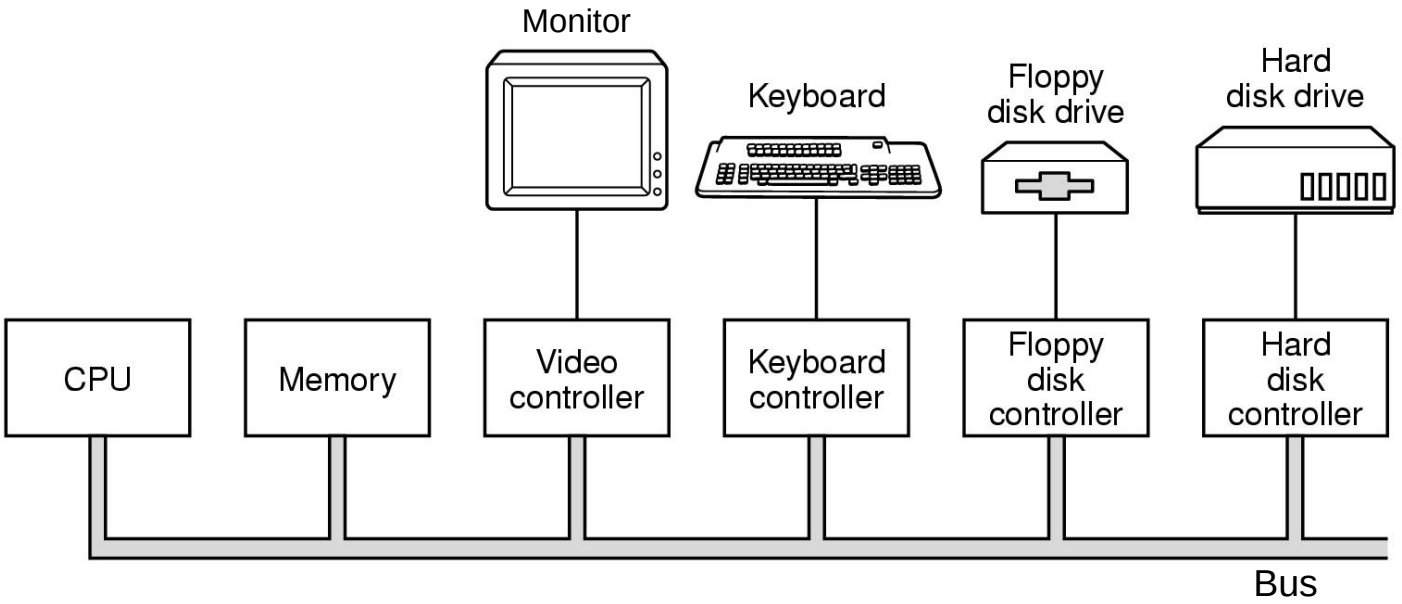
- 多用于工业过程控制、军事实时控制、金融等领域，包括实时控制、实时信息处理，其主要特征是实时性和可靠性。



一、操作系统概论

4、操作系统的功能

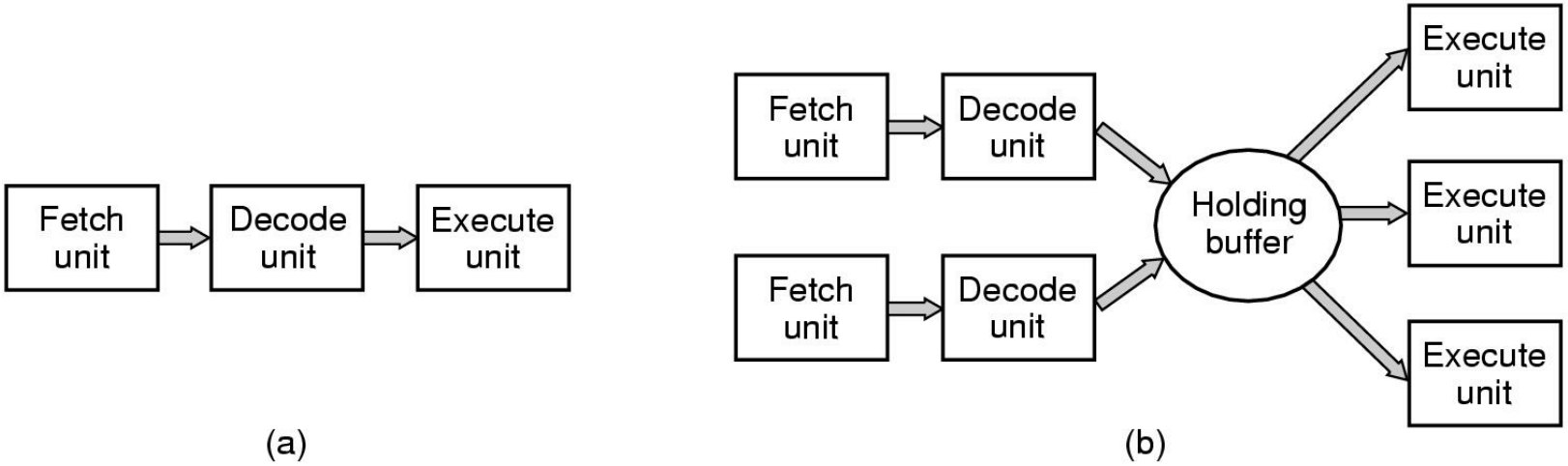
Computer Hardware Review (1)



- Components of a simple personal computer

  Processors； Memory； I/O Devices； Buses

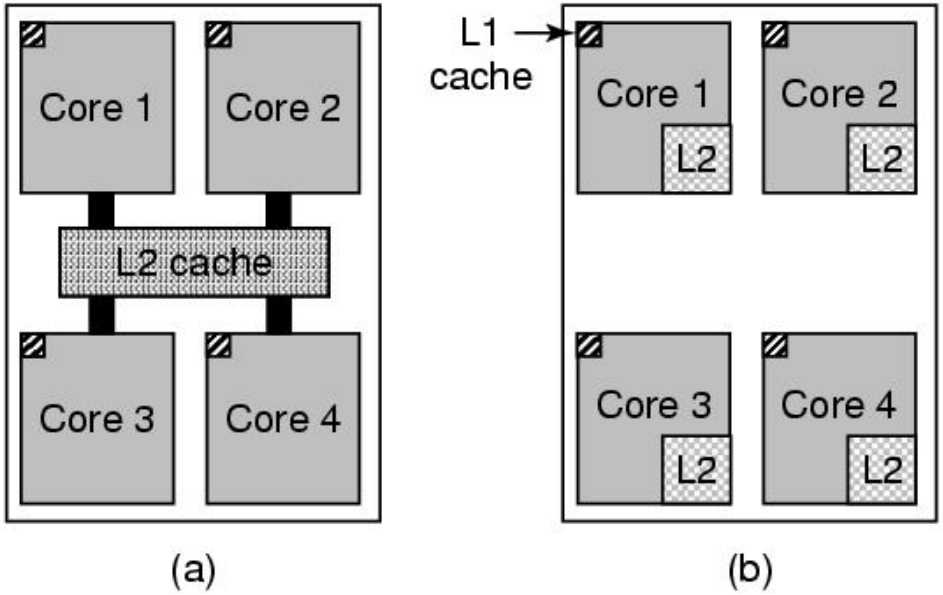
Computer Hardware Review (2)



