Operating Systems

Lecture 1 Introduction

JinpengChen
Email: jpchen@bupt.edu.cn

Outline

- Computer System Overview
- What is an Operating System?
- History of Operating Systems
- The Operating System Zoo
- Supplement Knowledge

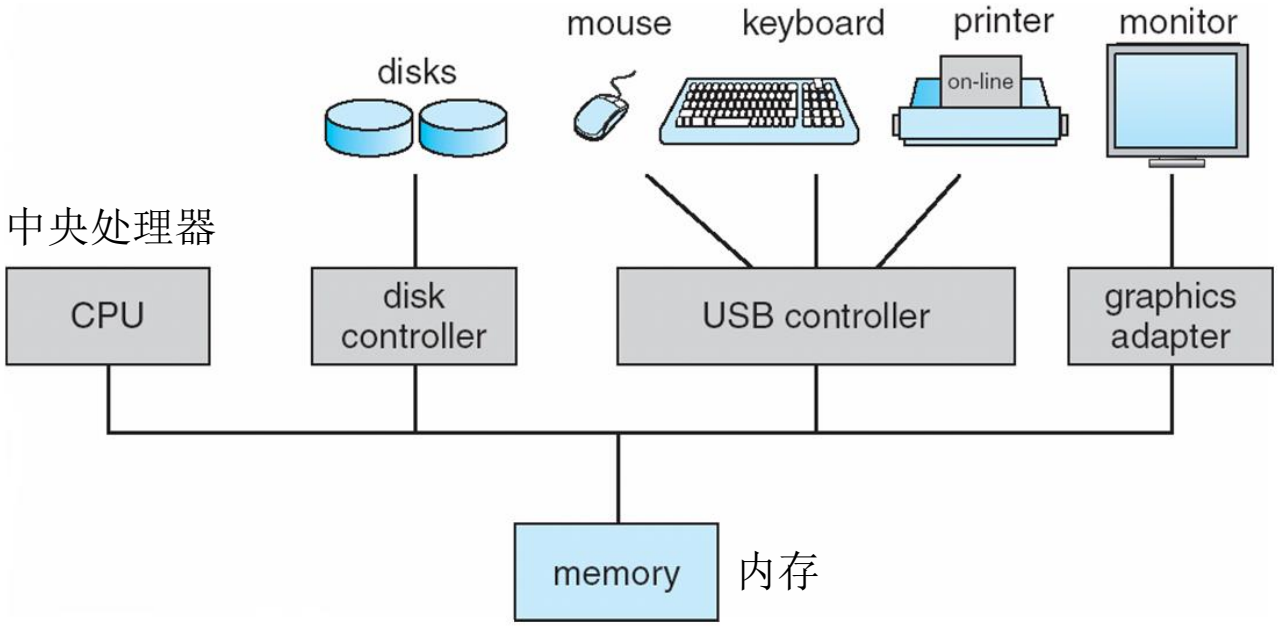
Computer System overview

- **Two questions?**
  - In a computer system，what kinds of hardware devices are there？
  - How to put the hardware devices into a system？What structure？

Computer System Organization

Computer-system operation

- One or more CPUs, device controllers connect through common bus providing access to shared memory.
- Concurrent execution of CPUs and devices competing for memory cycles.



Computer System Organization

- **Computer-system operation**
  - Each device controller is in charge of a particular device style
  - Each device controller has a local buffer
  - CPU moves data from/to main memory to/from local buffer
  - I/O is from the device to local buffer of controller
  - Device controller informs CPU that it has finished its operation by causing an interrupt

Common functions of interrupt

- Interrupt transfers control to interrupt service routine through interrupt vector， which contains addresses of all the service routine
- Interrupt architecture must save the address of the interrupt instruction
- Incoming interrupt are disabled while another interrupt is being processed to prevent a lost interrupt
- Trap：a trap is a software-generated interrupt caused either by an error or a user request
- An operating system is interrupt driven

