Modern Operating Systems

Huo QiRun

Tel.No.: 68902762-229

Email ：jzhqr@163.com

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?

操作系统的工作

- 程序的执行
  - 负责启动每个程序，以及结束程序的工作
- 完成与硬件有关的工作
- 完成与应用无关的工作
  - 易于使用，基本服务，统一性
- 计算机系统的效率与安全问题

CONTENTS

- INTRODUCTION
- PROCESSES AND THREADS
- MEMORY MANAGEMENT
- FILE SYSTEM
- INPUT/OUTPUT
- DEADLOCK
- Virtualization and the Cloud~
- Multiple Processor Systems~
- Security~
- CASE STUDY 1:UNIX、Linux and Android*
- CASE STUDY 2:Windows8*
- OPERATING SYSTEM DESIGN*
- READING LIST AND BIBLIOGRAPHY*

预期收获

- 深入到操作系统内部，理解并掌握操作系统的基本概念、基本原理、设计方法和实现技术
- 思维方式的学习，权衡（Trade Off），抽象
- 系统级编程能力的提高
- 了解操作系统的演化过程、发展研究动向、新技术以及新思想

学习操作系统课的重要性

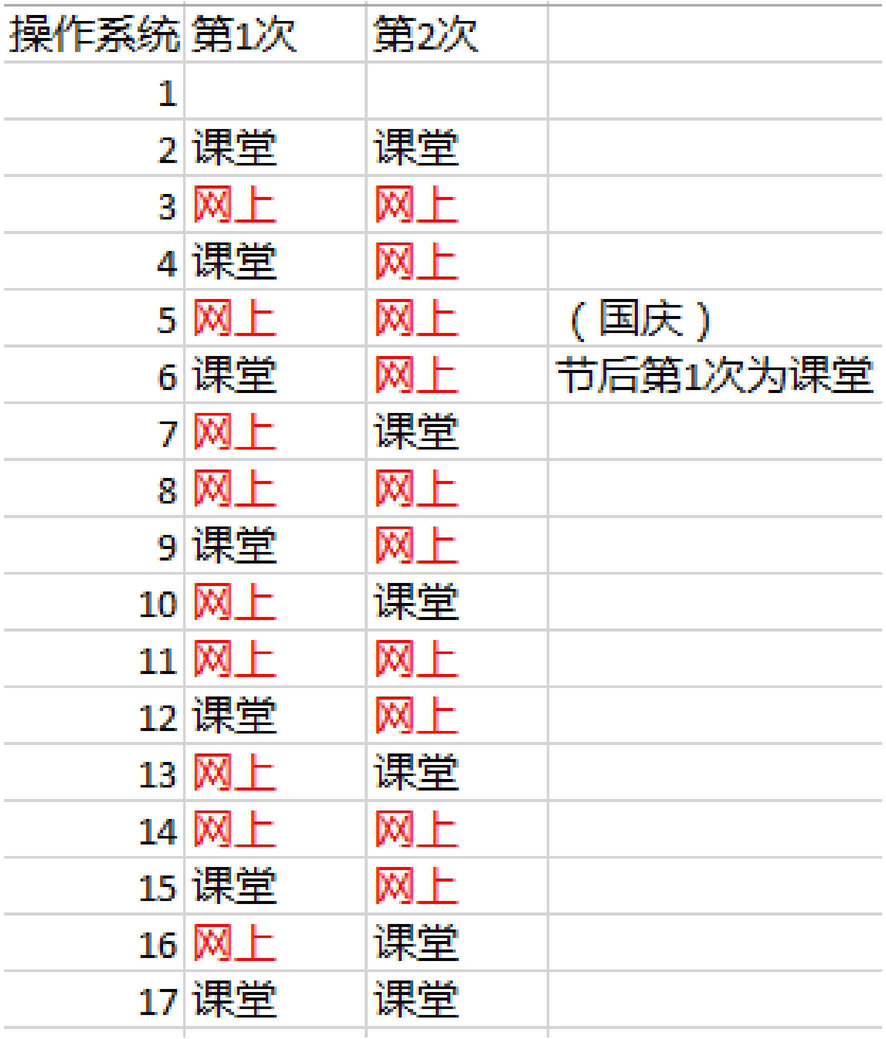
- 使用计算机必须要使用操作系统
  - 交互式
  - 编程、系统调用
- 操作系统的设计原理体现了软硬件最新的发展
  - 计算机体系结构（巨型机、大型机、微机、工作站、多核、GPU等）
  - 程序设计方法（并发、面向对象、结构化）
  - 加深对使用的OS的理解，有利于深入编程
  - 编程时借鉴操作系统的设计思想和算法
  - 大型系统的设计

HOW CAN I LEARN OS WELL?

- Study @ Class / Net
- Reading reference books after class
- Complete homework/exercise by yourself
- Summary

混合式课堂

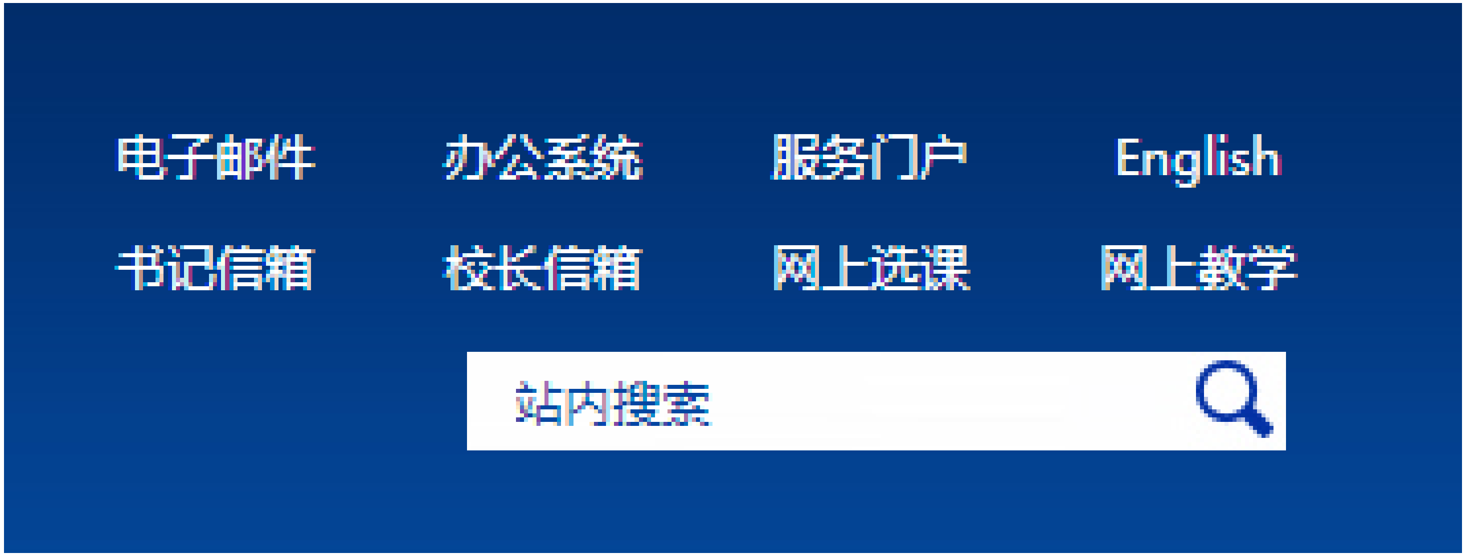
- 课程学习——网上为主
  - 视频、教材、课件等
  - 问题驱动、自主学习
  - 在线交流
- 总结深入——课堂为主
  - 反馈问题
  - 深入讨论
  - 知识梳理
- 应用巩固——网上为主
  - 课后作业
  - 章节测试



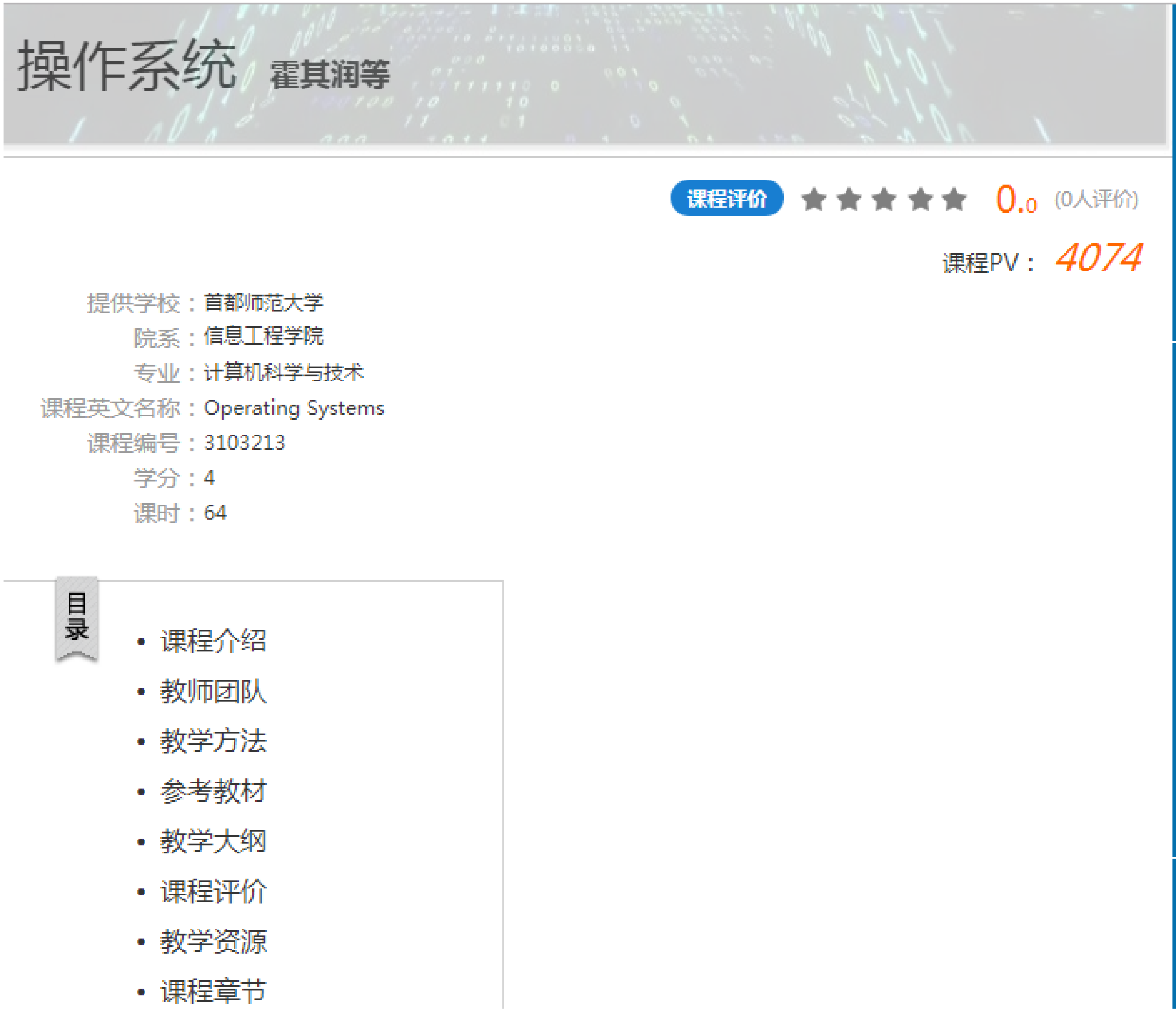
操作系统	第1次	第2次	
1			
2	课堂	课堂	
3	网上	网上	
4	课堂	网上	
5	网上	网上	（国庆）
6	课堂	网上	节后第1次为课堂
7	网上	课堂	
8	网上	网上	
9	课堂	网上	
10	网上	课堂	
11	网上	网上	
12	课堂	网上	
13	网上	课堂	
14	网上	网上	
15	课堂	网上	
16	网上	课堂	
17	课堂	课堂	

师星学堂

学校首页——网上教学——校内在线课程



师星学堂



操作系统 霍其润等

课程评价 ★★★★★ 0.0 (0人评价)