(a) A three-stage pipeline

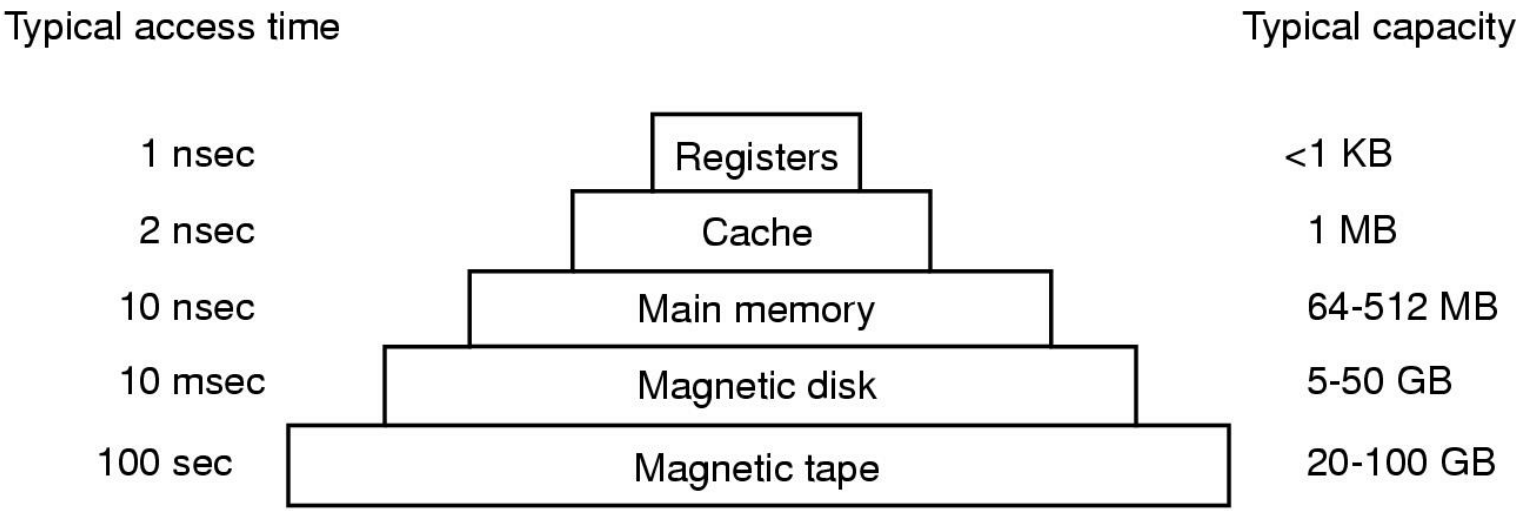
(b) A superscalar CPU

Multithreaded and Multicore Chips



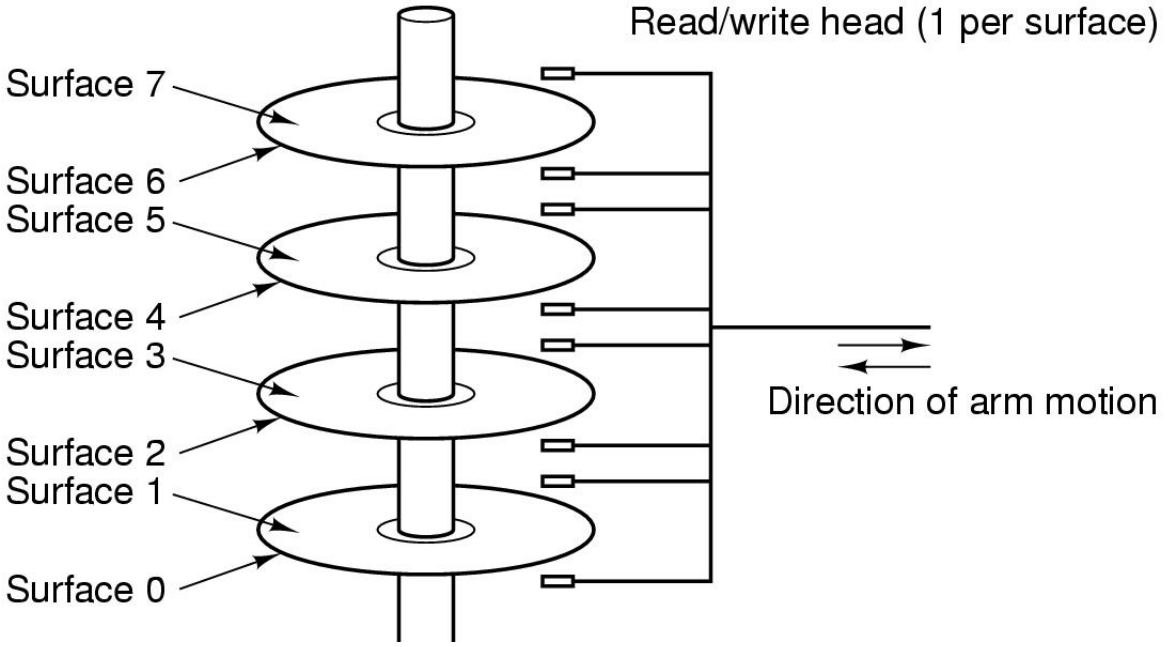
- (a) A quad-core chip with a shared L2 cache.
  (b) A quad-core chip with separate L2 caches.