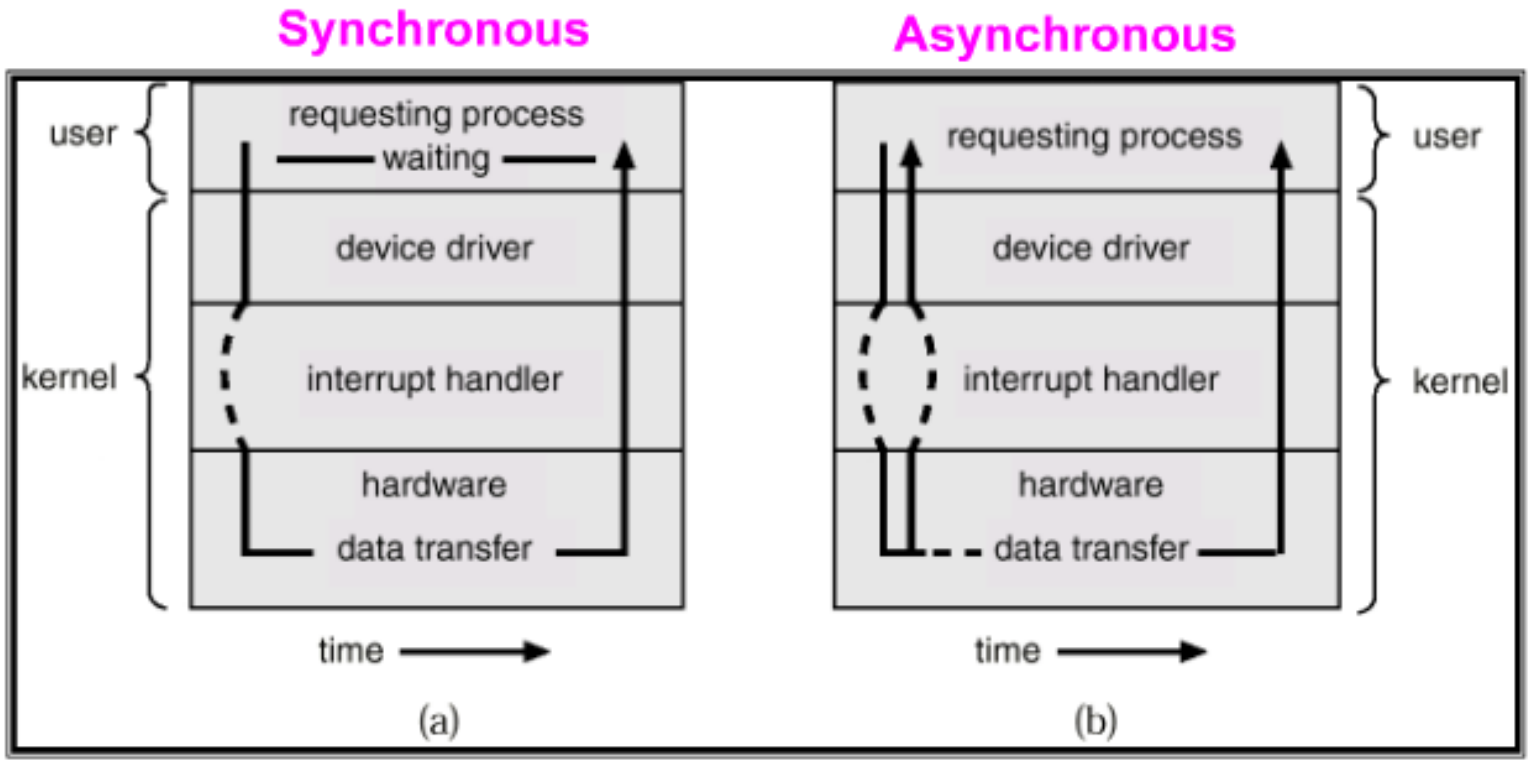
Interrupt handling

- OS preserves the state of the CPU by storing register and program counter
- Determines which type of interrupt has occurred:
  - Polling
  - Vectored interrupt system
- Separate segments of code determine what action should be taken for each type of interrupt

I/O structure

- After I/O starts, control returns to program ,two ways
  - Synchronous: Only upon I/O completion
    - ✓ Wait instruction idles the CPU until the next interrupt
    - ✓ Wait loop(contention for memory access)
  - Asynchronous: without waiting for I/O completion
    - ✓ System call --request to the operating system to allow user to wait for I/O completion
    - ✓ Device-status table--contains entry for each I/O device indicating its type, address, and state.
    - ✓ Operating system indexes into I/O device table to determine device status and to modify table entry to include interrupt.

Two I/O methods



Direct memory access

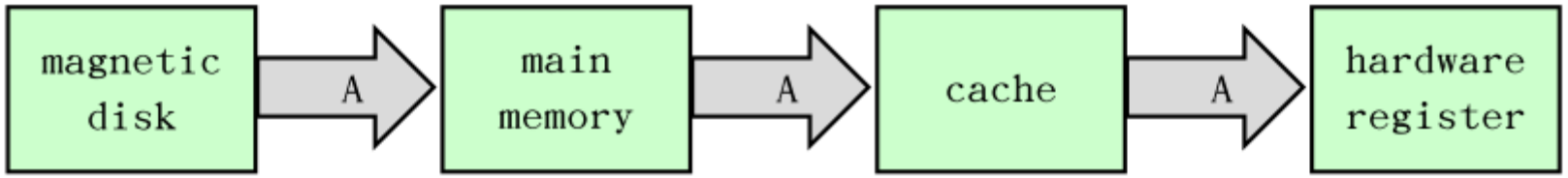
- Used for high-speed I/O devices able to transmit information at close to memory speeds
- Device controller transfer blocks of data from buffer storage directly to main memory without CPU intervention
- Only one interrupt is generated per block ,rather than the one interrupt per type.