课程PV： 4074

提供学校：首都师范大学

院系：信息工程学院

专业：计算机科学与技术

课程英文名称：Operating Systems

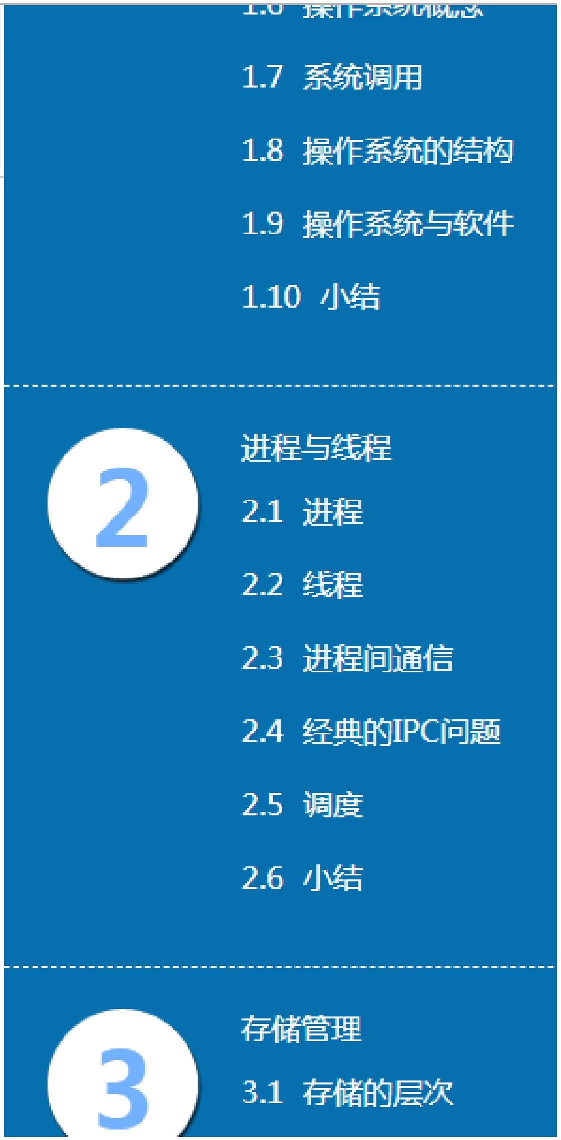
课程编号：3103213

学分：4

课时：64

目录

- 课程介绍
- 教师团队
- 教学方法
- 参考教材
- 教学大纲
- 课程评价
- 教学资源
- 课程章节



1.6 操作系统概念

1.7 系统调用

1.8 操作系统的结构

1.9 操作系统与软件

1.10 小结

2 进程与线程

2.1 进程

2.2 线程

2.3 进程间通信

2.4 经典的IPC问题

2.5 调度

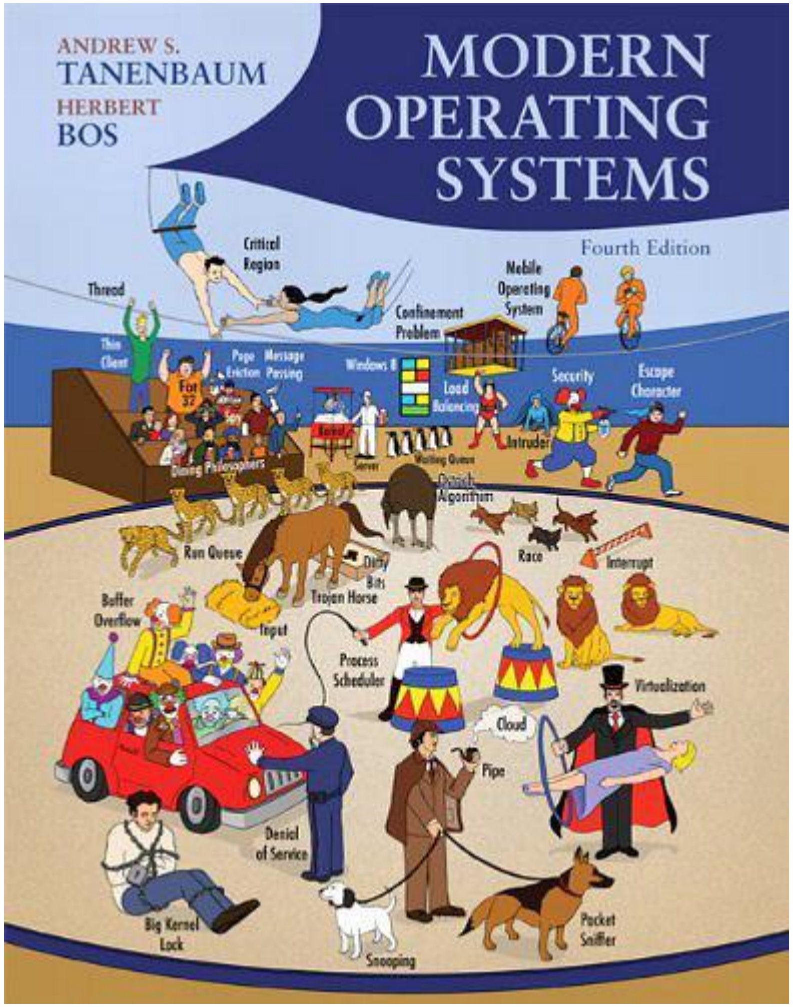
2.6 小结

3 存储管理

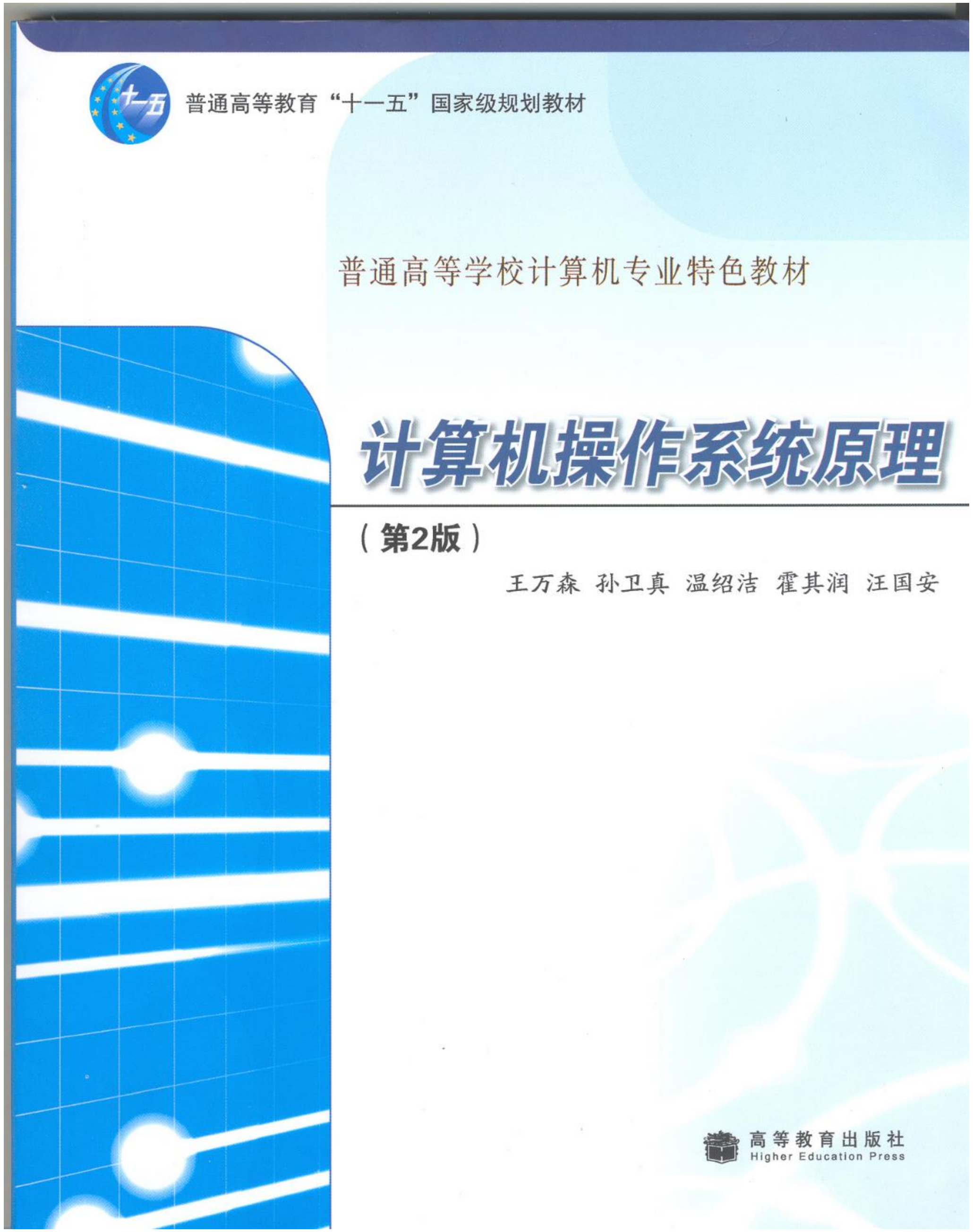
3.1 存储的层次

TEACHING MATERIAL

- Andrew S. Tanenbaum
- Modern Operating System 4nd edition



REFERENCE BOOKS



普通高等教育“十一五”国家级规划教材

普通高等学校计算机专业特色教材

计算机操作系统原理

（第2版）

王万森 孙卫真 温绍洁 霍其润 汪国安

高等教育出版社
Higher Education Press

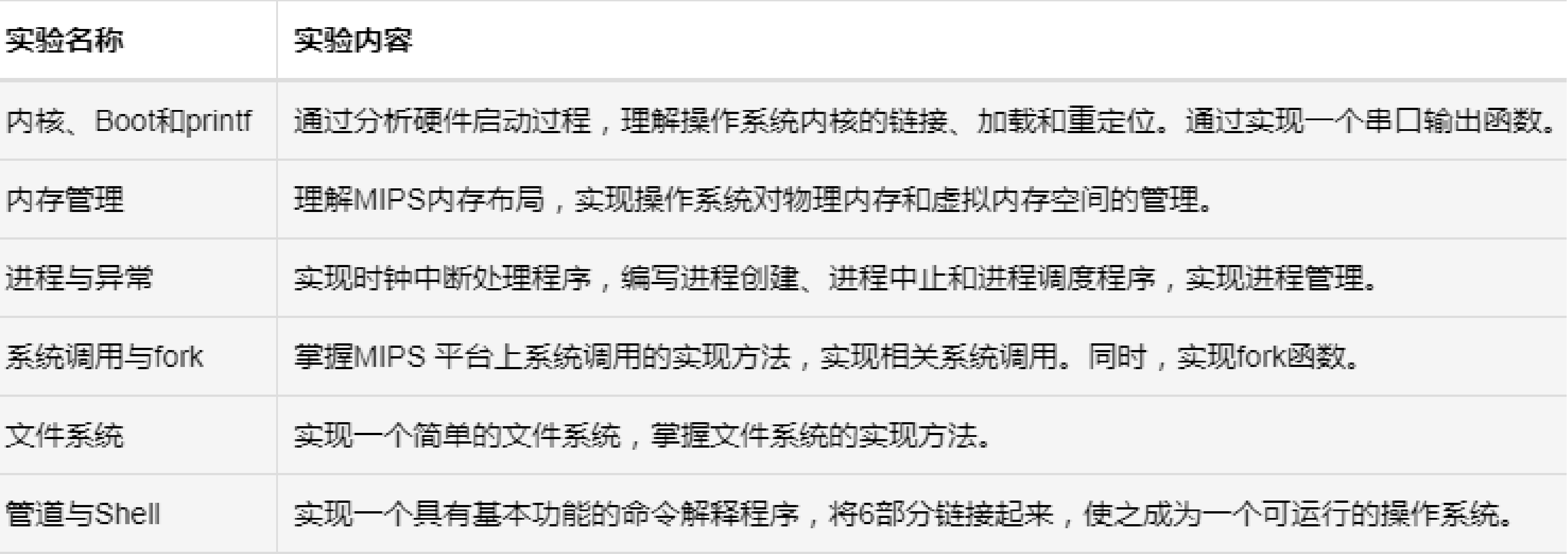
ACHIEVEMENT

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

配套实验

- 学时：32
- 目的
  - **从程序员的角度感知、理解操作系统**
  - 从设计者的角度实现操作系统

实验名称	实验内容
进程观测	熟悉实验环境,了解命令行等接口方式，较好地理解进程的概念。
进程控制	学习创建进程、观察进程和终止进程的程序设计方法。
线程同步	通过对事件和互斥体对象的使用，来加深对线程同步的理解。
线程间通信	通过对文件和文件映射对象的使用，来加深对线程同步的理解。



实验名称	实验内容
内核、Boot和printf	通过分析硬件启动过程，理解操作系统内核的链接、加载和重定位。通过实现一个串口输出函数。
内存管理	理解MIPS内存布局，实现操作系统对物理内存和虚拟内存空间的管理。
进程与异常	实现时钟中断处理程序，编写进程创建、进程中止和进程调度程序，实现进程管理。
系统调用与fork	掌握MIPS 平台上系统调用的实现方法，实现相关系统调用。同时，实现fork函数。
文件系统	实现一个简单的文件系统，掌握文件系统的实现方法。
管道与Shell	实现一个具有基本功能的命令解释程序，将6部分链接起来，使之成为一个可运行的操作系统。

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)

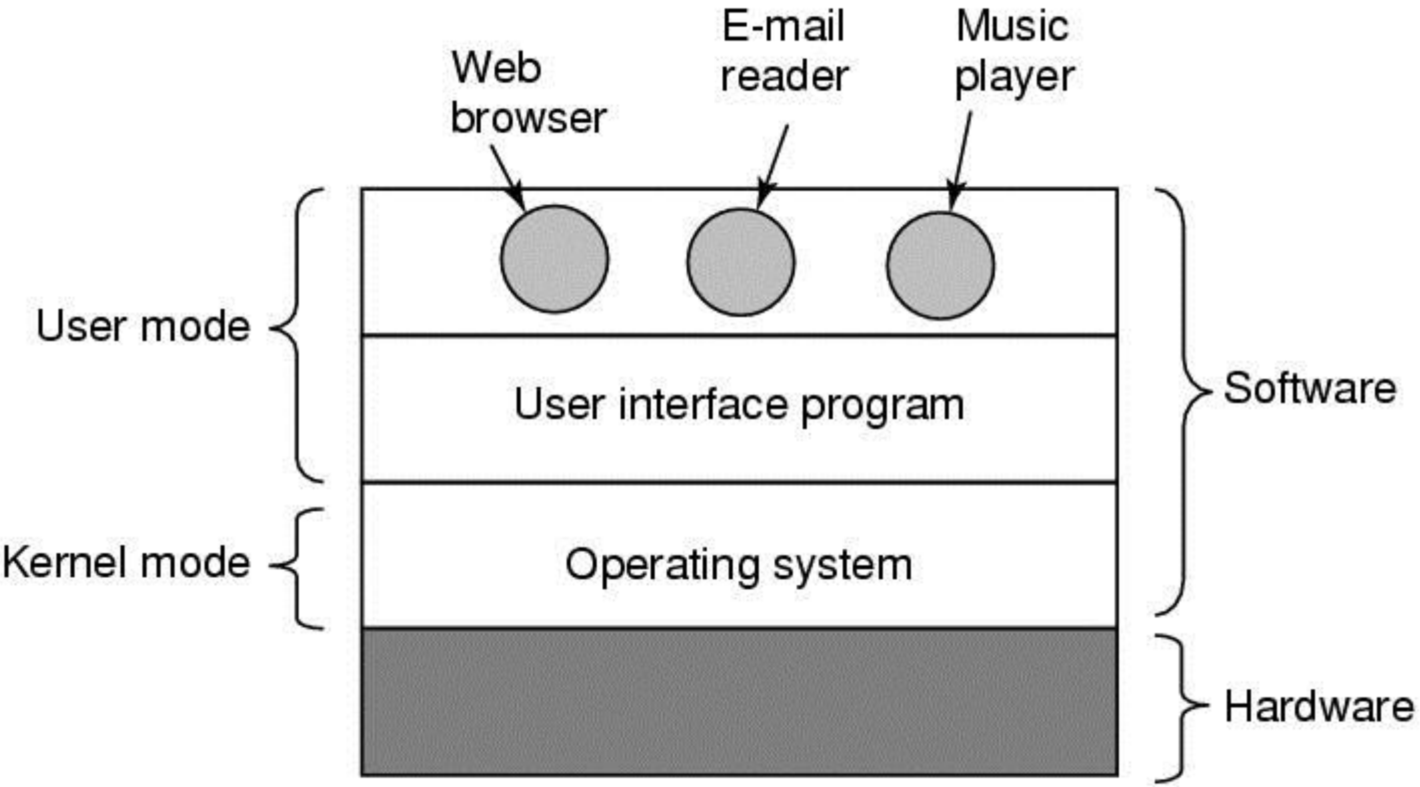


Figure 1-1. Where the operating system fits in.

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

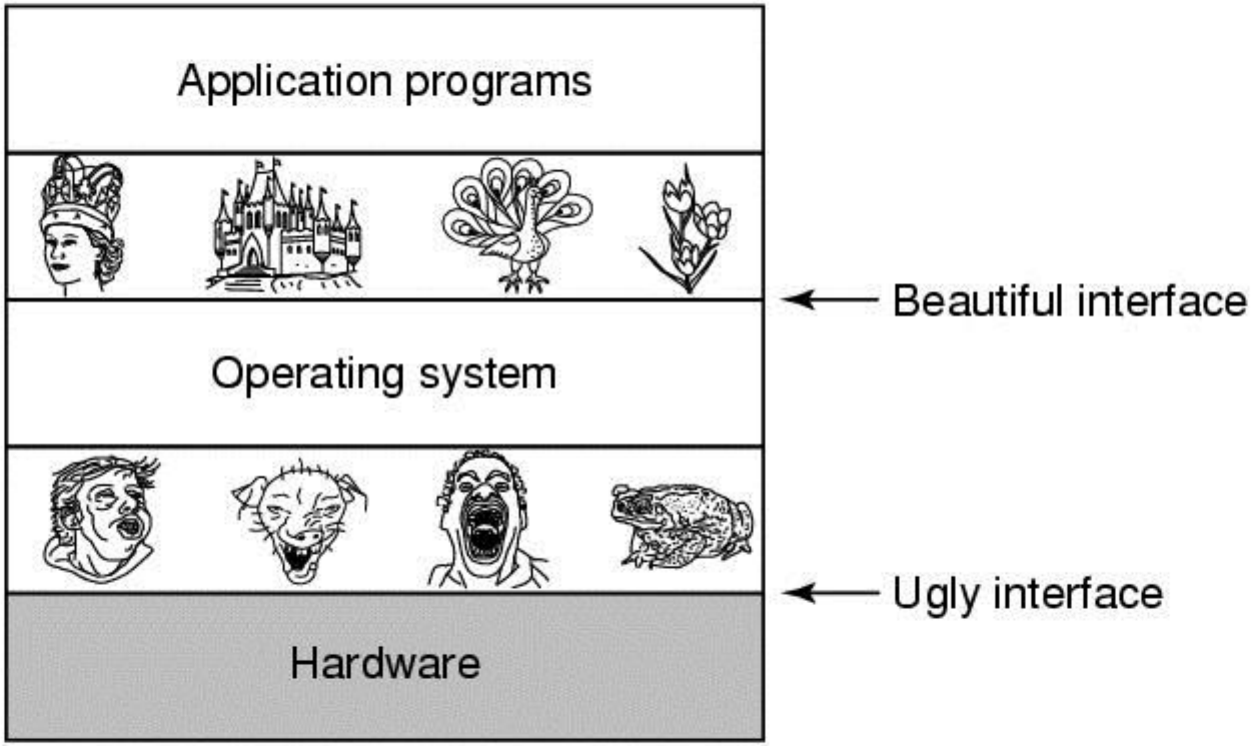


Figure 1-2. Operating systems turn ugly hardware into beautiful abstractions.