Computer Hardware Review (3)



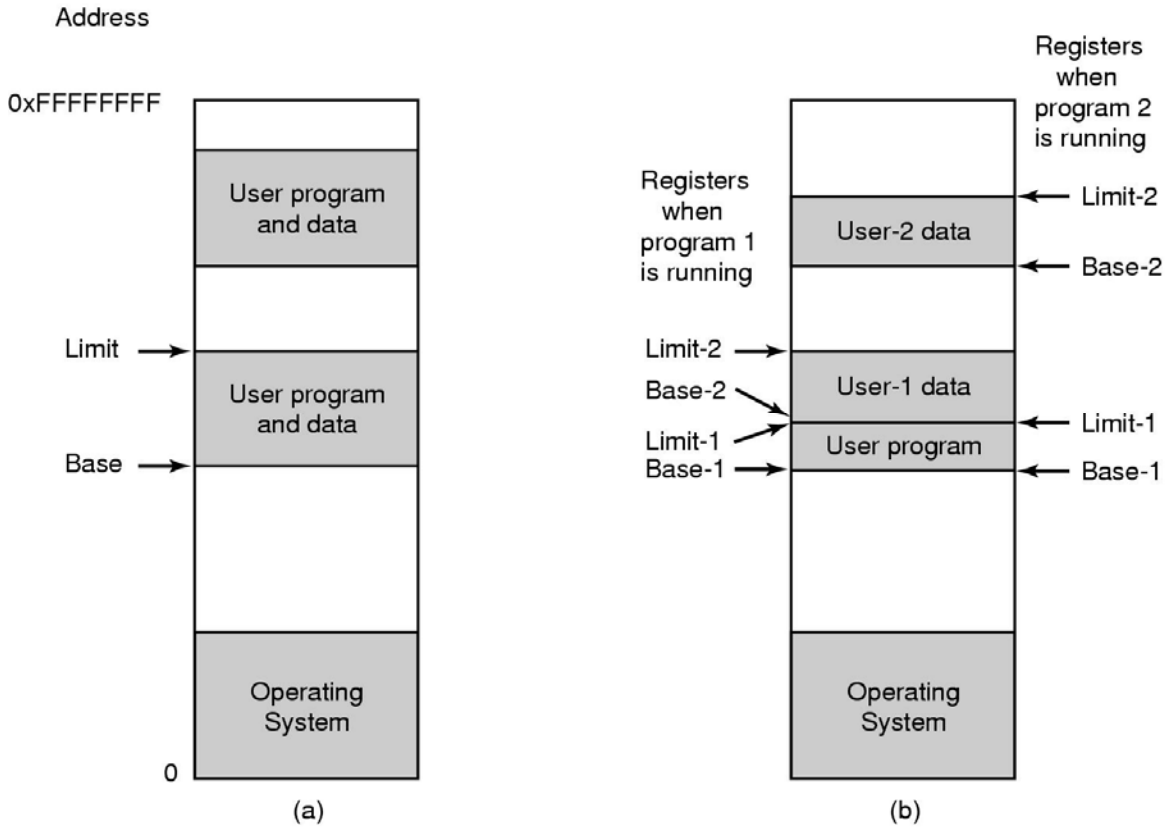
- Typical memory hierarchy
  - numbers shown are rough approximations

Computer Hardware Review (4)



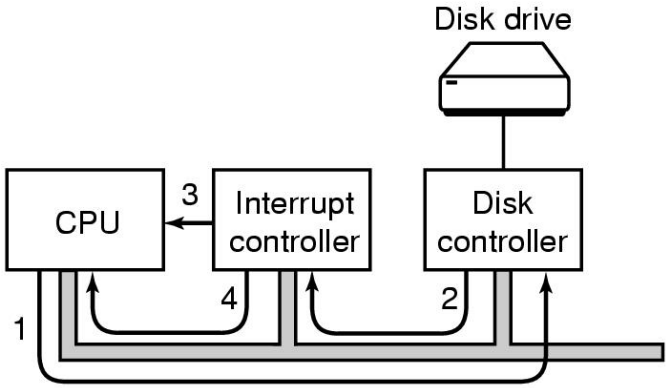
Structure of a disk drive

Computer Hardware Review (5)

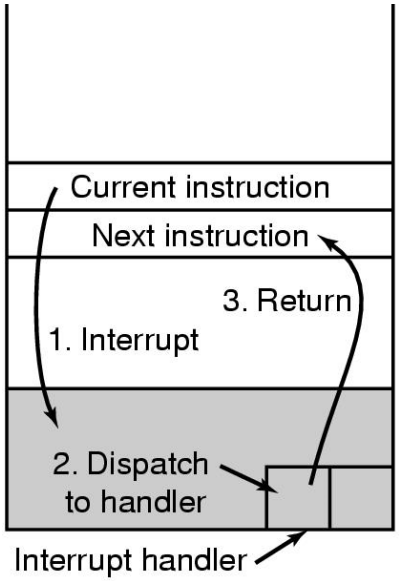


One base-limit pair and two base-limit pairs

Computer Hardware Review (6)



(a)

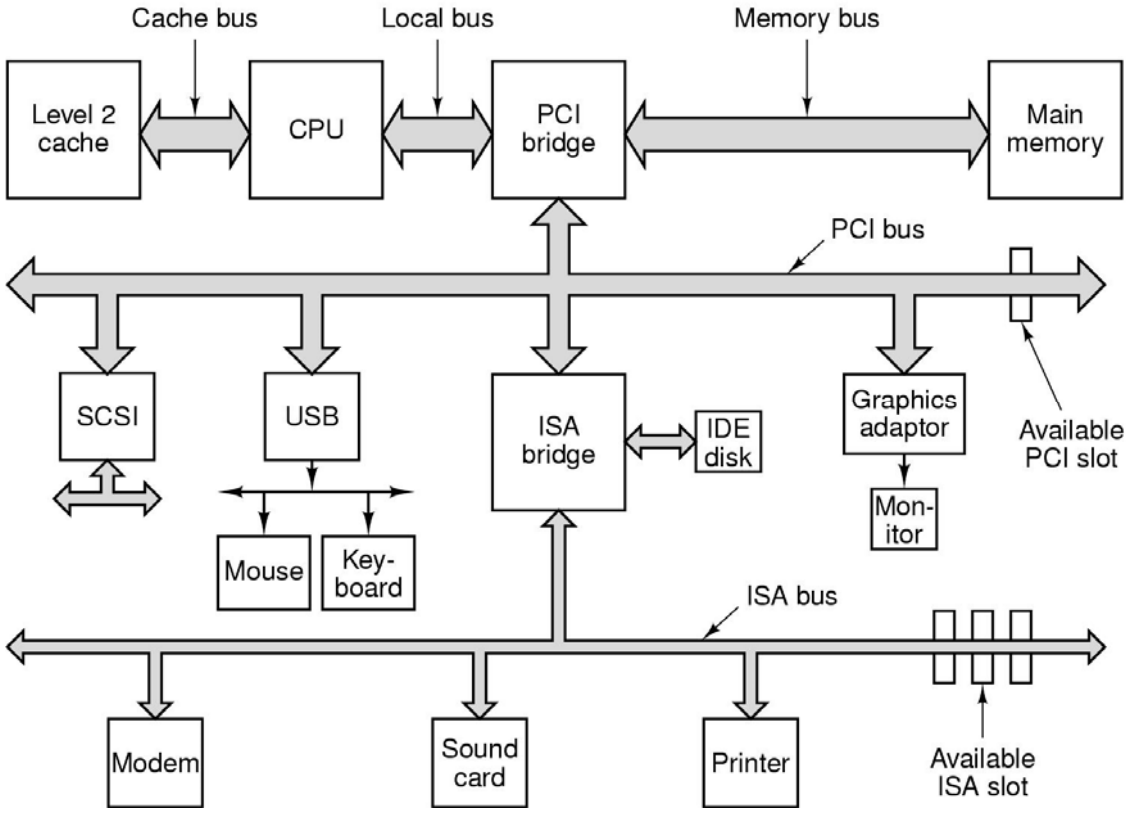


(b)