Caching(高速缓冲)

- Performed at many levels in a computer (in hardware, operating system, software)
- Information in use copied from slower to faster storage temporarily
- When accessing, first check in the cache,
  - if In: use it directly
  - Not in: get from upper storage system, and leave a copy in the cache

Coherency and consistency

- Coherency Information in use copied from slower to faster storage temporarily
- Migration of Integer A from Disk to Register



- The same data may appear in different level of the storage system
- Multiprocessor environment must provide **cache coherency** in hardware such that all CPUs have the most recently updated value

Performance of Various Levels of Storage

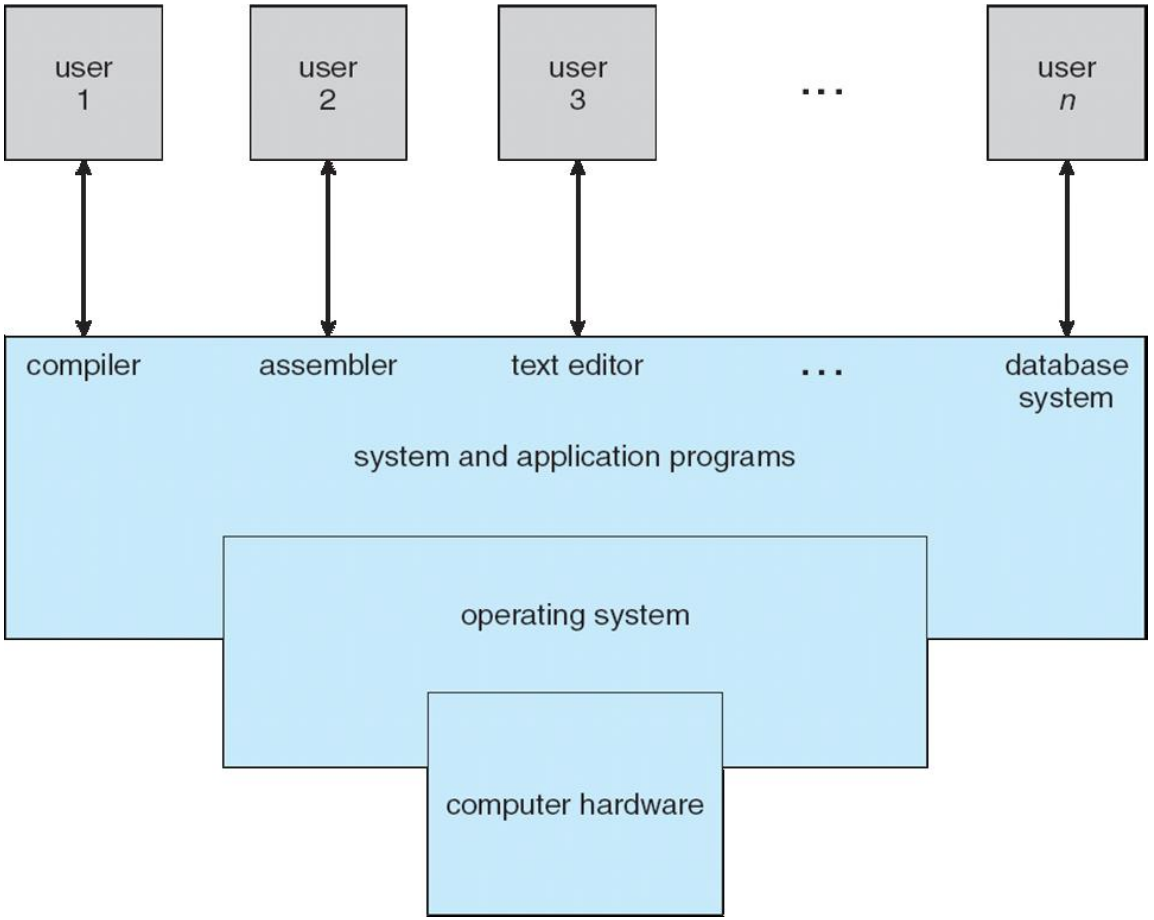
- Movement between levels of storage hierarchy can be explicit or implicit

Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	<1KB	>16MB	>16GB	>100GB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25—0.5	0.5—25	80—250	5,000.000
Bandwidth (MB/sec)	20,000—100,000	5000—10,000	1000—5000	20—150
Managed by	compiler	hardware	OS	OS
Backed by	cache	main memory	disk	CD or tape