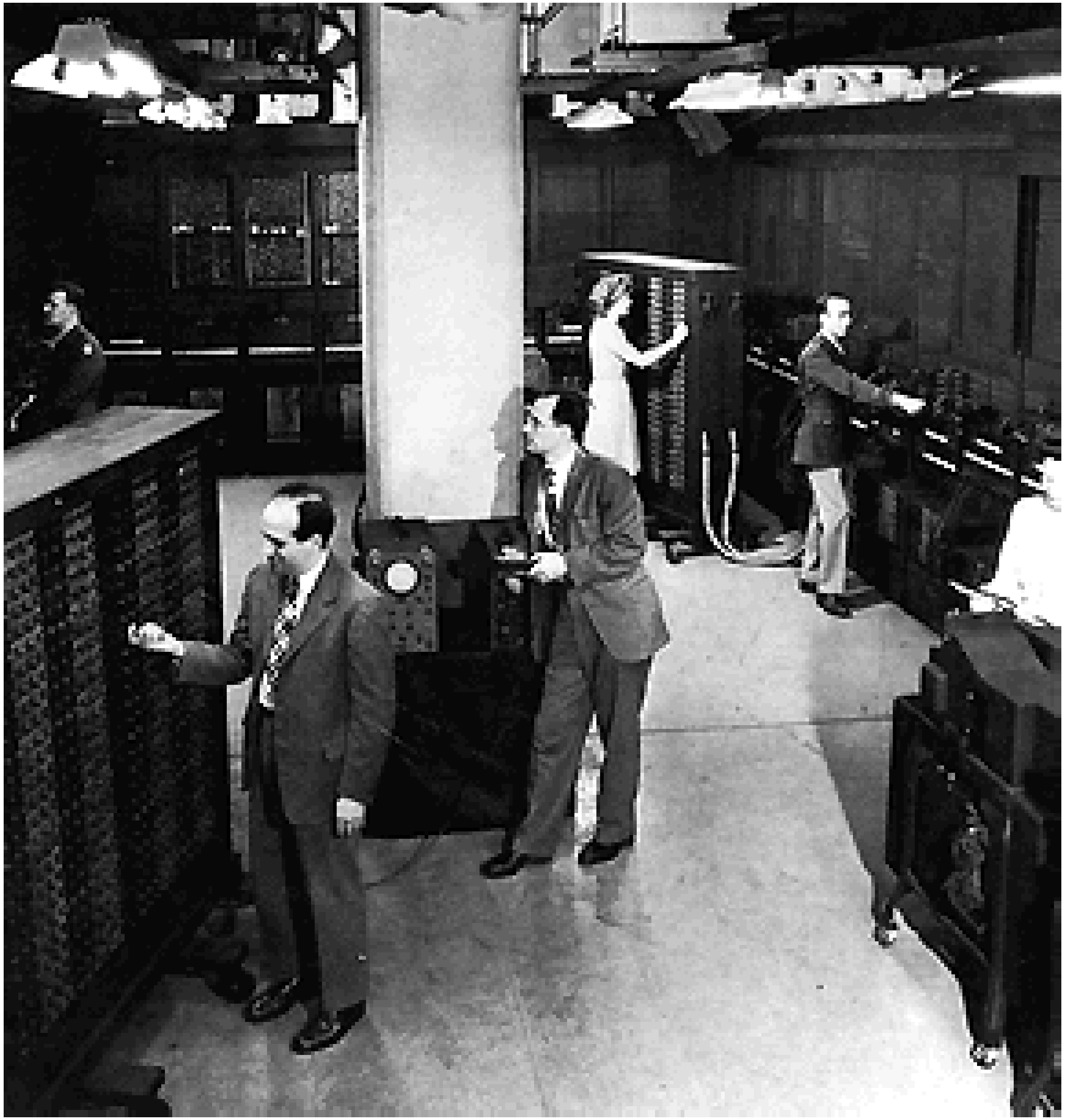
The Operating System as a Resource Manager

- 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

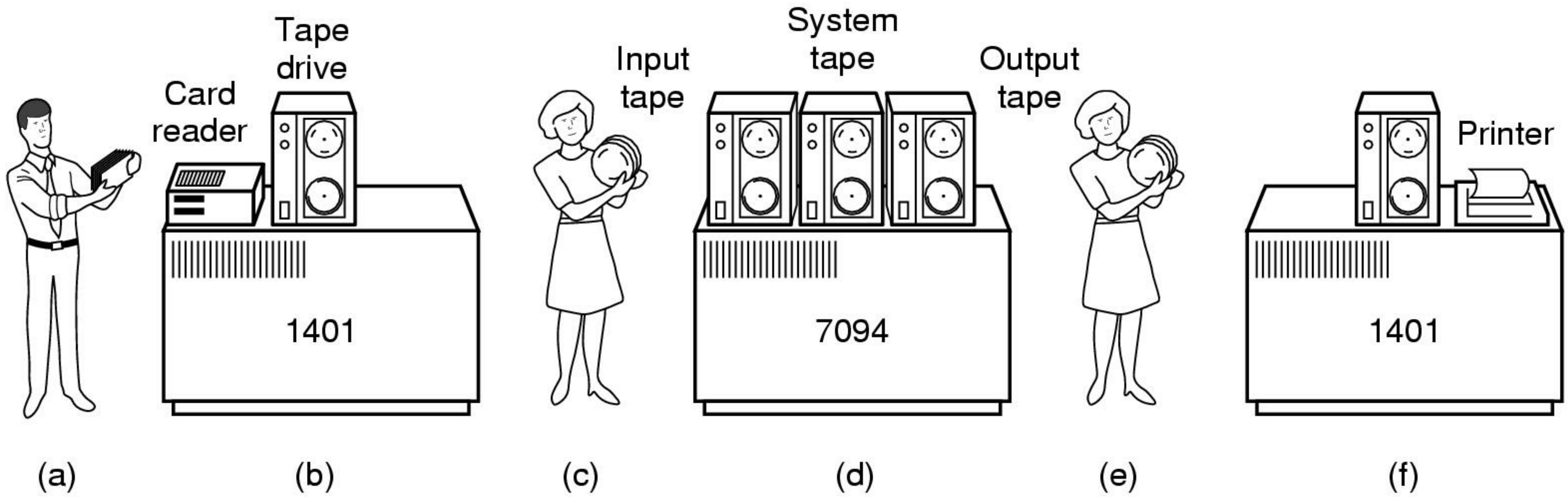
History of Operating Systems

- 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
- The fifth generation (1990–present)
  - mobile computers

History of Operating Systems (1)



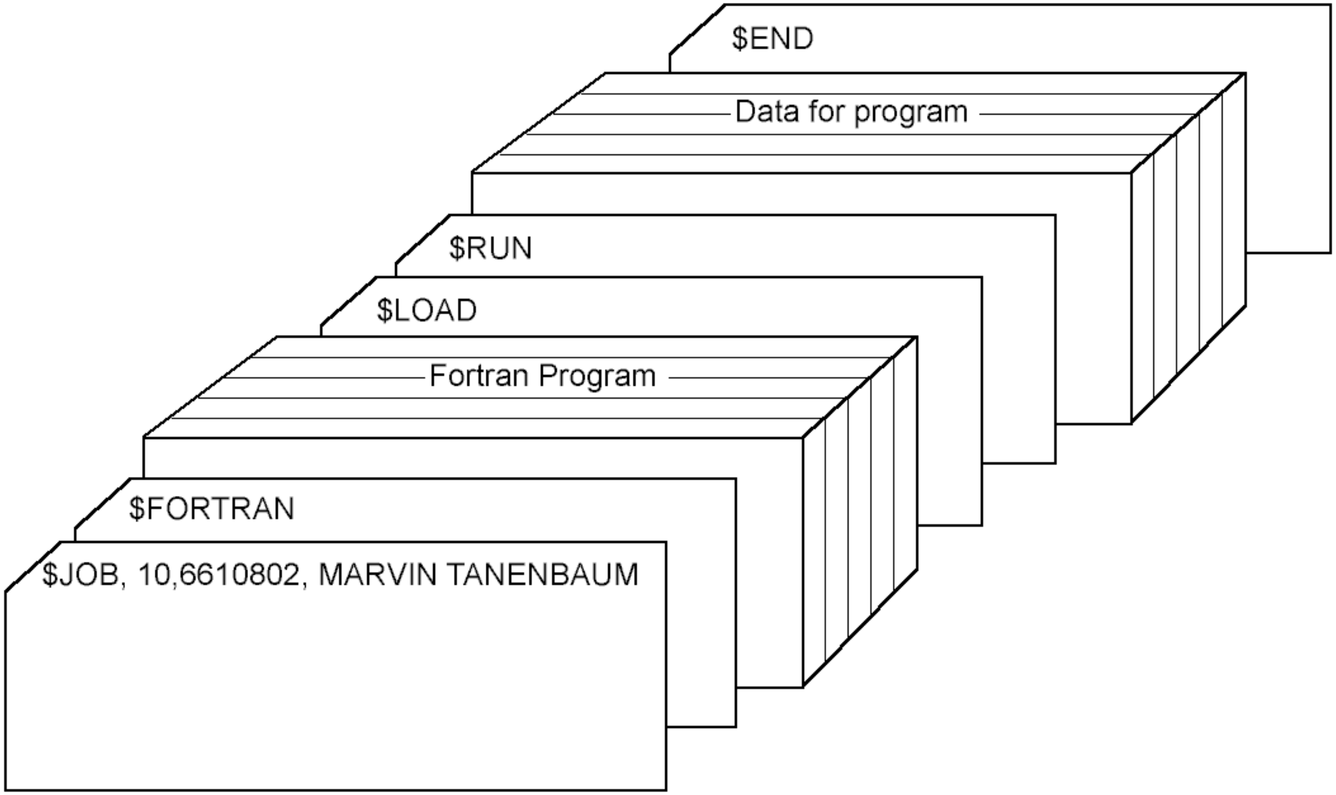
History of Operating Systems (2)



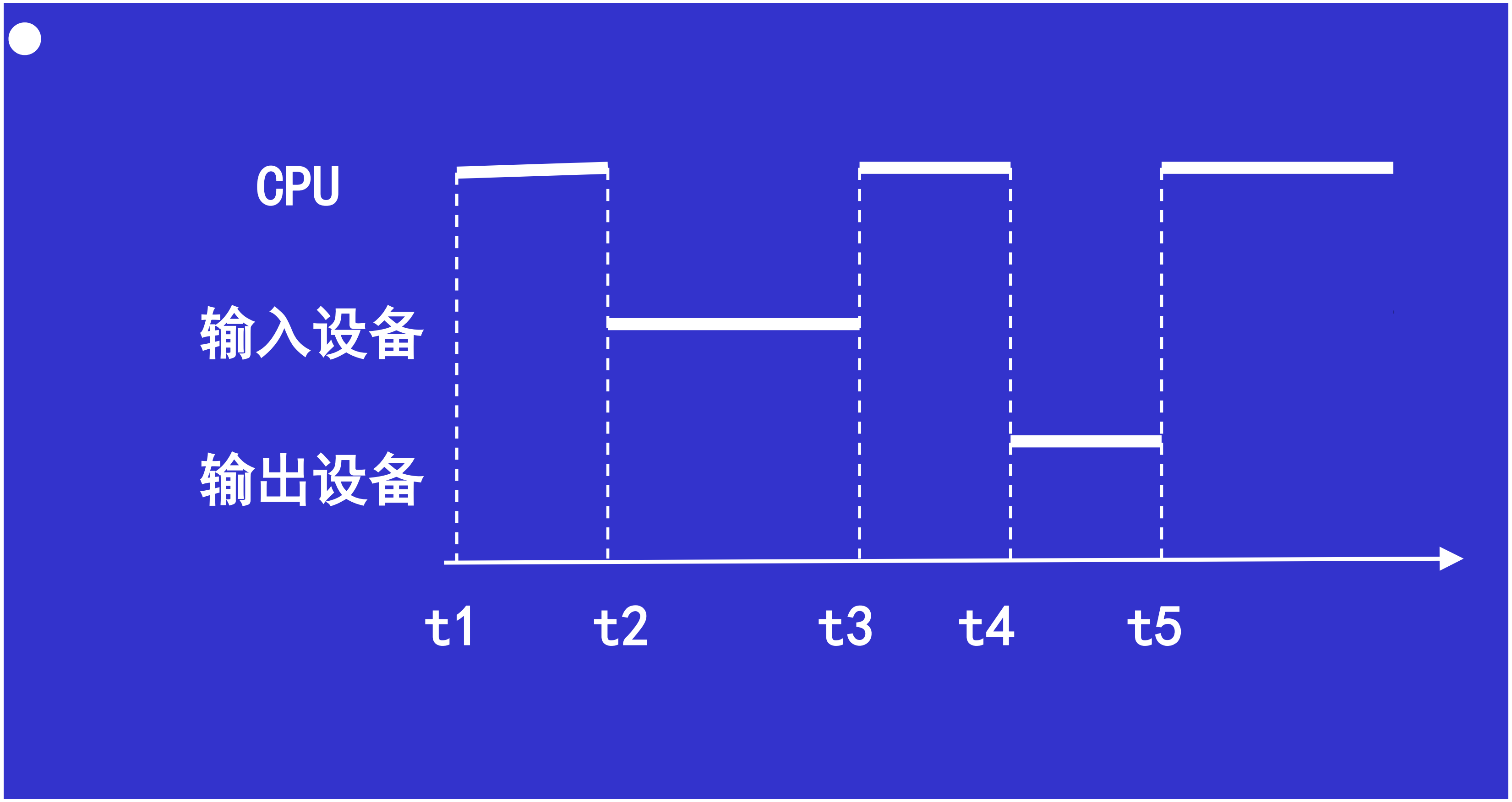
Early batch system

- 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)

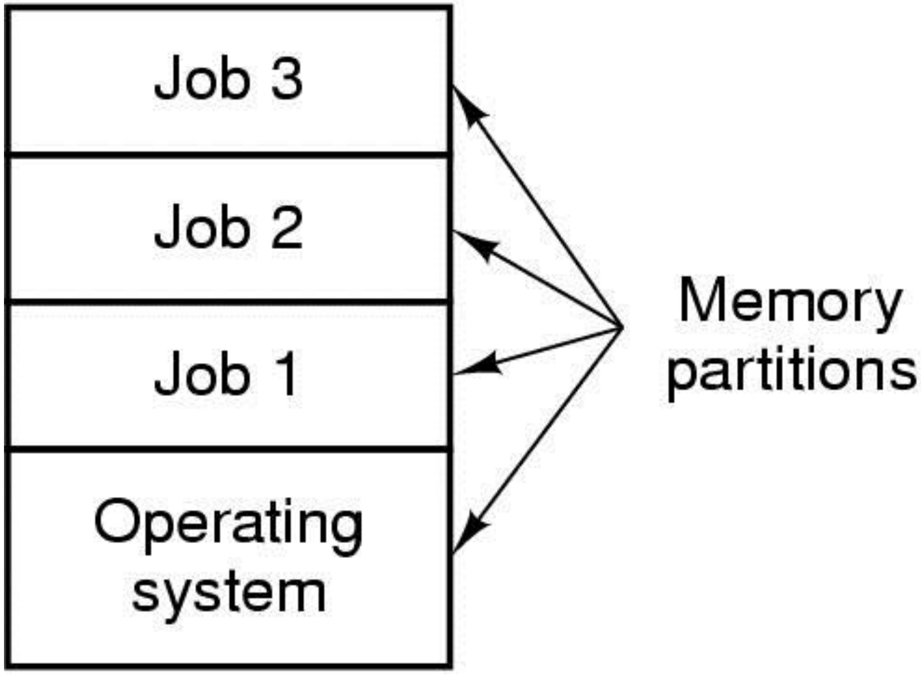


- Structure of a typical FMS job – 2nd generation

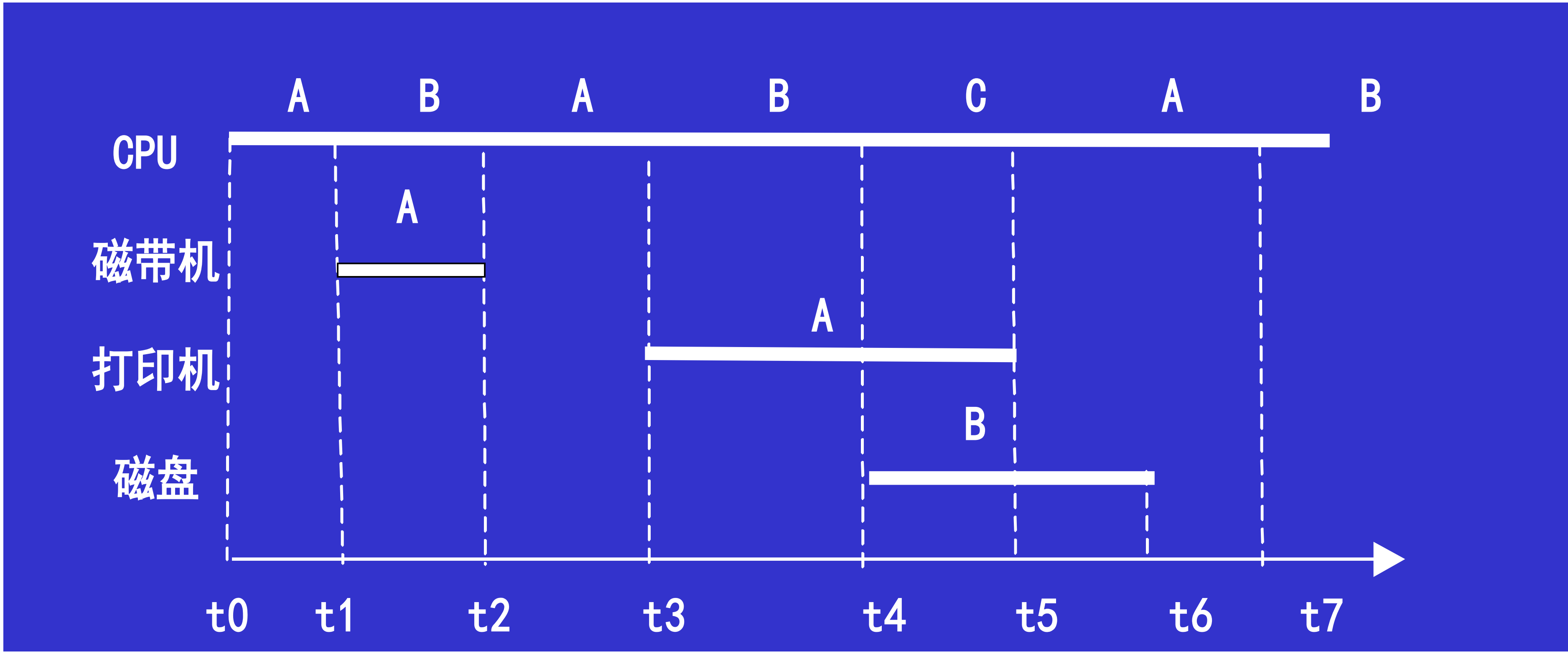


单道

History of Operating Systems (3)