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

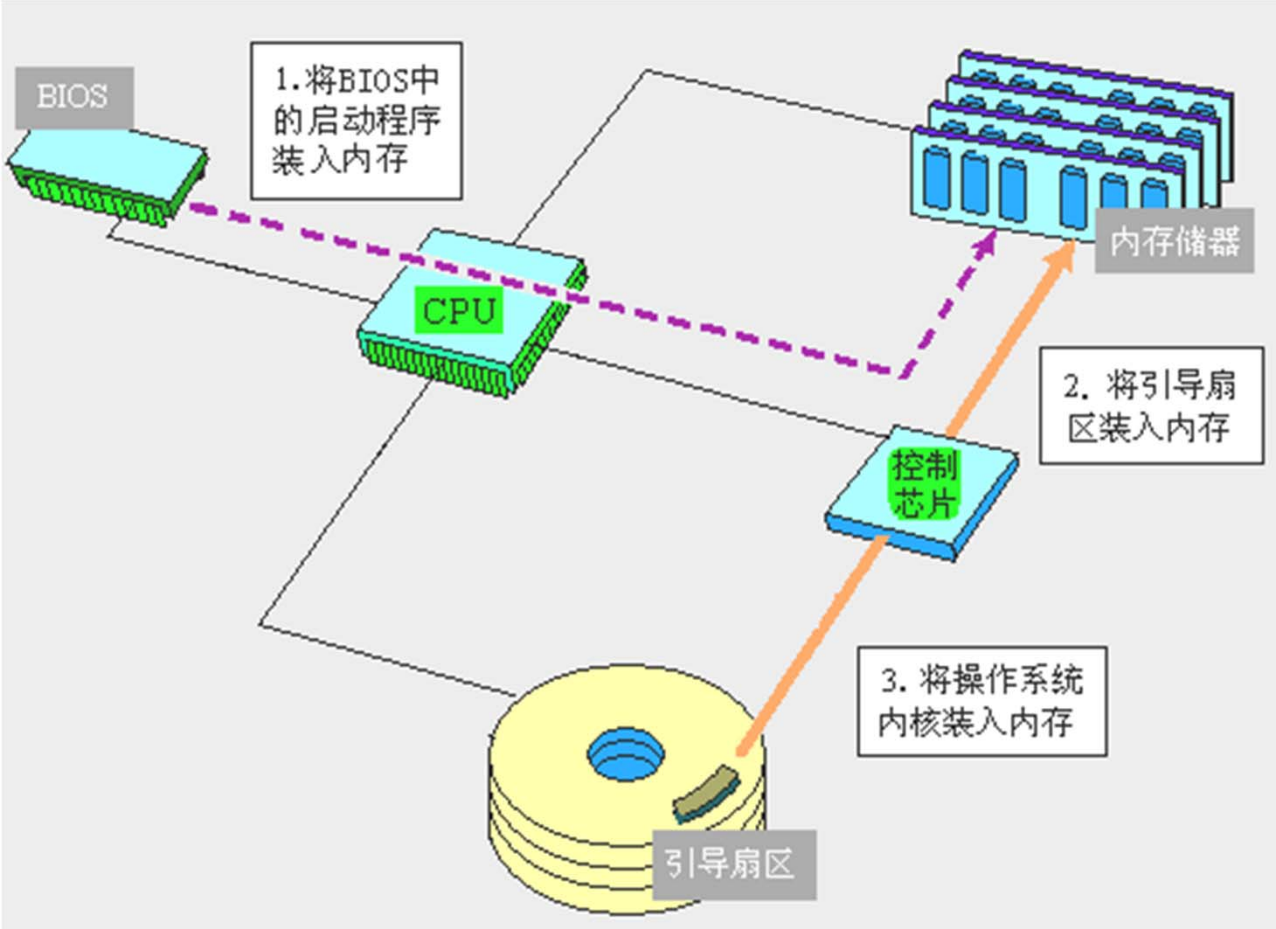
(b) How the CPU is interrupted

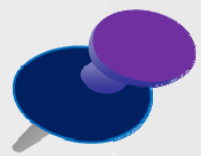
Computer Hardware Review (7)



Structure of a large Pentium system

Bootstrap :