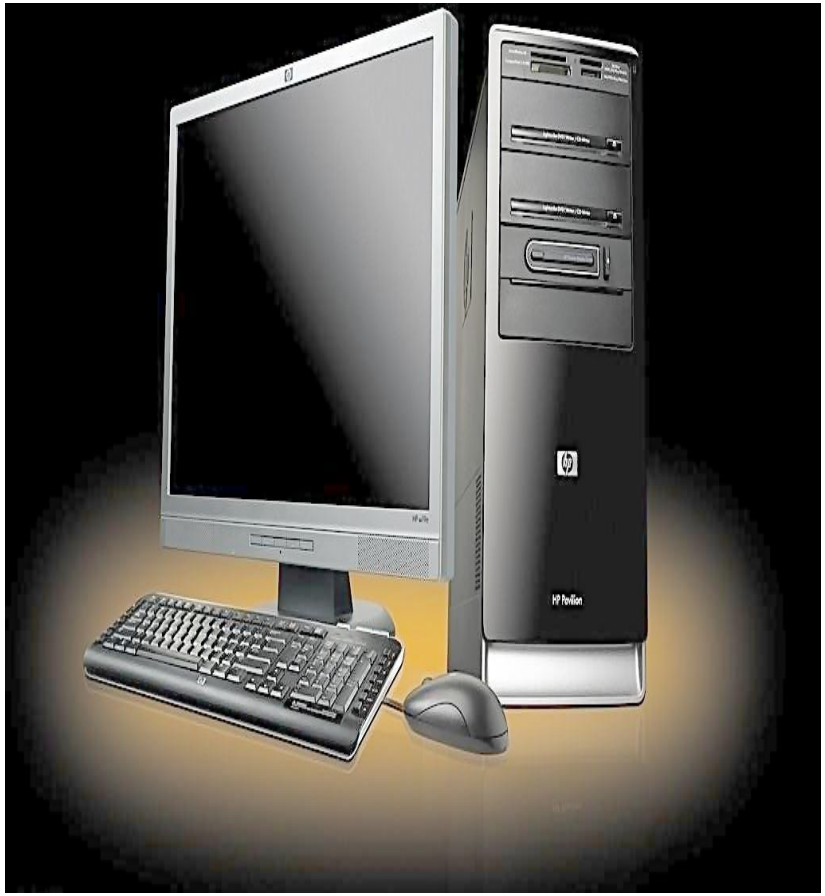
Computer System Components

- **Computer system can be divided into four components:**
  - Hardware – provides basic computing resources (CPU, memory, I/O devices).
  - Operating system – controls and coordinates the use of the hardware among the various application programs for the various users.
  - Application programs– define the ways in which the system resources are used to solve the computing problems of the users (Word processors, compilers, web browsers, database systems, video games).
  - Users (people, machines, other computers)

Four Components of a Computer System



Computer System overview



Computer instruction

- **Computer instruction**
  - 指令是计算机运行的最小的功能单元，是指挥计算机硬件运行的命令
  - 算数运算指令、逻辑运算指令、移位操作指令、数据传送指令、输入输出指令、转移指令等

- **Instruction format**

操作码	操作数1	操作数2	……

  - Example:

    模仿 小鸡

    加 1 2

Outline

- Computer System overview
- What is an Operating System?
- History of Operating Systems
- The Operating System Zoo
- Supplement Knowledge

What is an Operating System?

- **Why operating system?**
  - 如果程序员直接对硬件编程……
- 编写一个程序
  - Input：a name, such as Wukong
  - Output：Hello, Wukong!

What is an Operating System?

- **如果直接对硬件编程……**
  - 键盘、显示器显示的工作原理？
  - 如何把缓冲区的数据拷贝到内存？
  - 如何在屏幕的某个特定位置显示某个字符？

  ……
- **怎么办？**
  - Adds an intermediary between the computer user and the computer hardware：
    - ✓ Manages the computer hardware
    - ✓ Provides a basis for application programs
  - This intermediary is operating system

What is an Operating System?

- There are a lot of definitions.
- Do you know?

What is an Operating System?