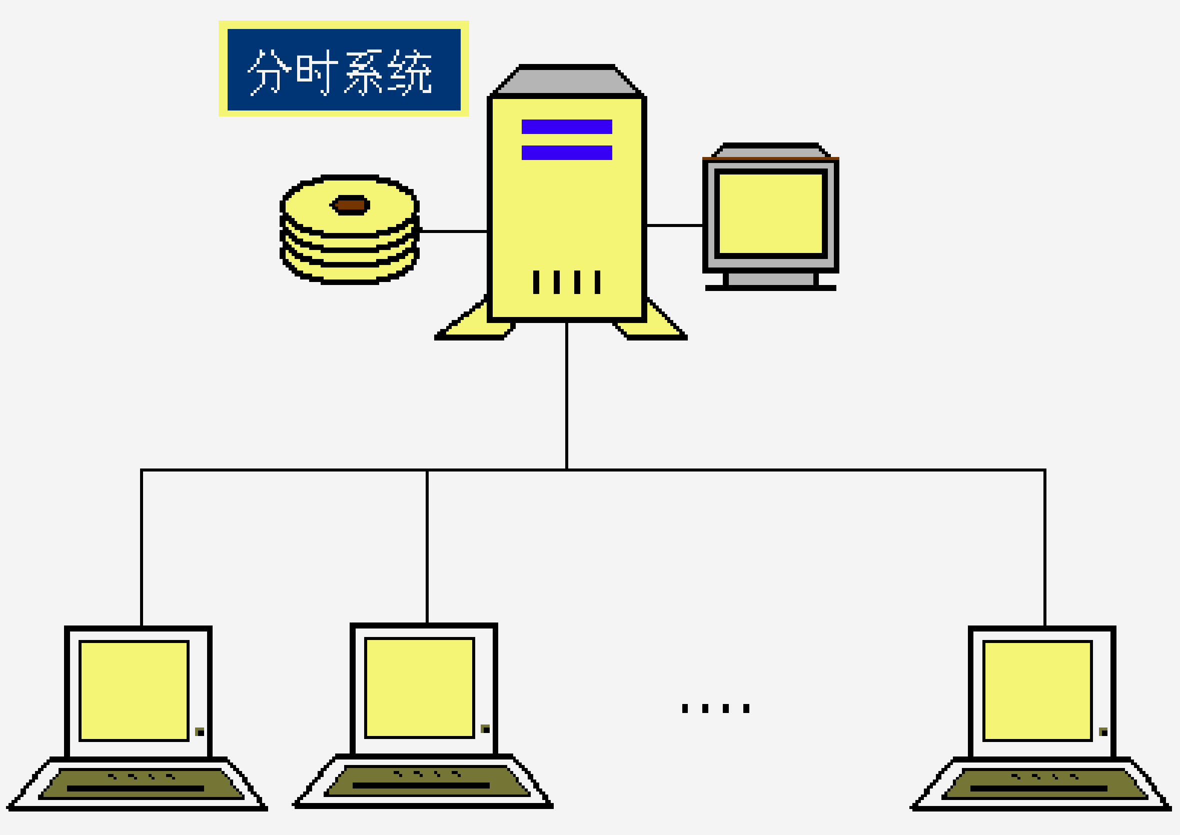
- Multiprogramming system
  - three jobs in memory – 3rd generation



出现：作业管理、处理机管理、存储管理、设备管理、文件系统管理等

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

History of Operating Systems (5)

- Symbian、Windows Phone
- iOS
- Andriod
- HarmonyOS

The Operating System Zoo

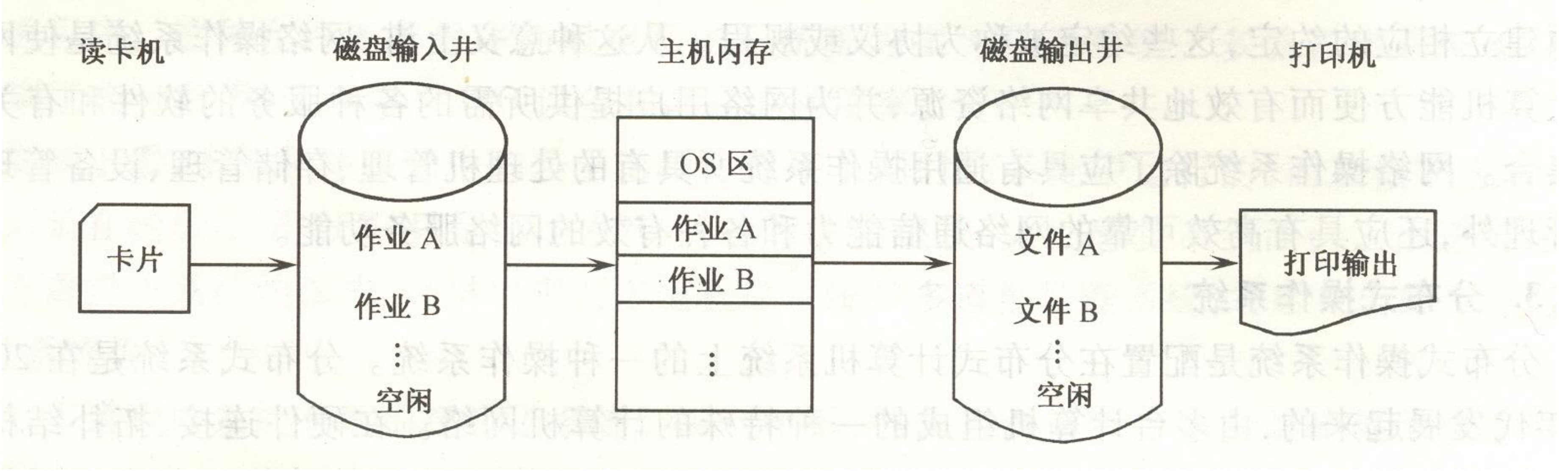
- Mainframe Operating Systems(OS/390; UNIX)
- Server Operating Systems(LINUX;WIN200x)
- Multiprocessor Operating Systems
- Personal Computer Operating Systems
- Handheld Computer Operating Systems(PDA)
- Embedded Operating Systems
- Sensor Node Operating Systems
- Real-Time Operating Systems
- Smart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

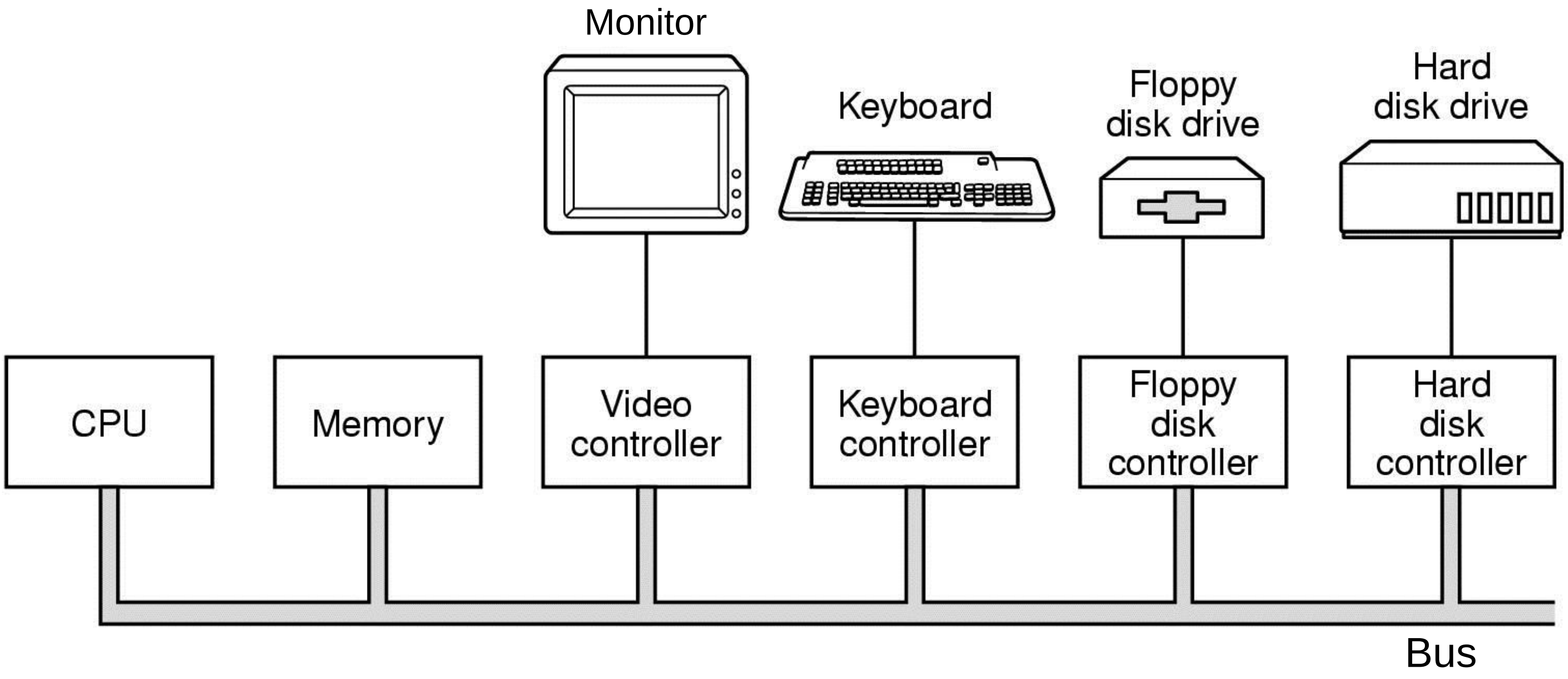
- 人机交互性好：在调试和运行程序时由用户自己操作
- 共享主机：多个用户同时使用
- 用户独立性：对每个用户而言好象独占主机

现在的许多操作系统都具有分时处理的功能。

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。

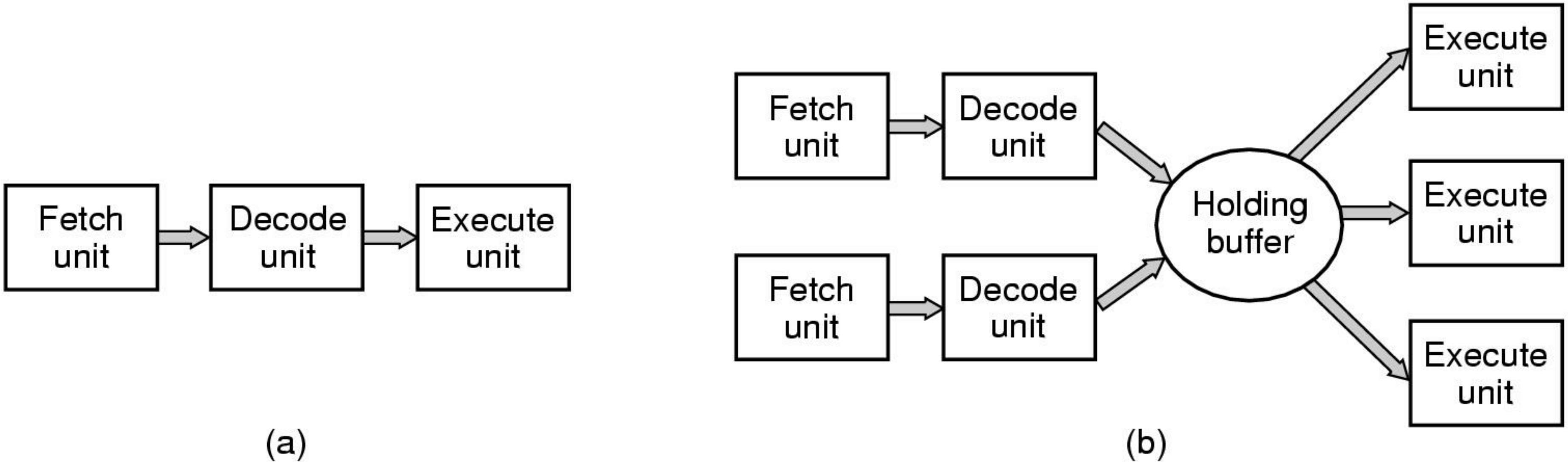
Computer Hardware Review (1)



- 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

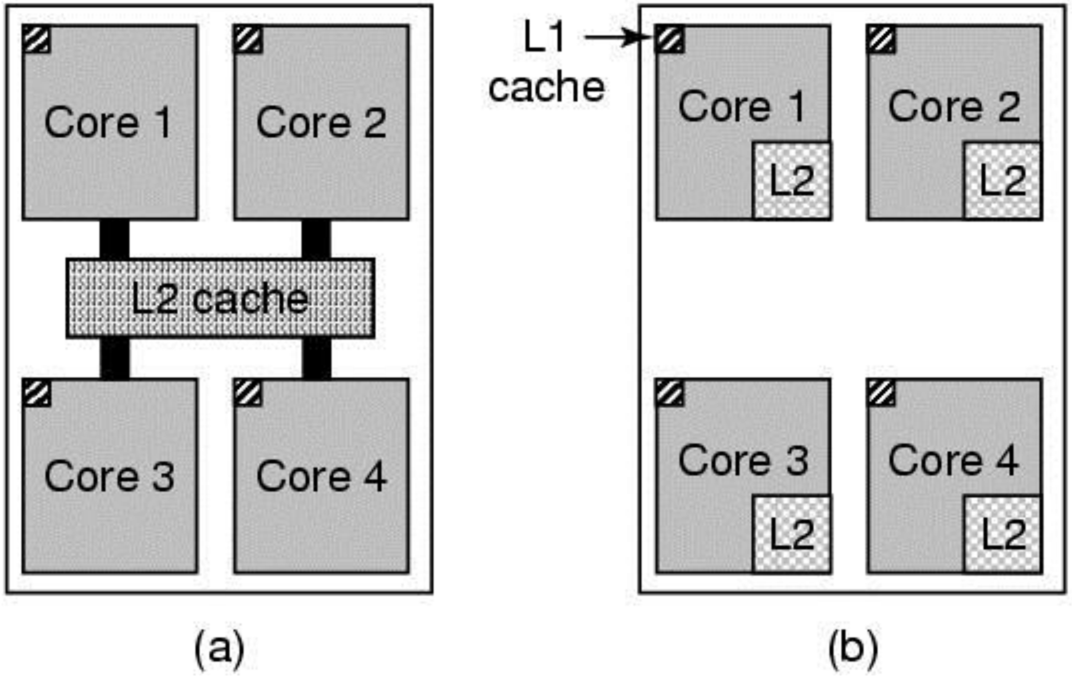
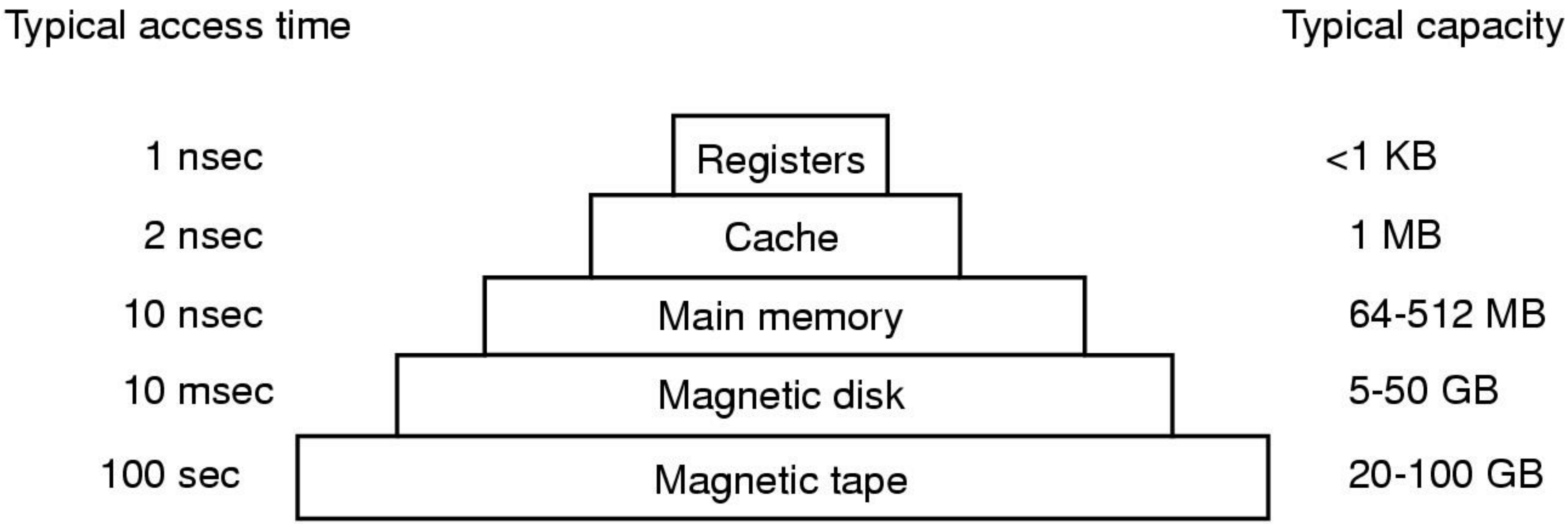


Figure 1-8. (a) A quad-core chip with a shared L2 cache.
(b) A quad-core chip with separate L2 caches.