BIOS
1.将BIOS中的启动程序装入内存
CPU
内存储器
2. 将引导扇区装入内存
控制芯片
3. 将操作系统内核装入内存
引导扇区



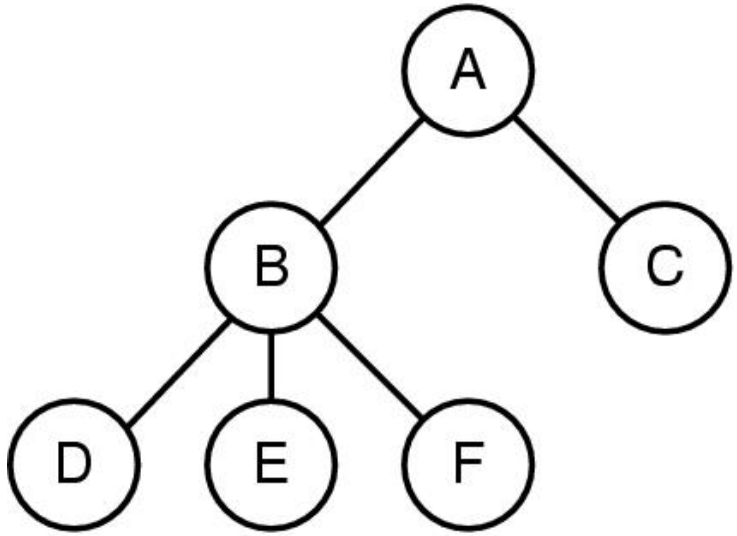
一、操作系统概论

5、操作系统的概念

OPERATING SYSTEM CONCEPTS

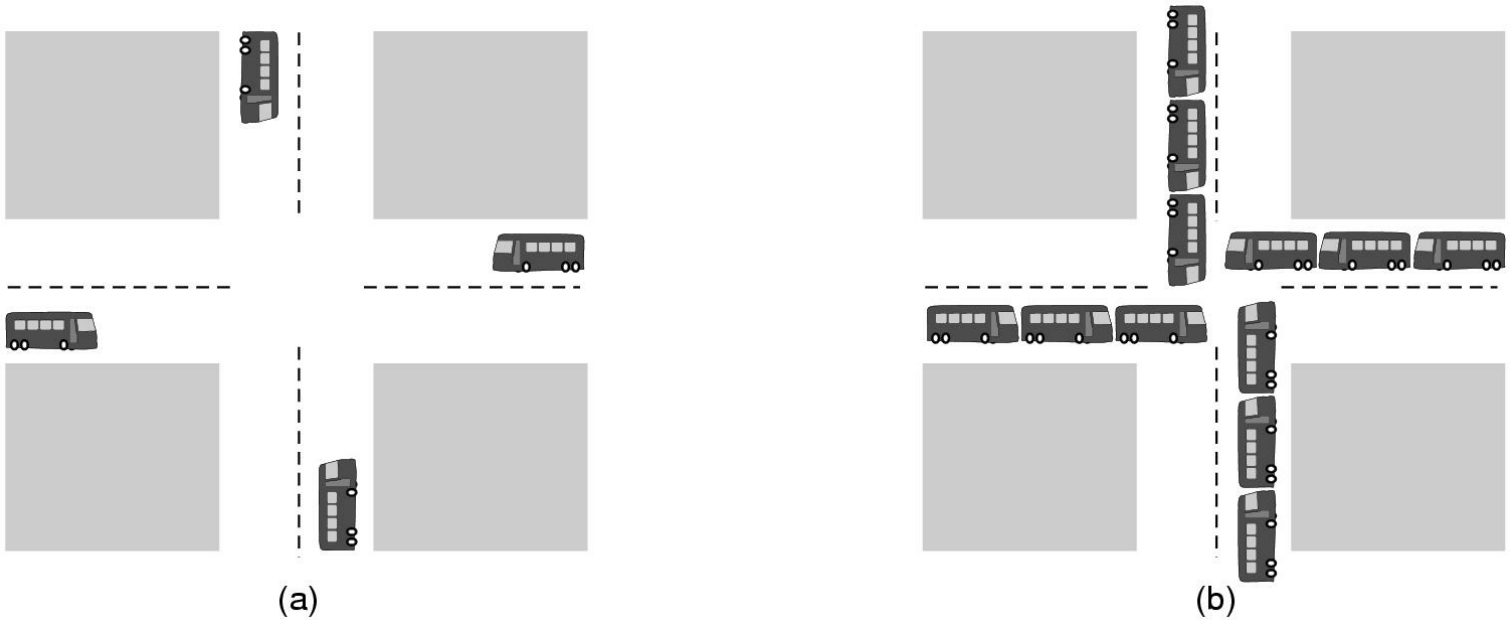
- Processes
- Address Spaces
- Input/Output
- Files
- Protection
- The Shell

Operating System Concepts (PROCESSES)



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

Operating System Concepts (DEADLOCK)



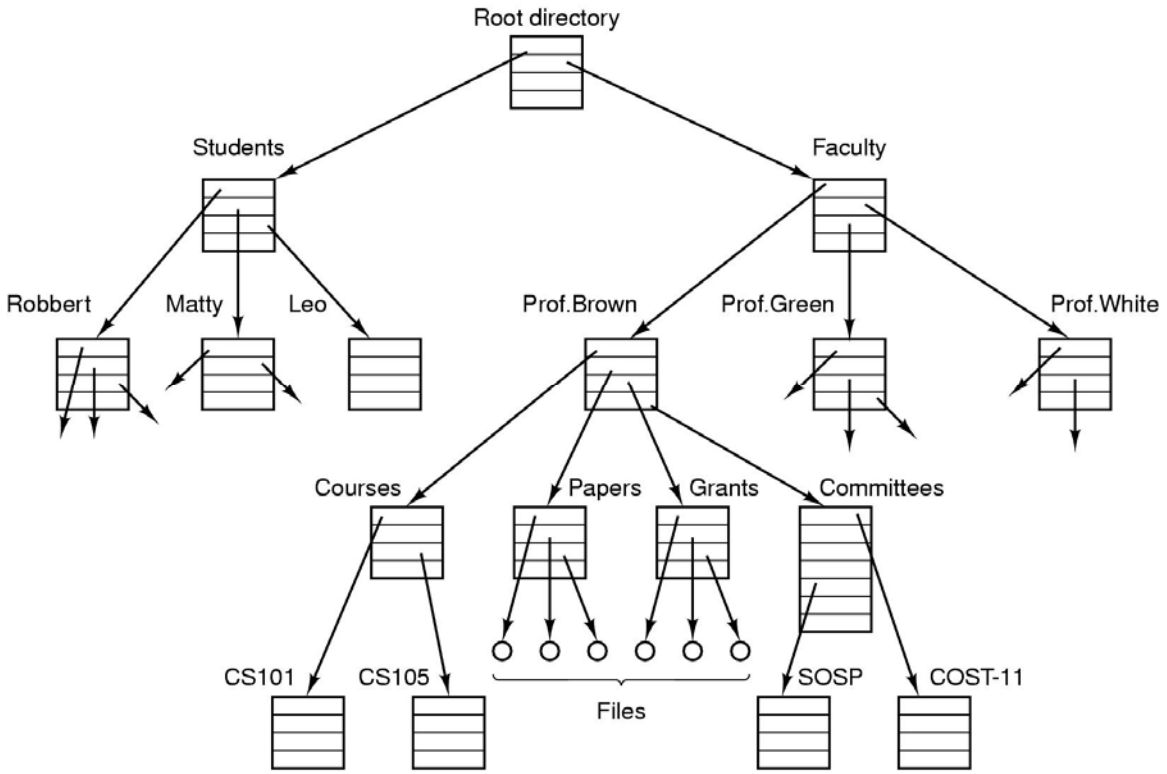
(a) A potential deadlock. (b) an actual deadlock.