- 不同人眼中的OS是不同的
  - OS设计者：如何管理CPU、内存、I/O设备等系统部件，使之能正常运转
  - 应用程序开发人员：API应用程序编程接口

    int MessageBox（...）；

    HWND CreatWindow（...）；

    int DrawText（...）；

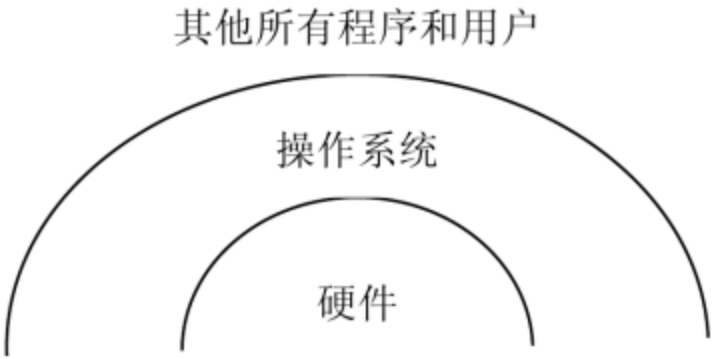
    BOOL PlaySound（...）；

    HDC BeginPaint（...）；

    ……

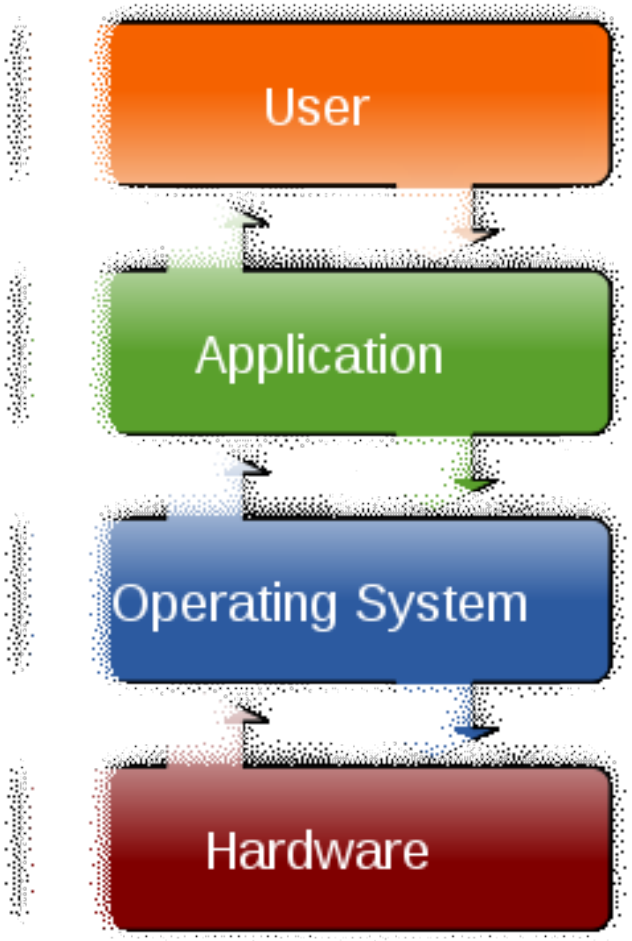
What is an Operating System?

- Computer system = hardware + software (+data)
  - Operating System(操作系统, OS) is the first software layer on the physical hardware, and can be viewed as the first expansion of computer hardware system.
  - All applications running in the OSes, more or less, directly or indirectly, call the OS functions. For example, the simplest program “helloworld”.



What is an Operating System?

- **Definition from Wikipedia**
  - An operating system (OS) is an interface between hardware and applications; it is responsible for the management and coordination of activities and the sharing of the limited resources of the computer.



What is an Operating System?

- 没有一个统一的、适用的定义！
- An Operating System is a program that
  - Manages the computer hardware
  - Provides a basis for application programs
  - Acts as an intermediary between the computer user and the computer hardware
- OS is a resource allocator that
  - Manages and allocates all resources （管理和分配资源）
  - Decides between conflicting requests for efficient and fair resource use
- OS is a control program that
  - Controls execution of programs to prevent errors and improper use of the computer
  - controls operations of I/O devices

What is an Operating System?

- Kernel – the one program running at all times (all else being application programs). -- 全时运行的一个程序