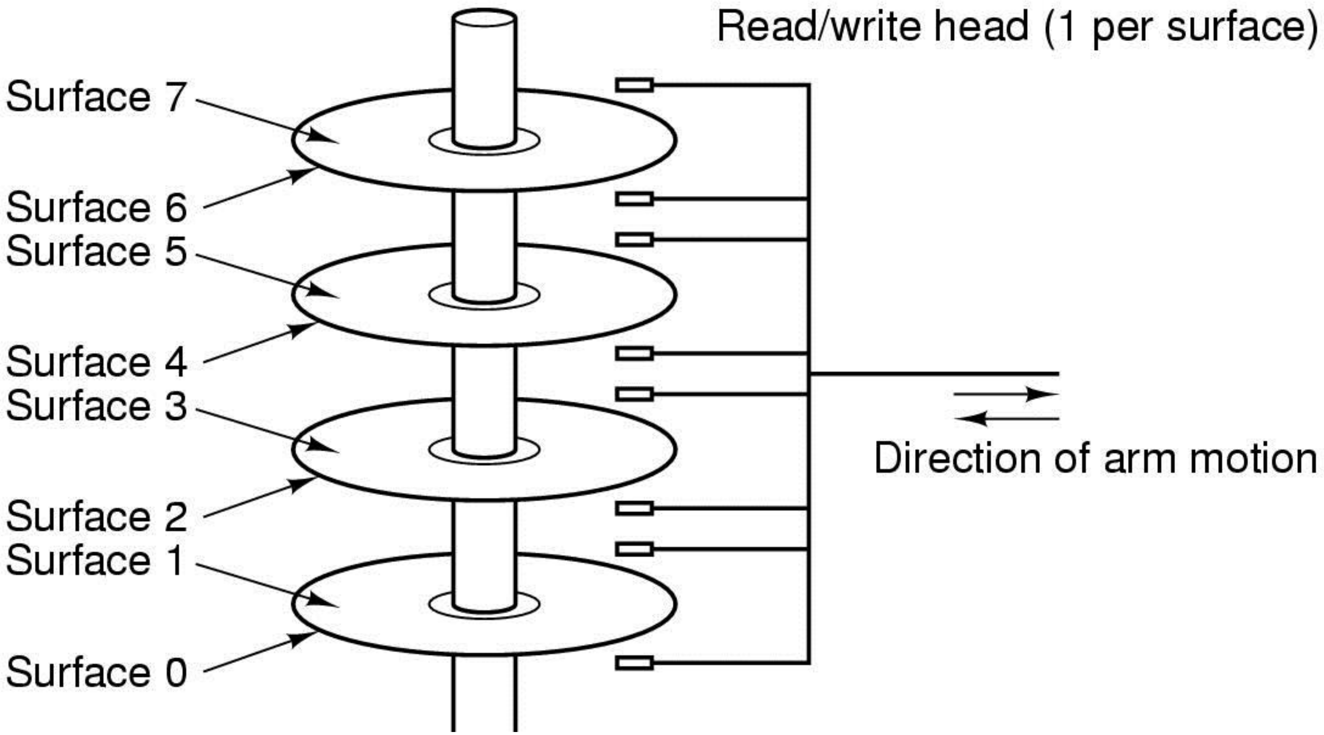
Computer Hardware Review (3)



Typical access time		Typical capacity
1 nsec	Registers	<1 KB
2 nsec	Cache	1 MB
10 nsec	Main memory	64-512 MB
10 msec	Magnetic disk	5-50 GB
100 sec	Magnetic tape	20-100 GB

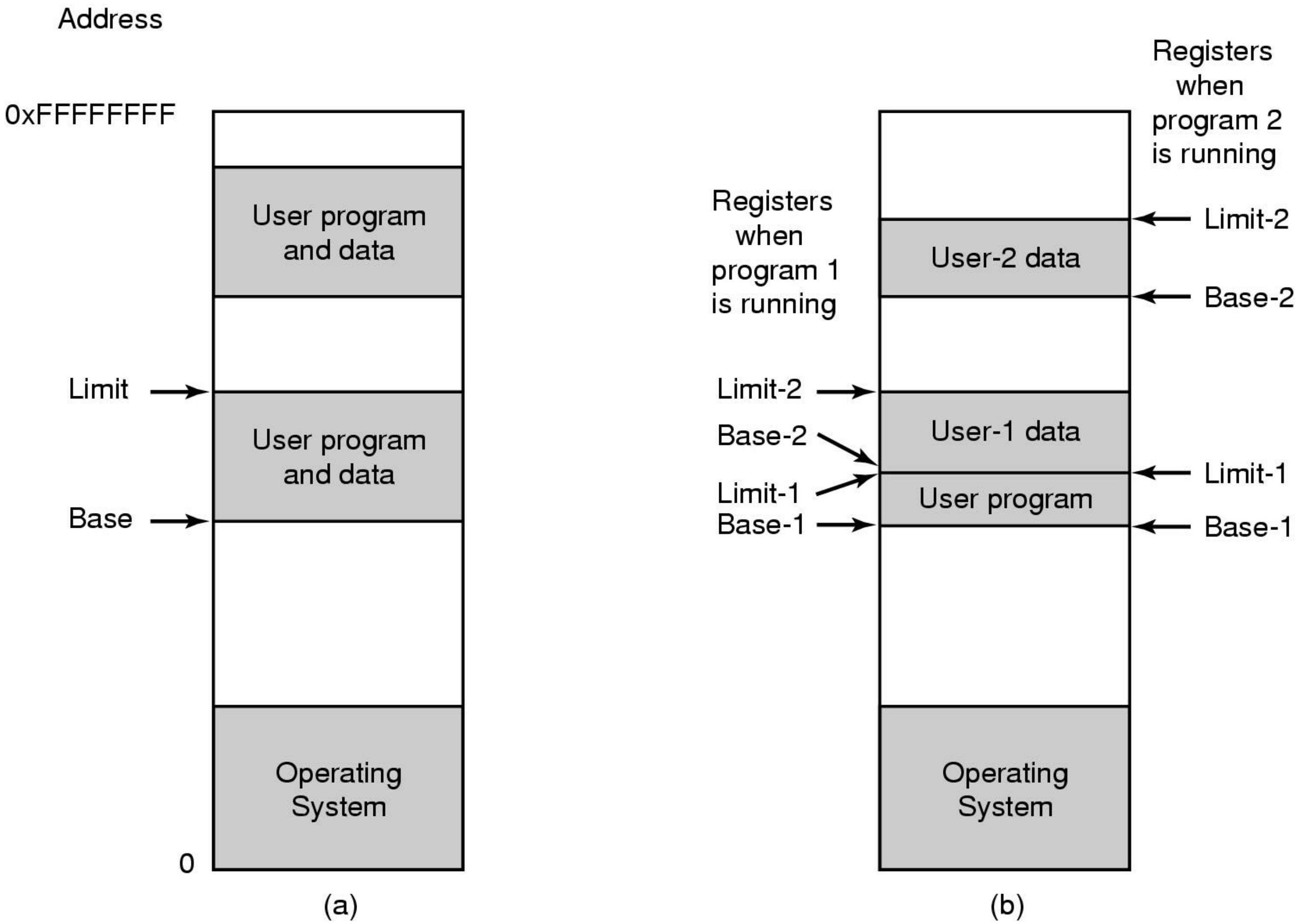
- Typical memory hierarchy
  - numbers shown are rough approximations

Computer Hardware Review (4)



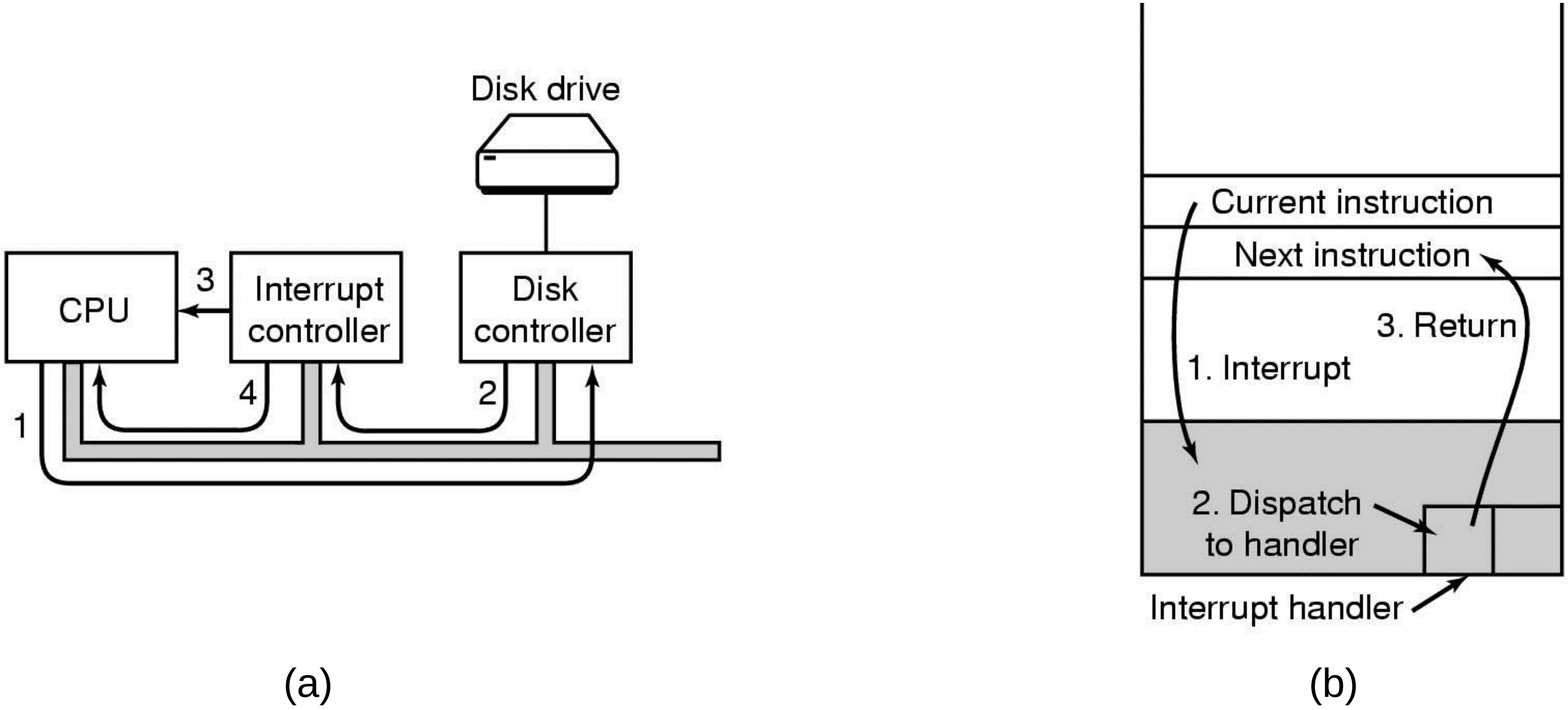
Structure of a disk drive

Computer Hardware Review (5)



One base-limit pair and two base-limit pairs

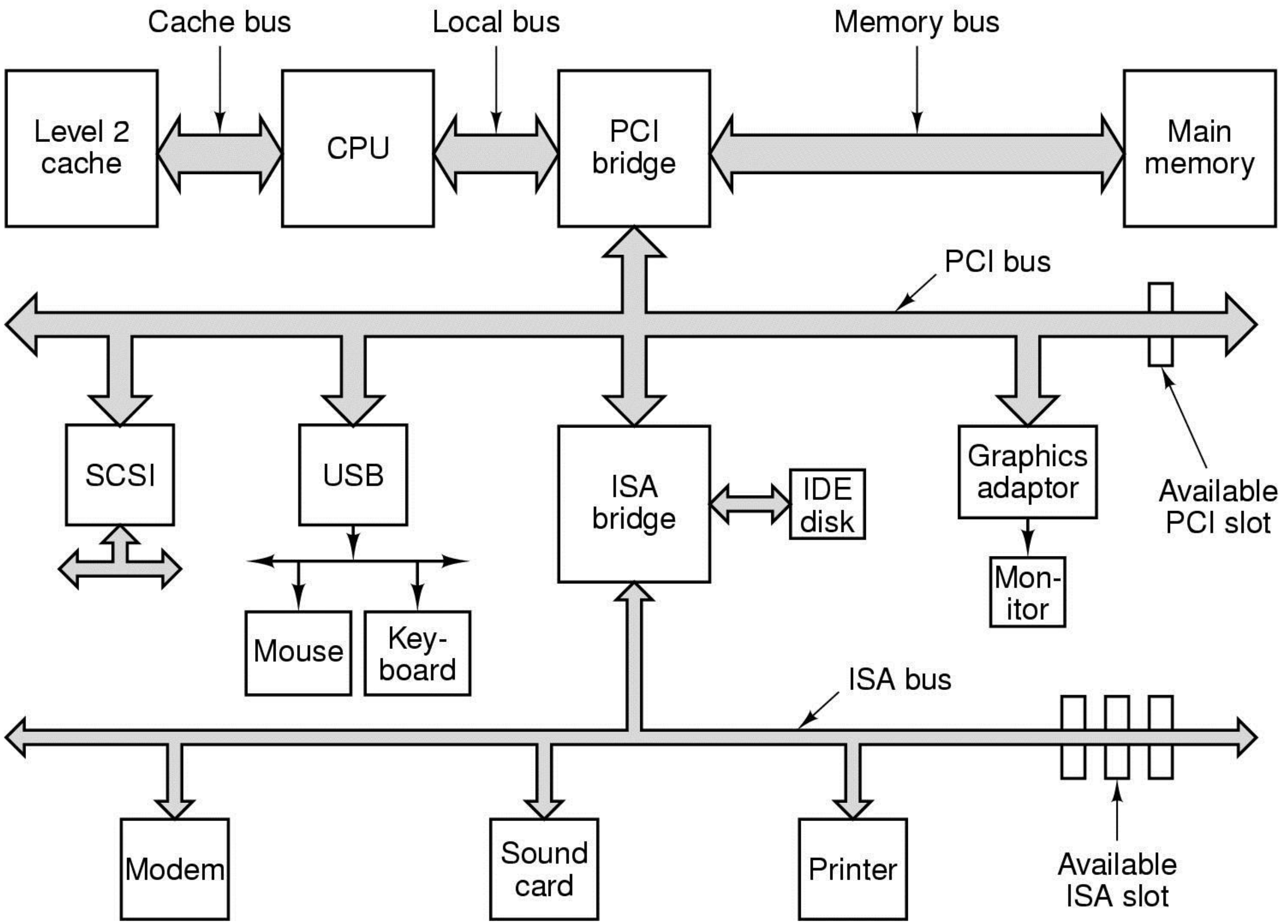
Computer Hardware Review (6)