OPERATING SYSTEM CONCEPTS
(MEMORY MANAGEMENT&INPUT/OUTPUT)

- Protection mechanism
- Virtual memory

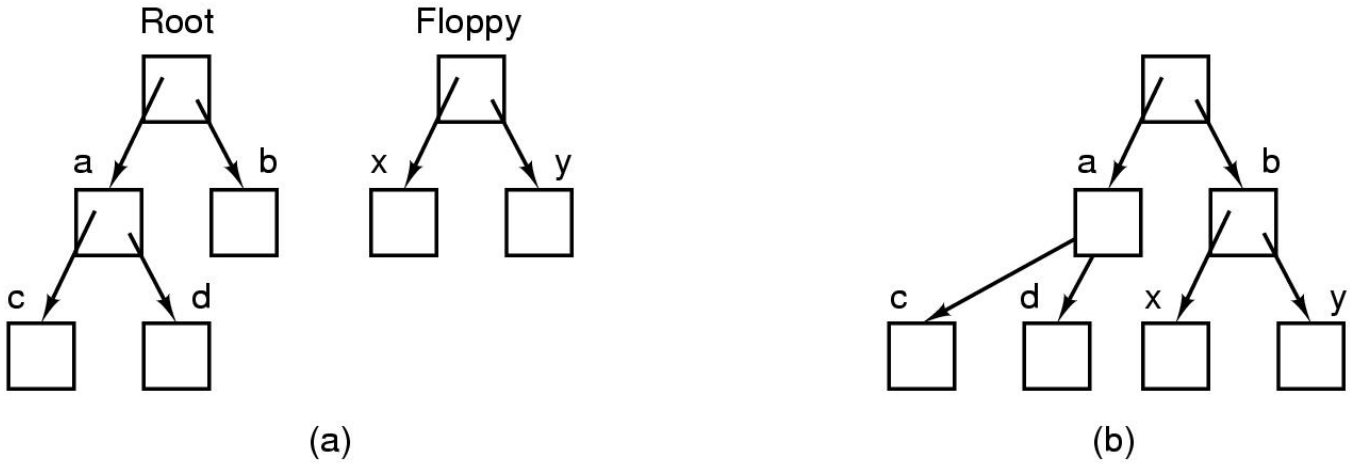
- I/O subsystem
- I/O software

Operating System Concepts (FILES1)



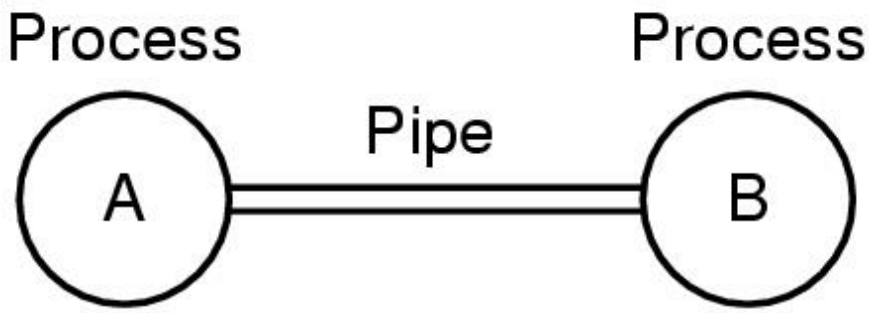
File system for a university department

Operating System Concepts (FILES2)



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

Operating System Concepts (FILES3)



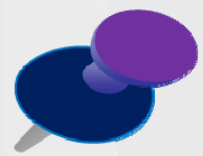
Two processes connected by a pipe

OPERATING SYSTEM CONCEPTS(SECURITY)

- Protection code

OPERATING SYSTEM CONCEPTS(THE SHELL)

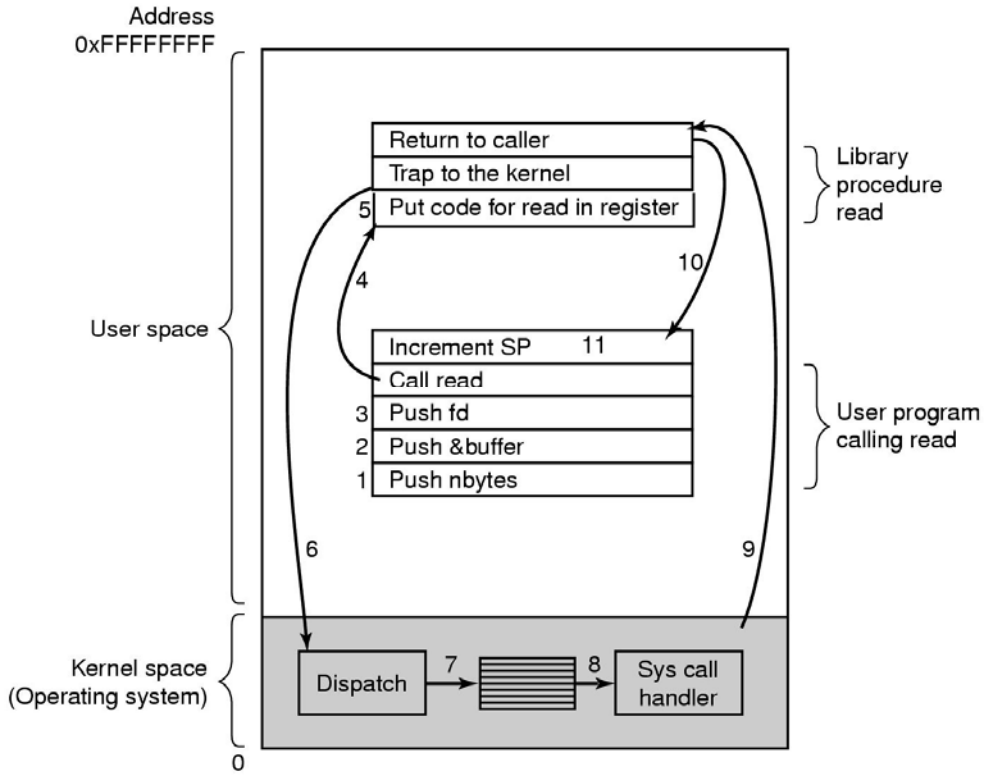
- Command input & interpreters
- Human & machine interface



一、操作系统概论

6、操作系统的系统调用

Steps in Making a System Call



There are 11 steps in making the system call 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

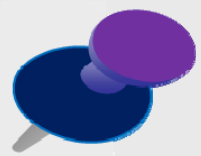
Some System Calls For Miscellaneous Tasks