Our Traditional Definition

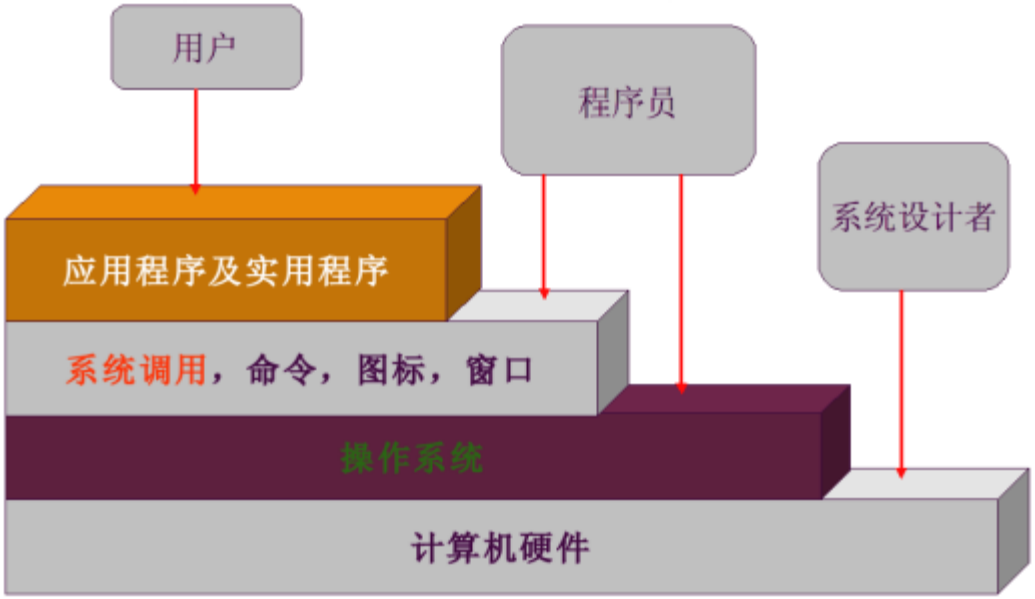
- A set of programs that control and manage computer hardware and software, and organize computer's workflow so as to make the computer system convenient for users to use reasonably and efficiently.
- 控制和管理计算机软硬件资源，合理组织计算机工作流程，方便用户合理使用计算机的程序的集合。
- 操作系统是一组控制和管理计算机软硬件资源、合理地对各类作业进行调度以及方便用户的程序的集合

Operating System Goals

- Operating system purpose: Provide an environment in which a user can execute programs.
- 在计算机硬件上配置OS的（设计）目标有以下几点
  - convenience(方便性)
    - ✓ Execute user programs and make solving user problems easier
    - ✓ Make the computer system convenient to use
  - Effectiveness(有效性)
    - ✓ Use the computer hardware in an efficient manner （提高软硬件资源的利用率）
  - Extensibility(可扩充性)
    - ✓ 适应软硬件的发展需求
  - openness(开放性)
    - ✓ 可移植性、互操作性
- 方便性和有效性是操作系统最重要的两个目标。