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

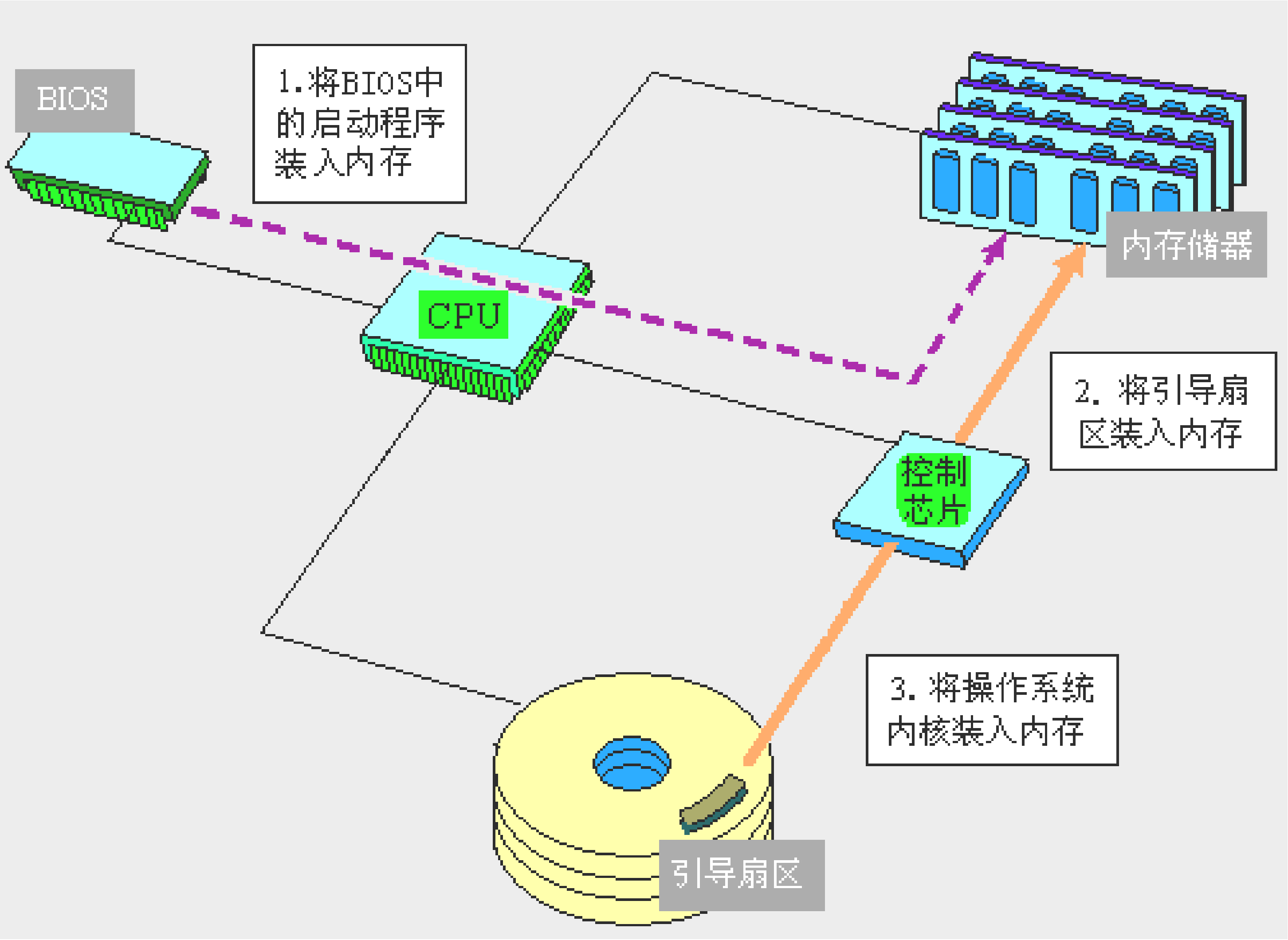
(b) How the CPU is interrupted

Computer Hardware Review (7)



Structure of a large Pentium system

Bootstrap :



1.将BIOS中的启动程序装入内存

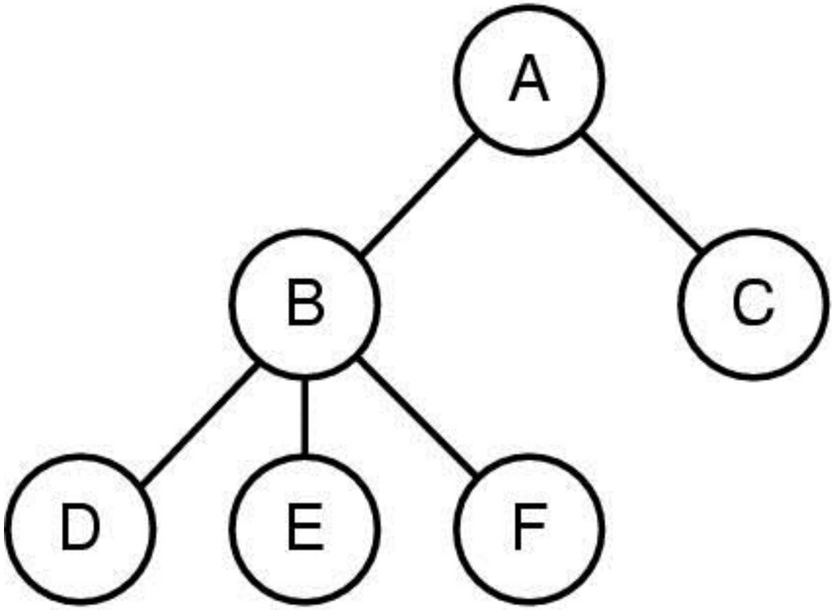
2. 将引导扇区装入内存

3. 将操作系统内核装入内存

OPERATING SYSTEM CONCEPTS

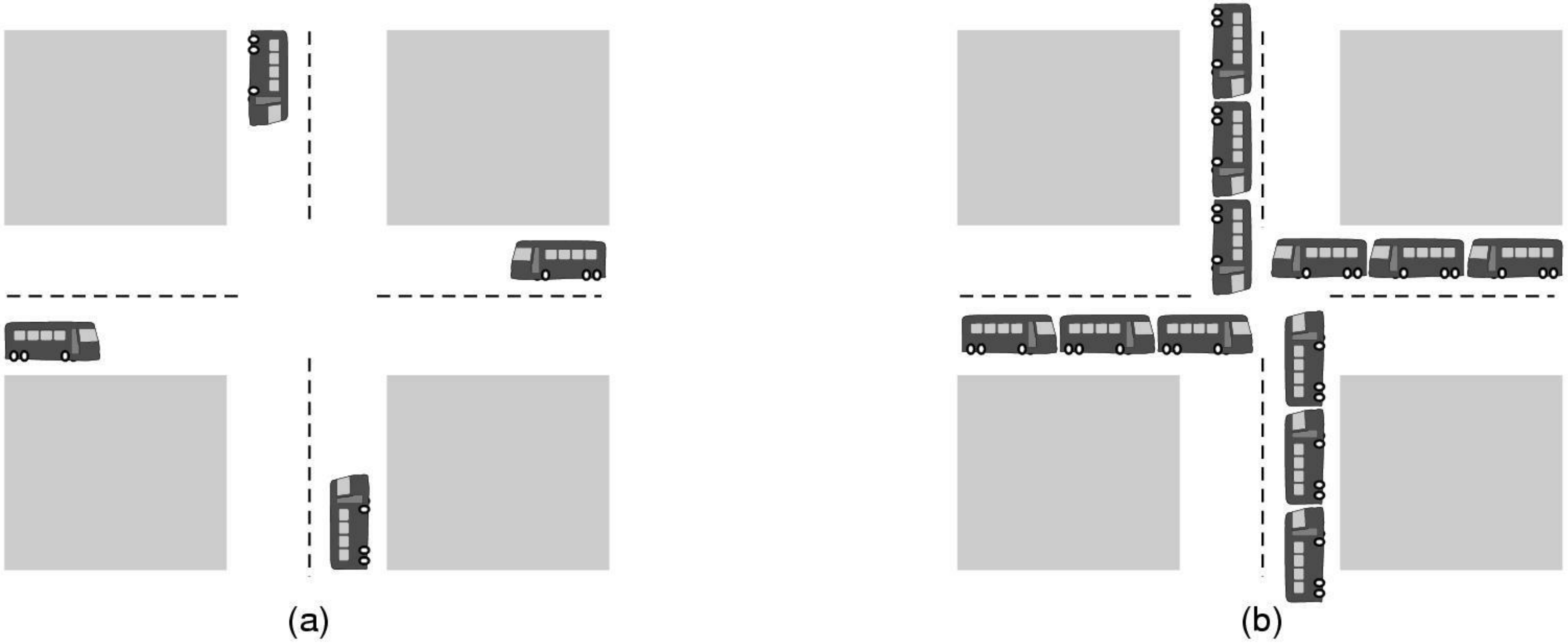
- Processes
- Address Spaces
- Input/Output
- Files
- Protection
- The Shell

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)



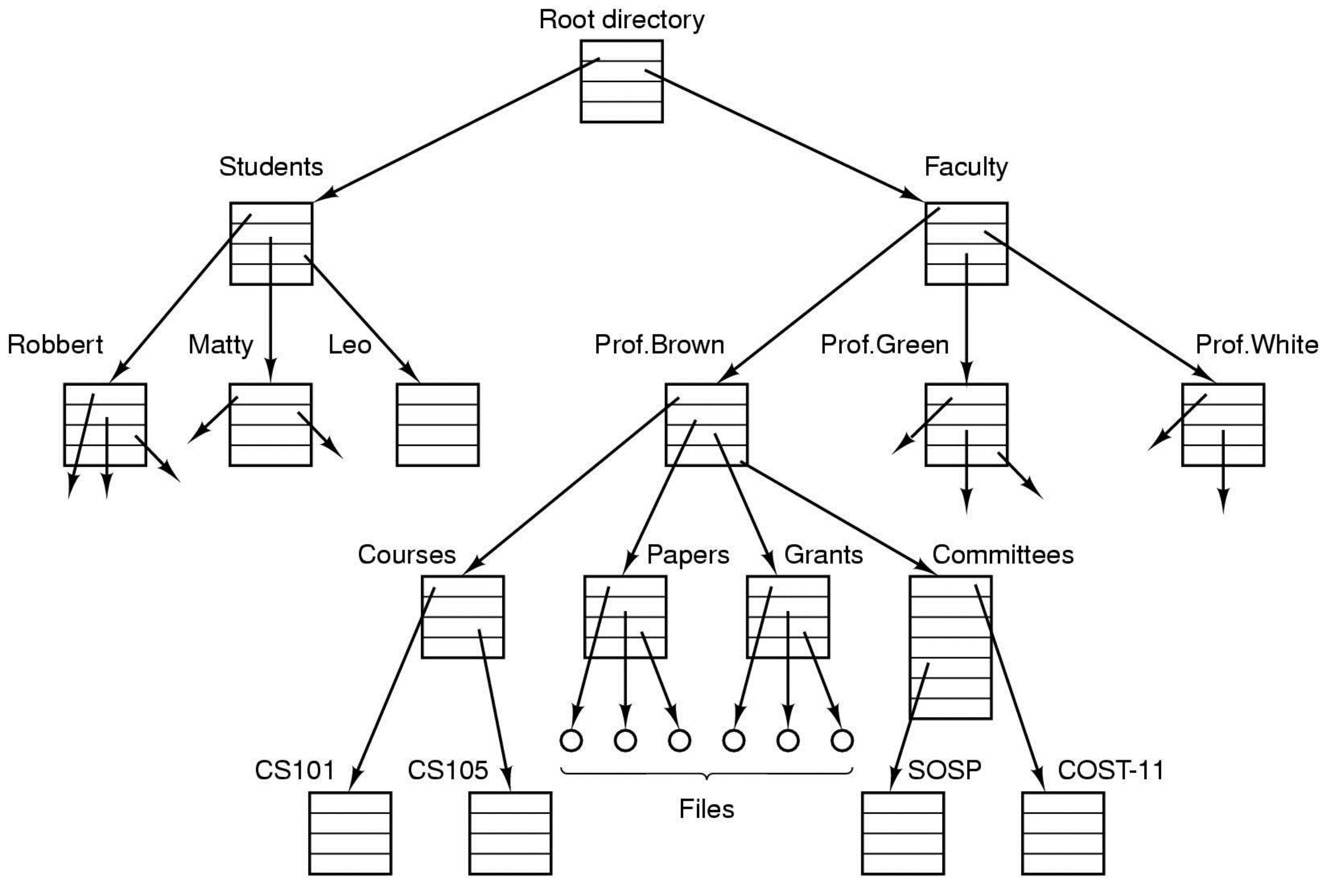
(a) A potential deadlock. (b) an actual deadlock.

OPERATING SYSTEM CONCEPTS
(MEMORY MANAGEMENT&INPUT/OUTPUT)

- Protection mechanism
- Virtual memory

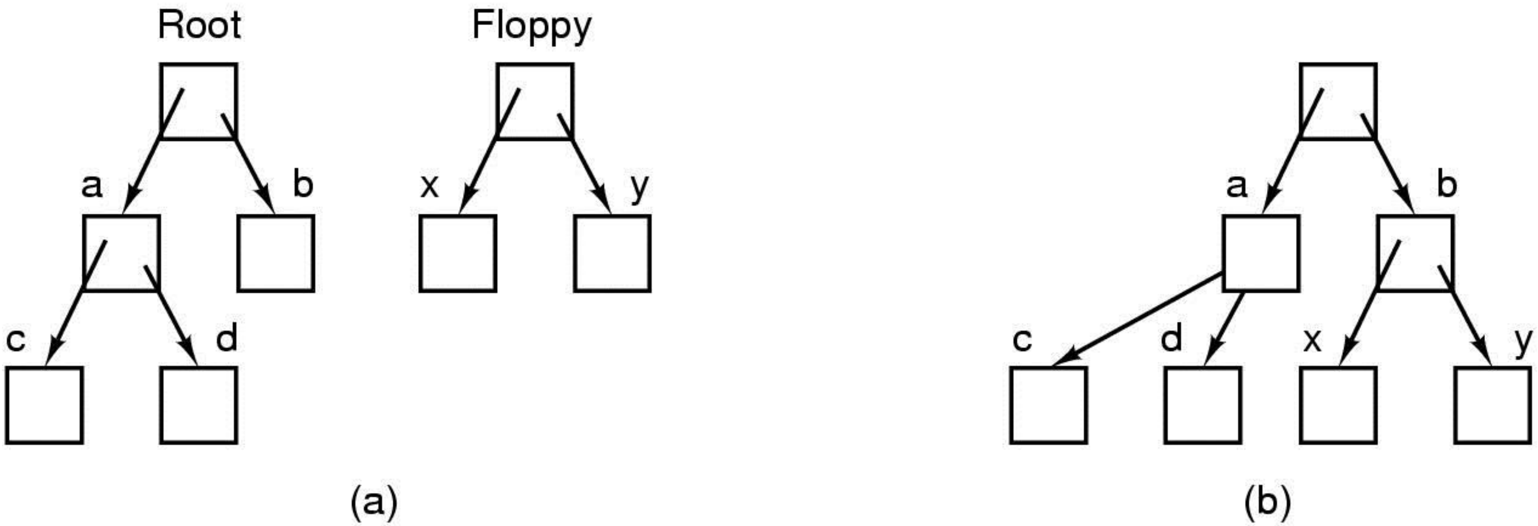
- I/O subsystem
- I/O software

Operating System Concepts (FILES1)