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

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



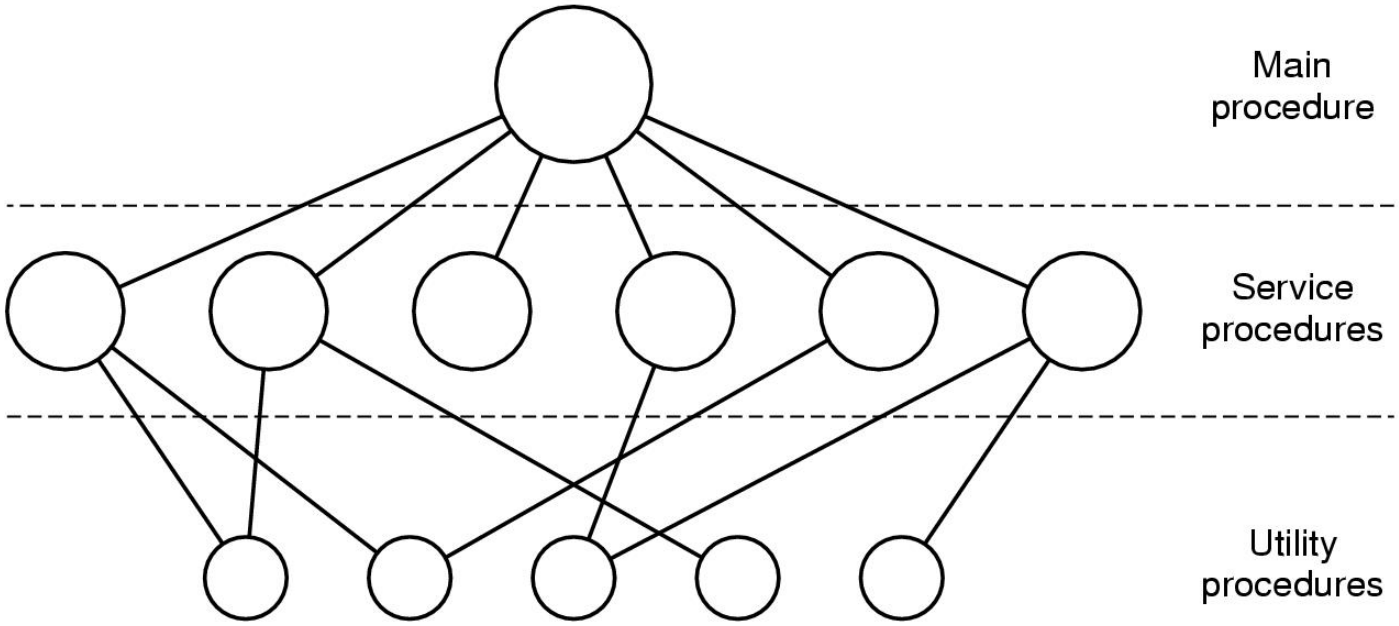
一、操作系统概论

7、操作系统的结构

OPERATING SYSTEM STRUCTURE

- Monolithic Systems
- Layered Systems
- Virtual Machines
- Exokernels
- Client-Server Model

Operating System Structure (1)



Simple structuring model for a monolithic system

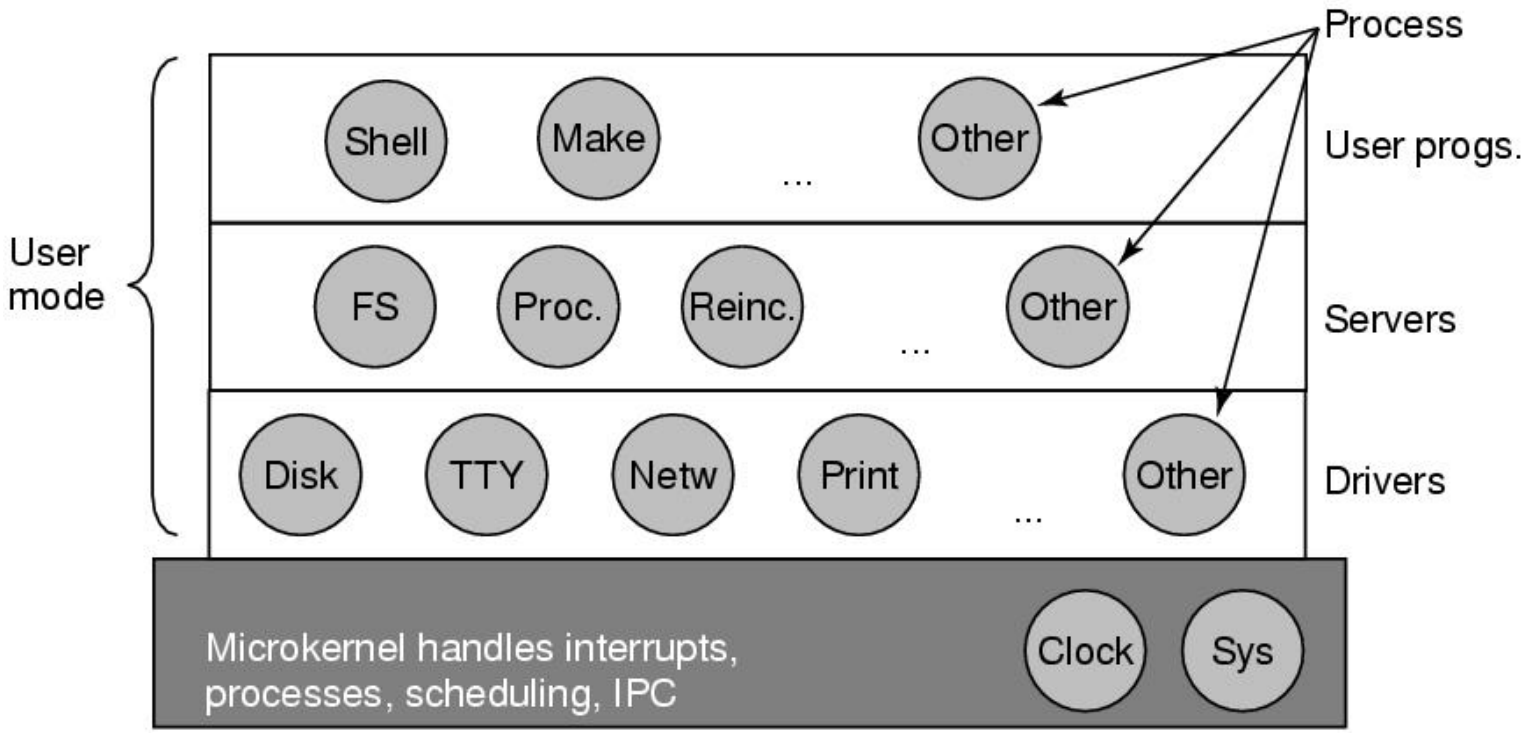
Operating System Structure (2)