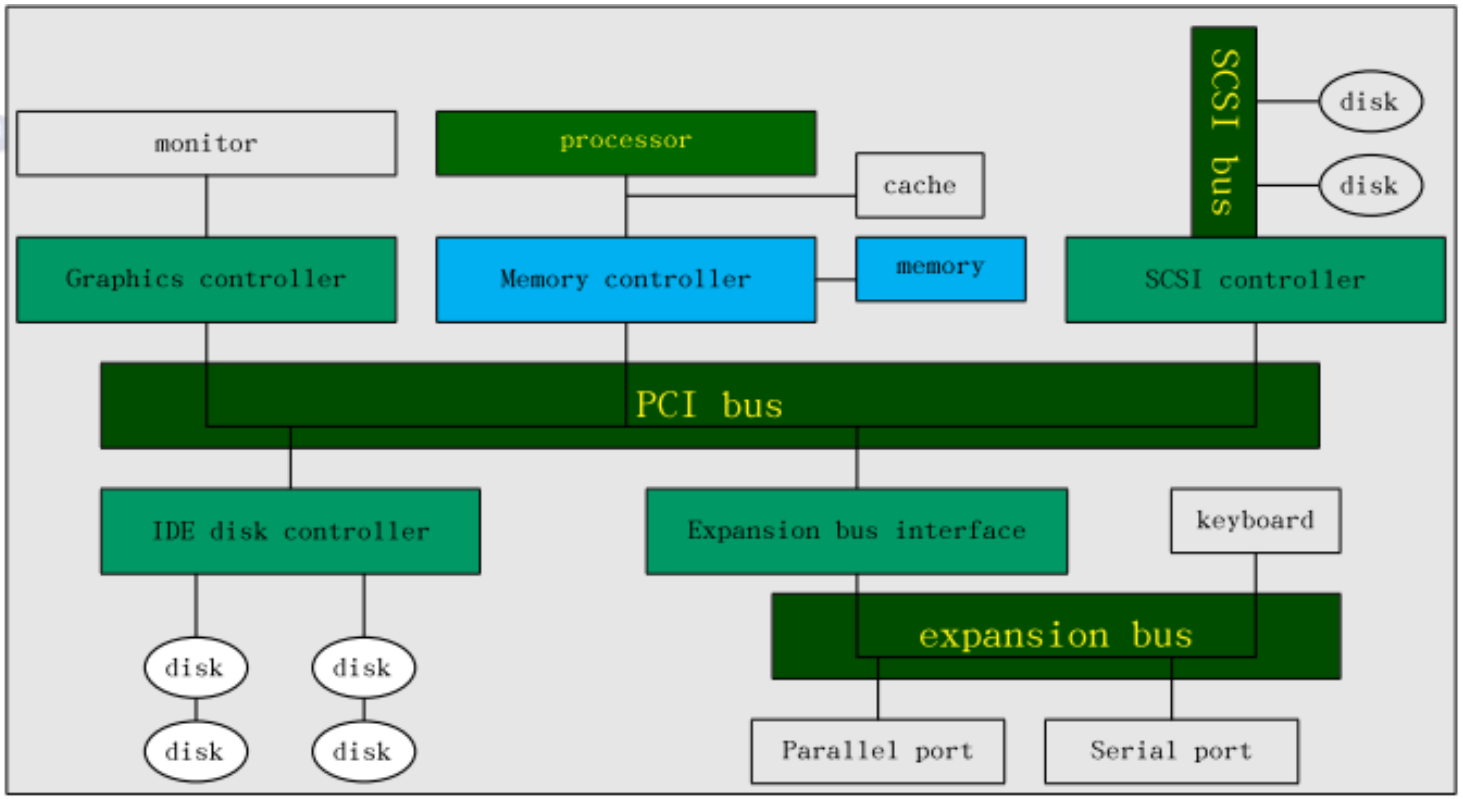
Roles of operating system

- 对操作系统作用的理解，有不同的观点
  - 用户与计算机硬件系统之间的接口(interface)
    - ✓ 命令接口(Command Line Interface, CLI)、
    - ✓ 图形用户接口(Graphical User Interface, GUI)
    - ✓ 编程接口（系统调用接口(system call)）



Roles of operating system

- 对操作系统作用的理解，有不同的观点
  - 计算机资源的管理者(resource allocator)
    - ✓ 四类资源：处理机、存储器、I/O设备、文件



Roles of operating system

- 对操作系统作用的理解，有不同的观点
  - 扩充机器（或虚拟机Virtual Machine）
    - ✓ 虚拟机：覆盖了软件的机器
    - ✓ 层次性

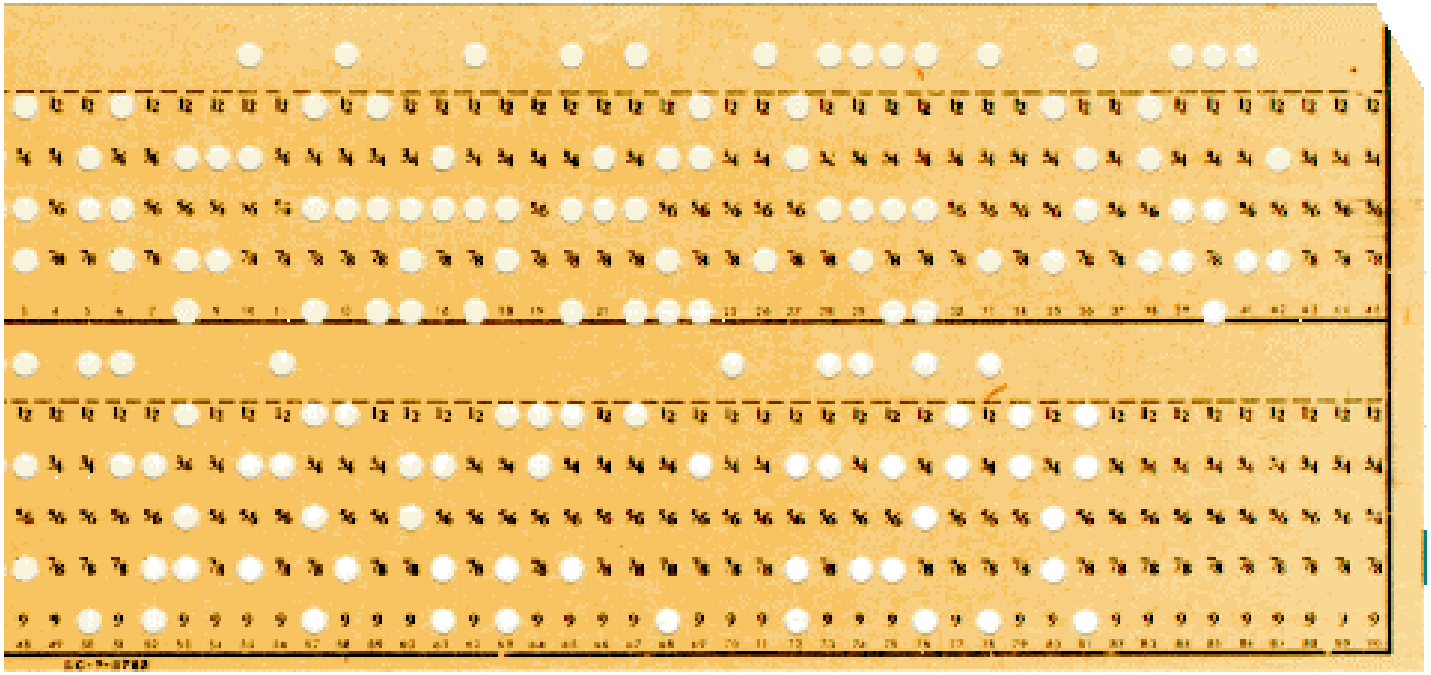
Outline

- Computer System overview
- What is an Operating System?
- History of Operating Systems
- The Operating System Zoo
- Supplement Knowledge