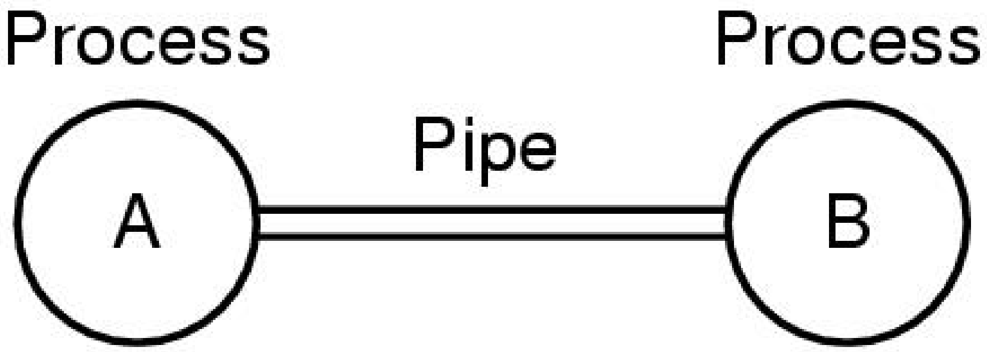
File system for a university department

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

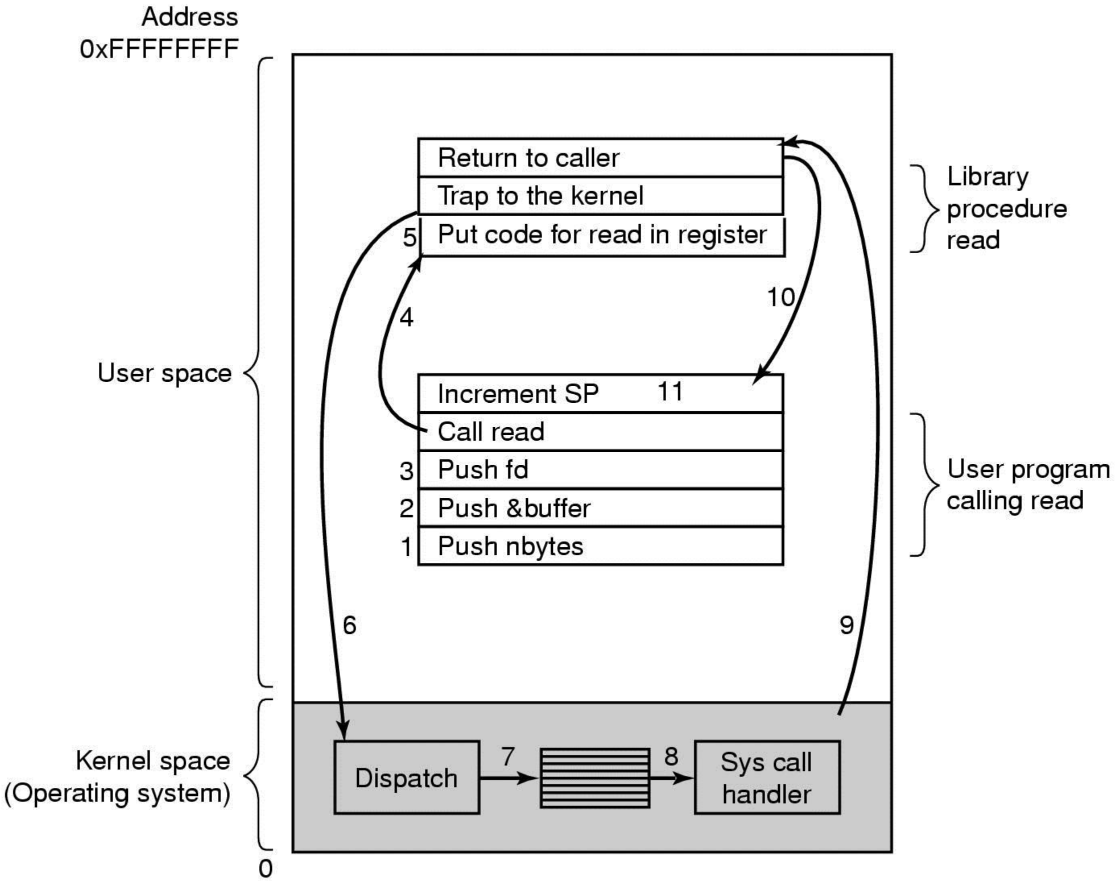
OPERATING SYSTEM CONCEPTS(SECURITY)

- Protection code

OPERATING SYSTEM CONCEPTS(THE SHELL)

- Command input & interpreters
- Human & machine interface

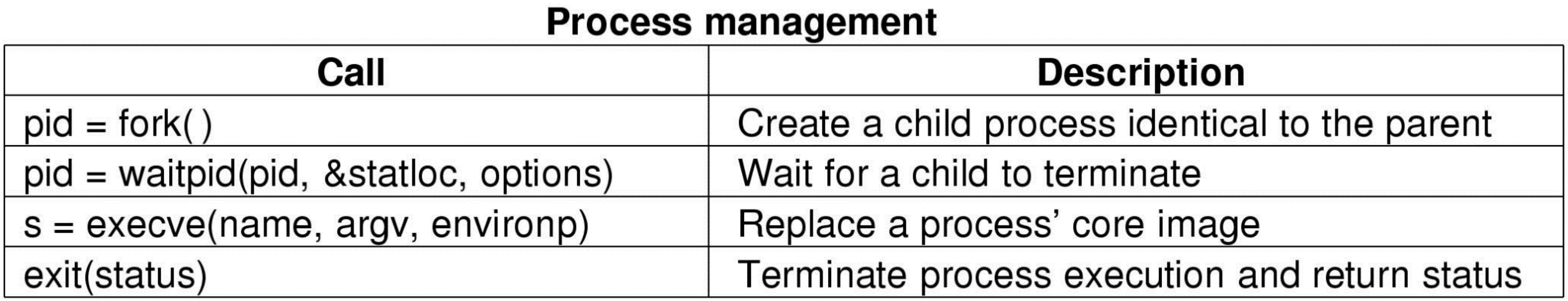
Steps in Making a System Call



There are 11 steps in making the system call

read (fd, buffer, nbytes)

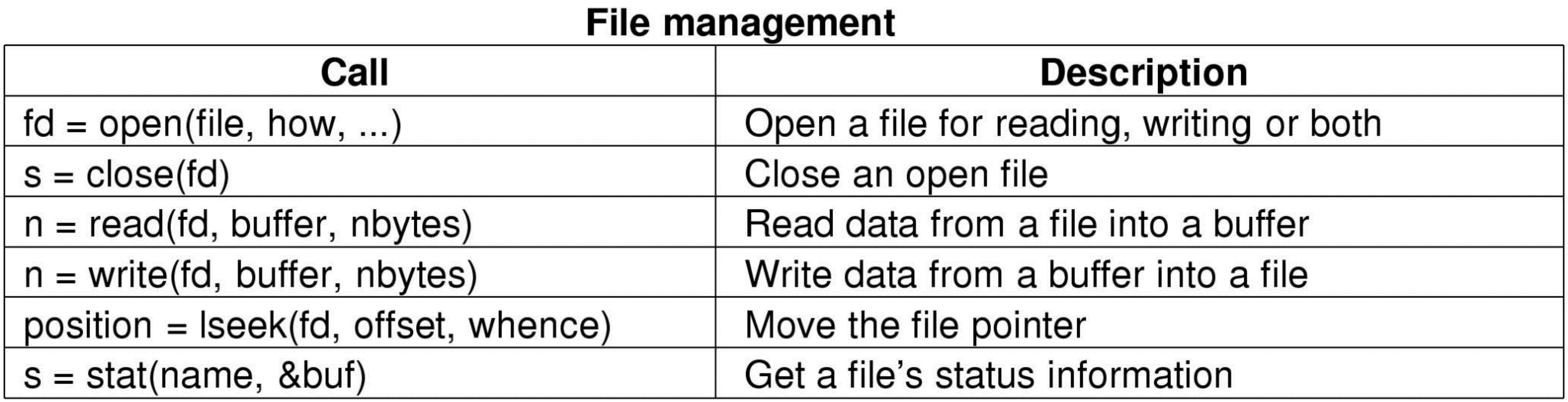
Some System Calls For Process Management



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

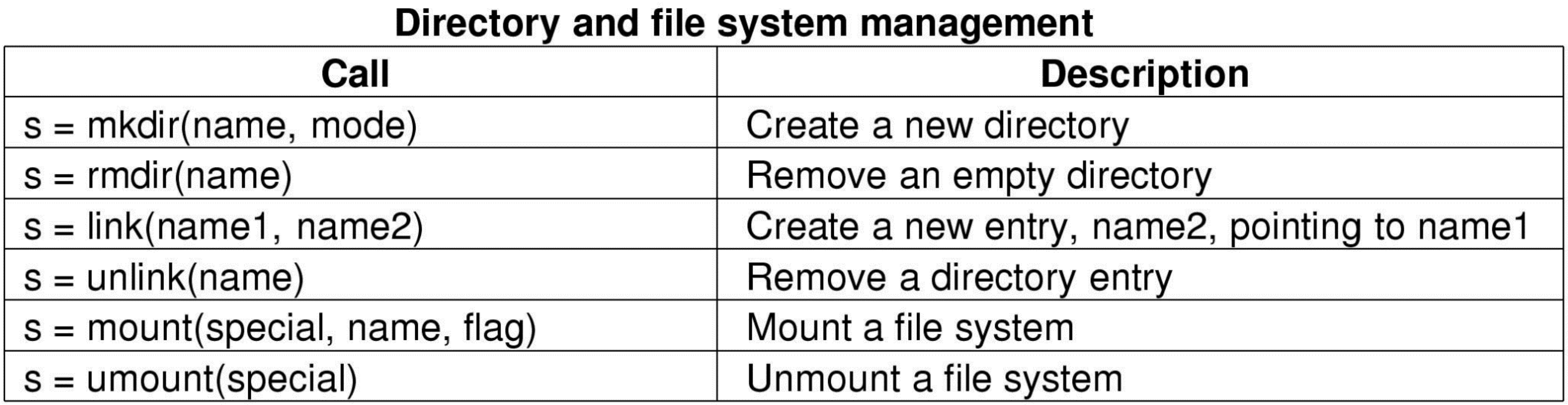
Some System Calls For File Management



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

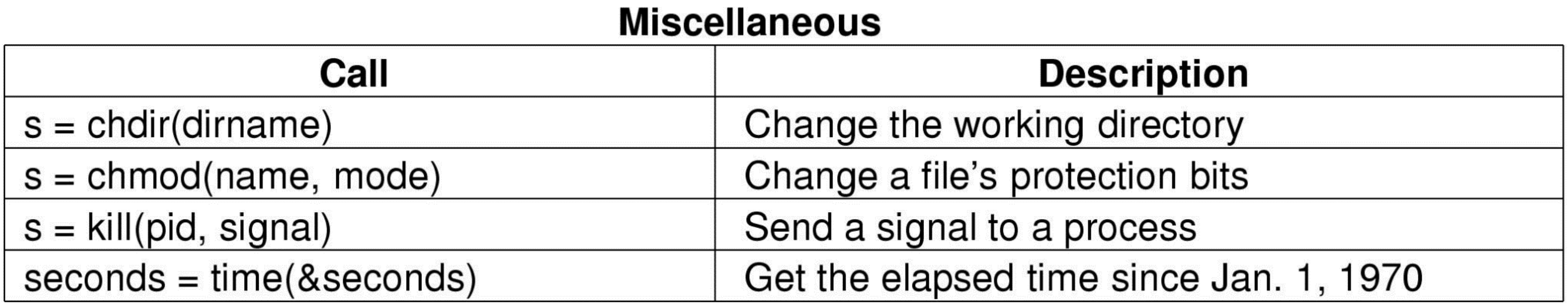
Some System Calls For Directory Management



Directory and file system management

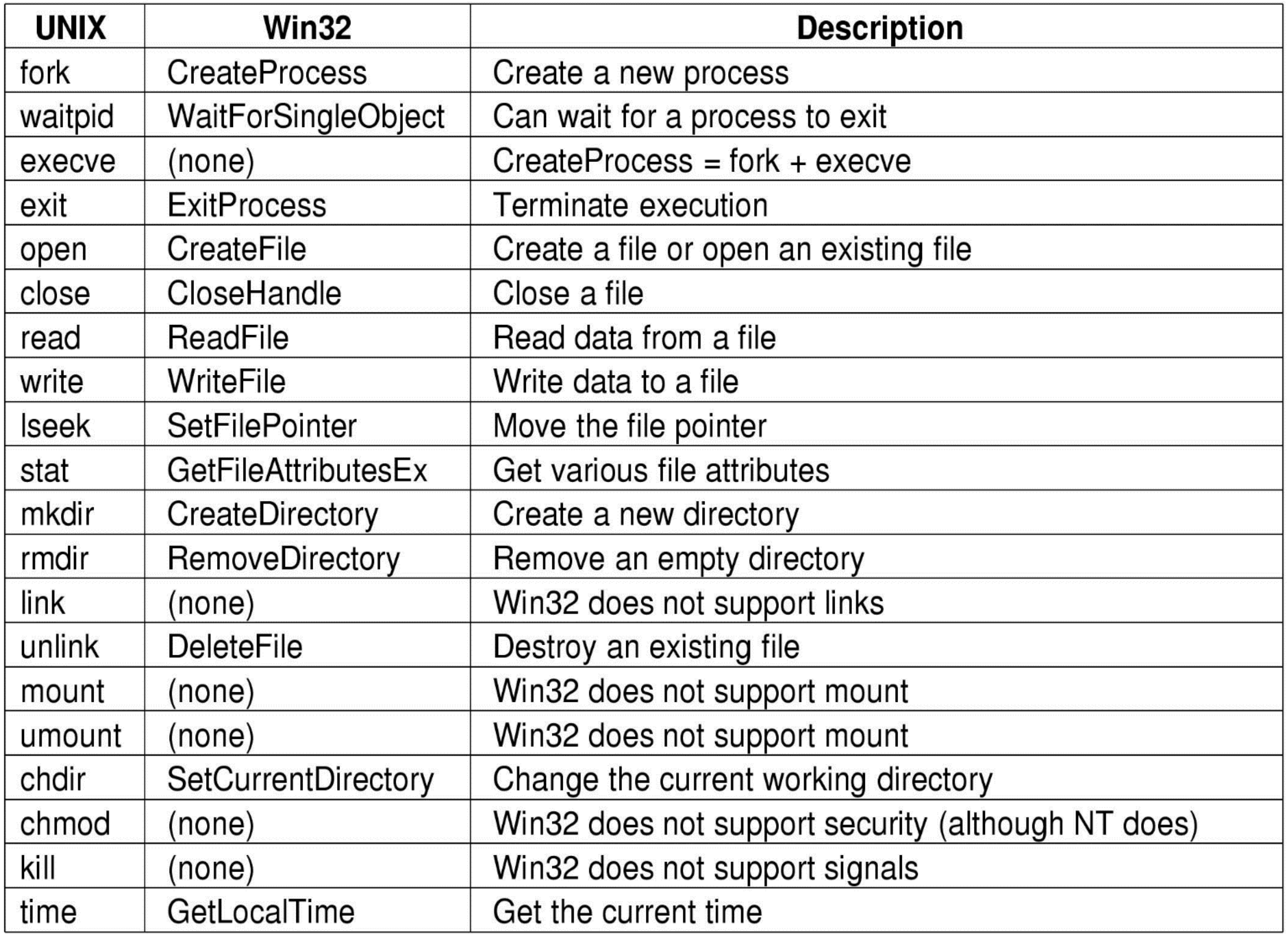
Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system

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



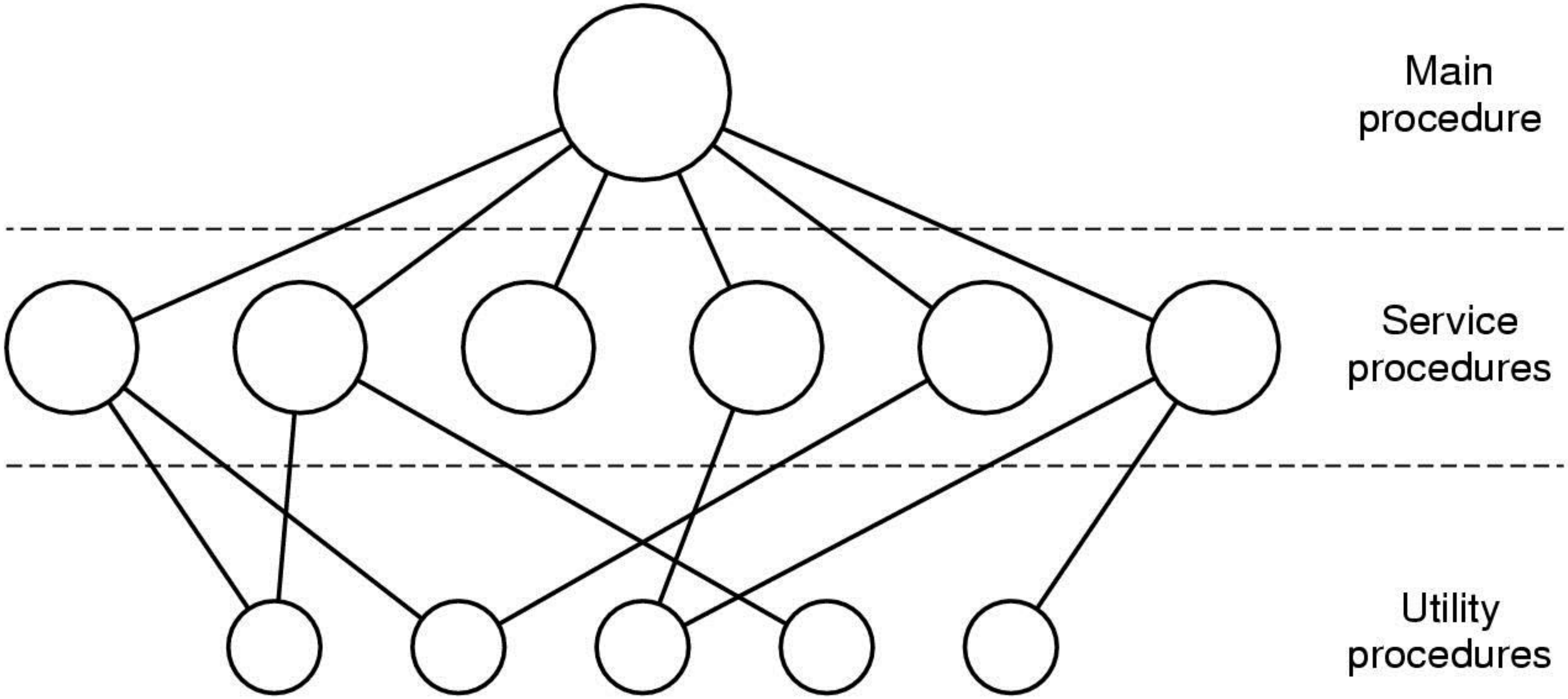
UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	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