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

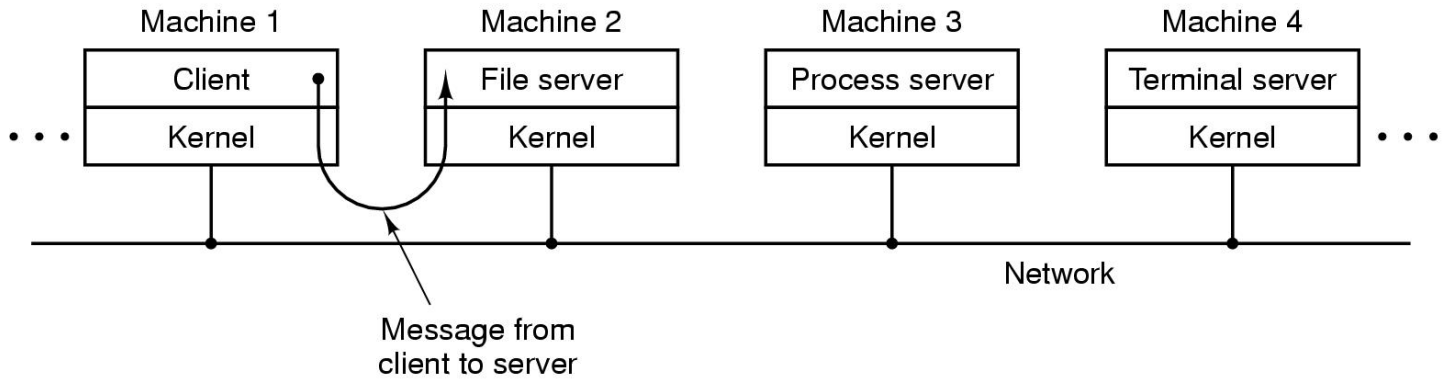
Operating System Structure (3)

Microkernels



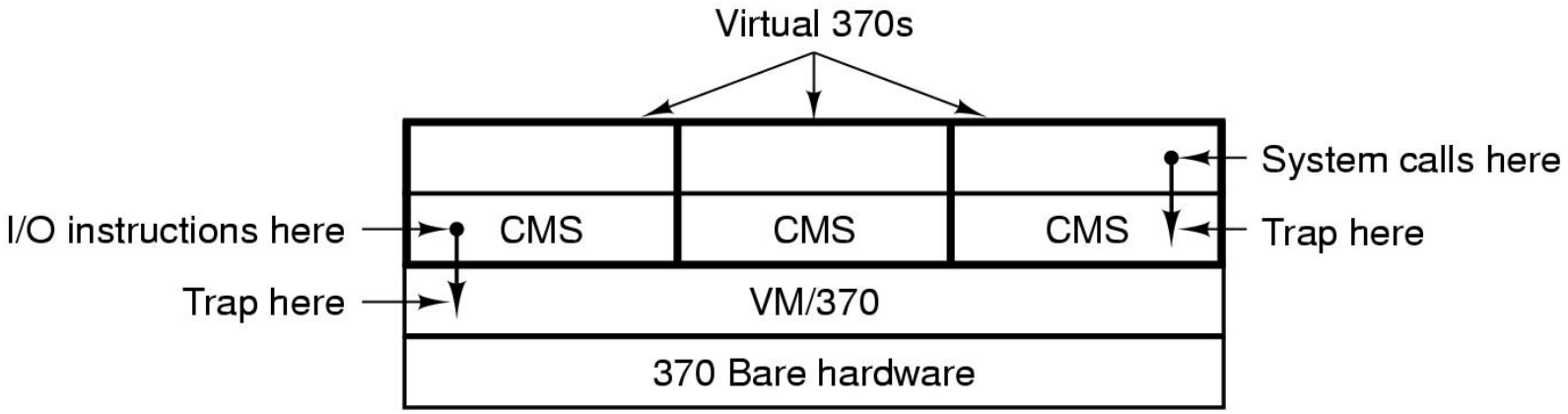
Structure of the MINIX 3 system.

Operating System Structure (4)

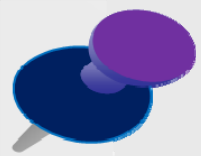


The client-server model in a distributed system

Operating System Structure (5)



Structure of VM/370 with CMS



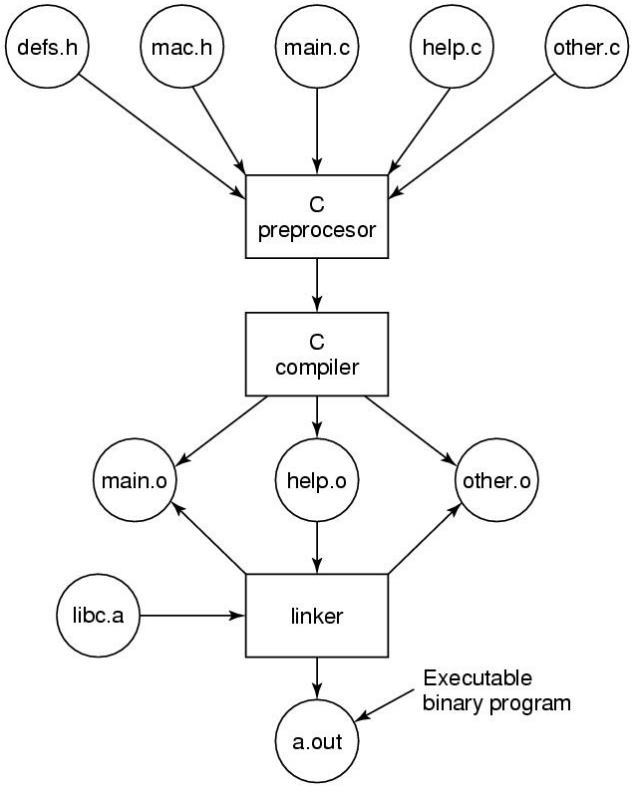
一、操作系统概论

8、操作系统与软件

The World According to C

- The C language
- Header files
- Large programming projects
- The model of run time

The Model of Run Time

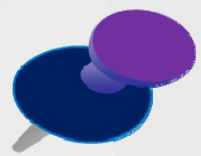


The process of compiling C and header files to make an executable.

Metric Units

Exp.	Explicit	Prefix	Exp.	Explicit	Prefix
10^{-3}	0.001	milli	10^{3}	1,000	Kilo
10^{-6}	0.000001	micro	10^{6}	1,000,000	Mega
10^{-9}	0.000000001	nano	10^{9}	1,000,000,000	Giga
10^{-12}	0.000000000001	pico	10^{12}	1,000,000,000,000	Tera
10^{-15}	0.000000000000001	femto	10^{15}	1,000,000,000,000,000	Peta
10^{-18}	0.000000000000000001	atto	10^{18}	1,000,000,000,000,000,000	Exa
10^{-21}	0.000000000000000000001	zepto	10^{21}	1,000,000,000,000,000,000,000	Zetta
10^{-24}	0.000000000000000000000001	yocto	10^{24}	1,000,000,000,000,000,000,000,000	Yotta

The metric prefixes



Q & A

Thanks !