OPERATING SYSTEM STRUCTURE

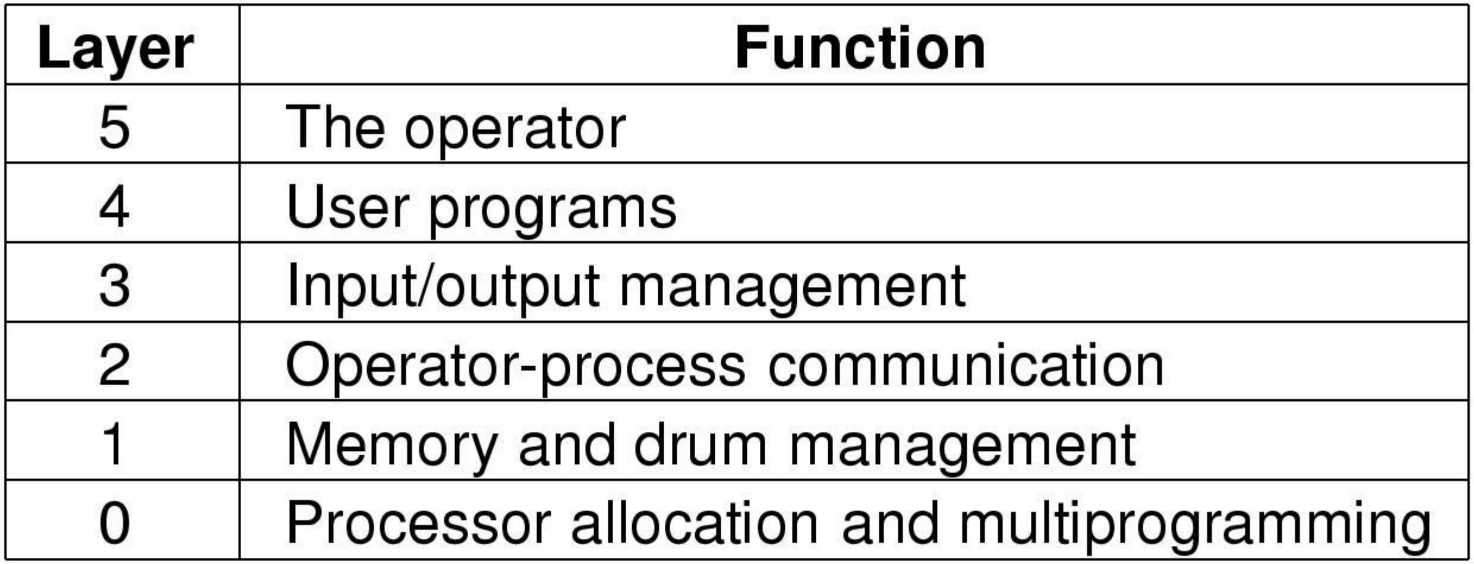
- Monolithic Systems
- Layered Systems
- Virtual Machines
- Exokernels
- Client-Server Model

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

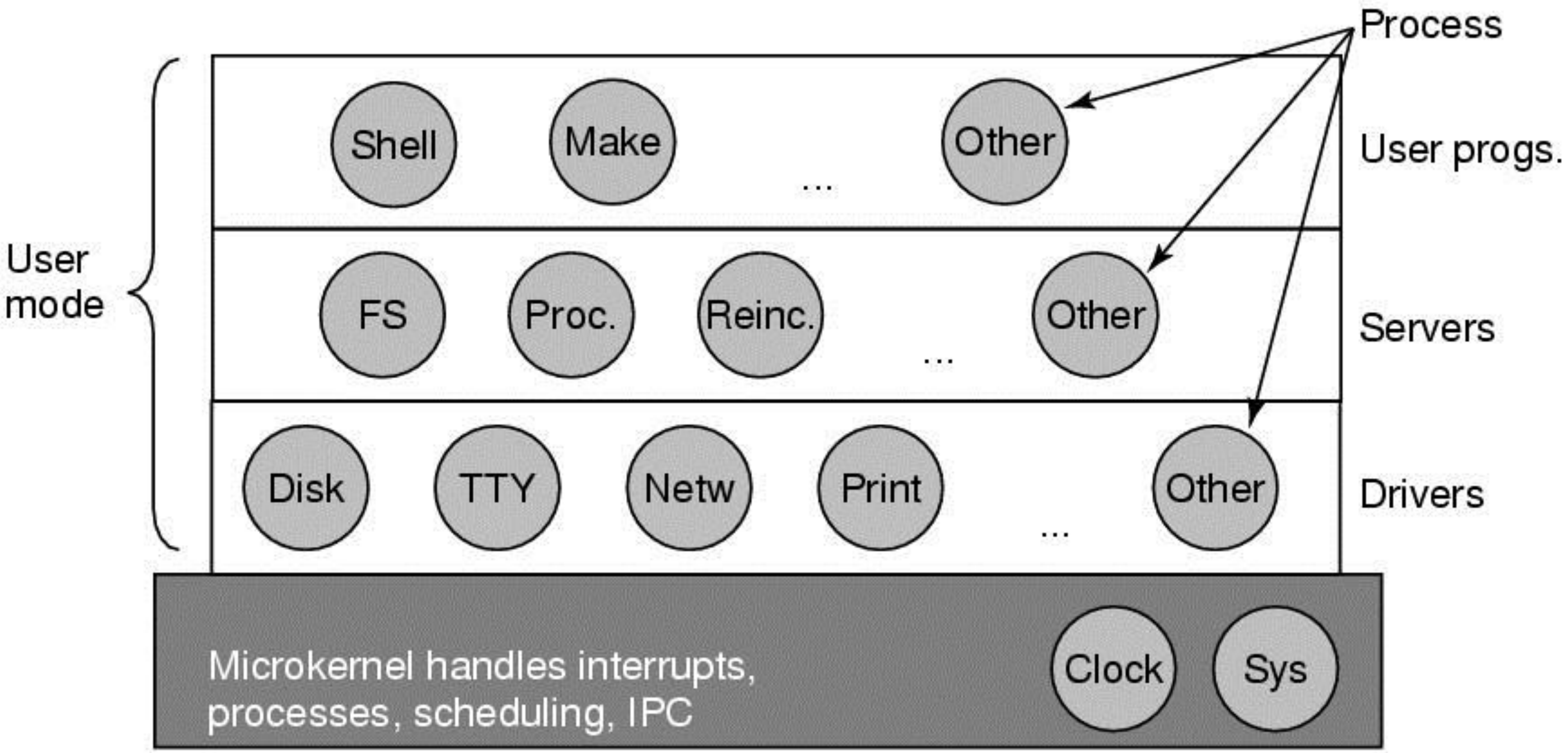
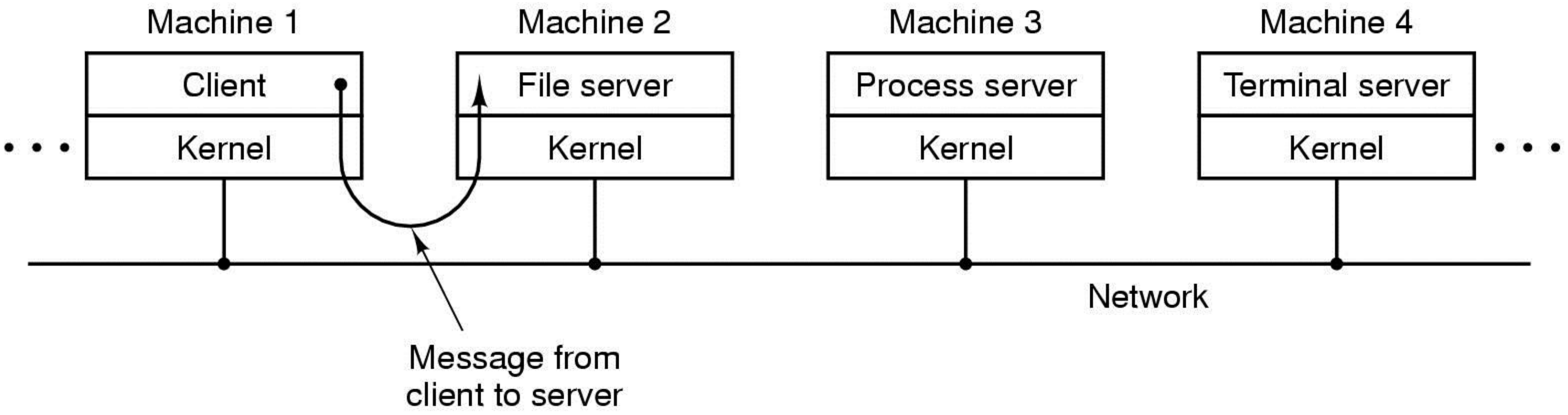


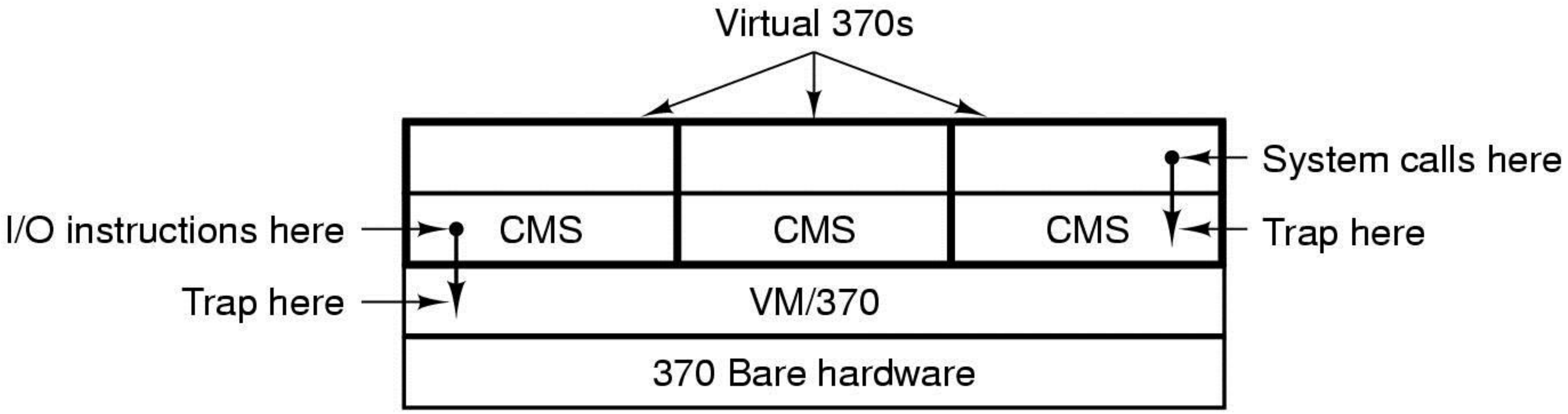
Figure 1-26. Structure of the MINIX 3 system.

Operating System Structure (4)



The client-server model in a distributed system

Operating System Structure (5)



Structure of VM/370 with CMS

The World According to C

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

The Model of Run Time

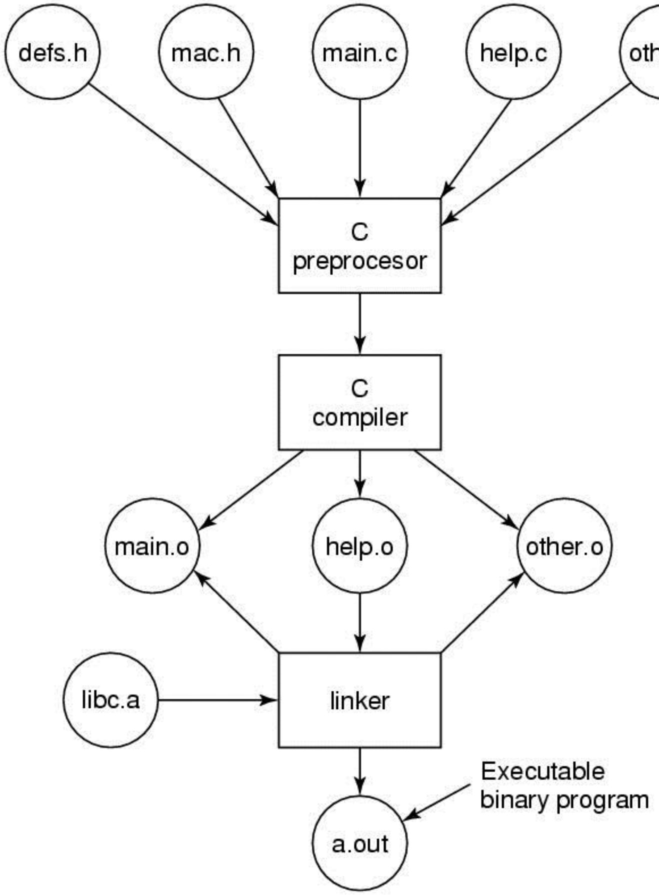
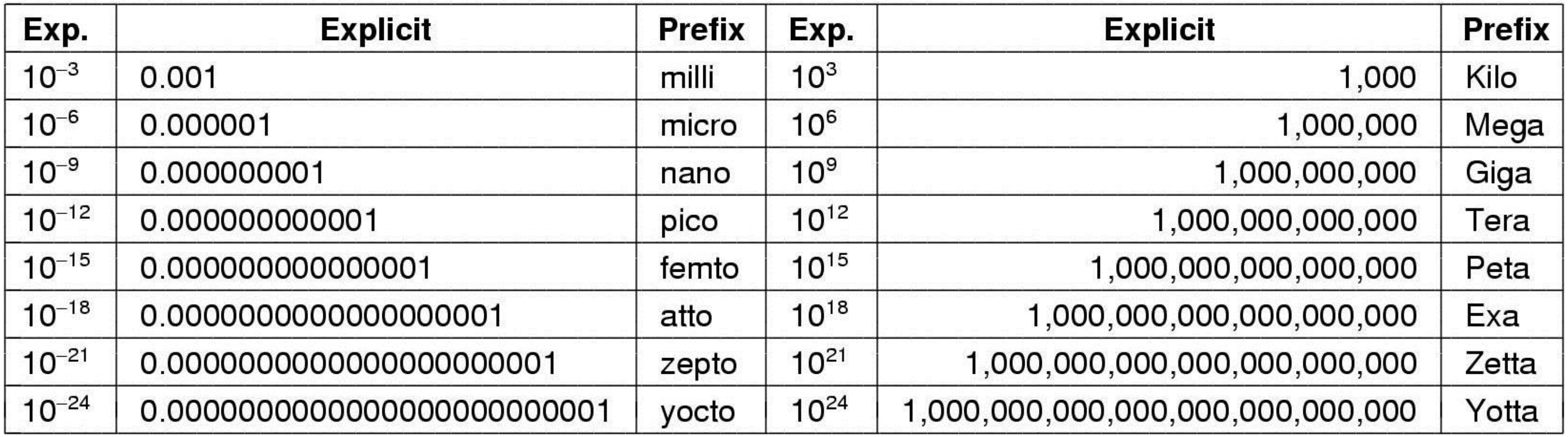


Figure 1-30. The process of compiling C and header files to make an executable.

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

小结

- OS的概念、地位作用
- OS的发展
- OS的基本类型

PROBLEMS 1 (HOMEWORKS1)

- 课后练习：1，3，12，17，28

要求：一周内提交至师星学堂，
文件命名：学号姓名Homework1.doc

* 参与讨论区话题