History of Operating Systems

- Phase 1: 1945-1955 （硬件昂贵，无操作系统）
- OS: no OS at all ! (programs were entered by setting some switches) ---Manual system （人工操作）
- Computer designer is: builder, programmer, operator, and “sys admin”
- A physicist who wanted to calculate the trajectory of a missile would sign up for an hour on the computer in advance.
  - ✓ When his time came, he would come into the room, feed in his program from punched cards or paper tape, watch the lights flash, maybe do a little debugging, get a print-out, and leave.

Punched Card



History of Operating Systems

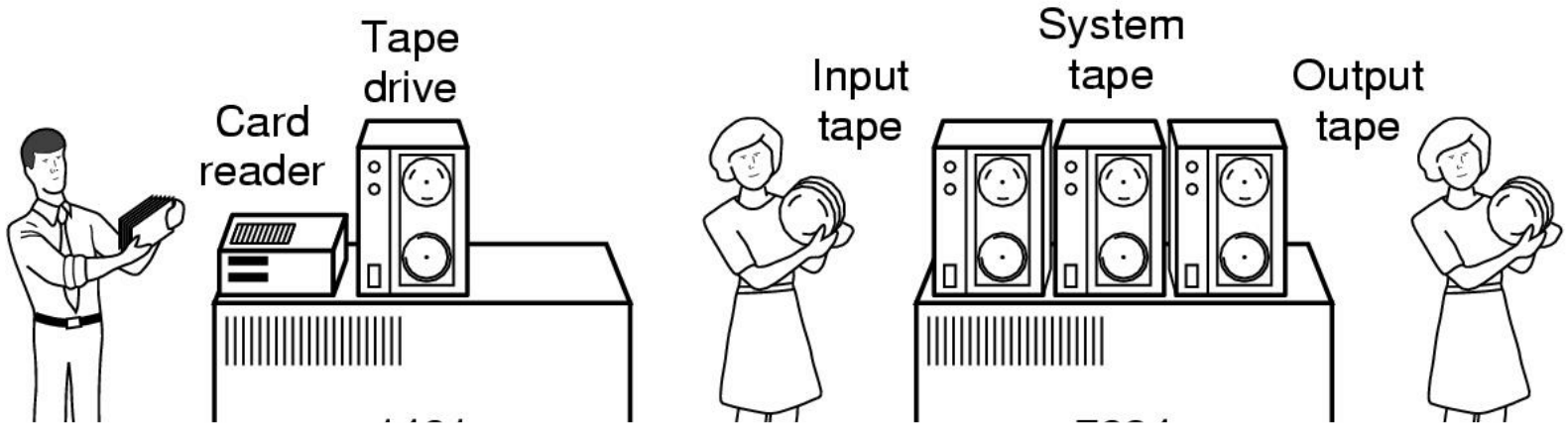
- Phase 2: 1955-1965（硬件昂贵，人力便宜）
- Universities started to buy computers (spending millions of dollars)
- OS: batch system
- Programming language: FORTRAN & Assembly
- Computer: Mainframes
- Low CPU utilization -- slow mechanical I/O devices

IBM
STRETCH

DEC
PDP-1

Early Batch System

- Phase 2: 1955-1965（硬件昂贵，人力便宜）



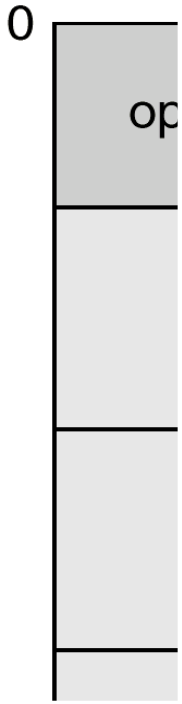
- bring cards to 1401
- read cards to tape
- put tape on 7094 which does computing
- put tape on 1401 which prints output

Batch Systems 作业=程序+数据+处理步骤

- Phase 2: 1955-1965（硬件昂贵，人力便宜）
- Hire an operator 设置一个操作员
- User ≠operator
- Reduce setup time by batching similar jobs
  批量处理同类作业减少了设置时间
- OS's main task: Automatic job sequencing – automatically transfers control from one job to another.
- Resident monitor
- ✓ initial control in monitor 初始化管理程序
- ✓ control transfers to job 转换控制到作业

History of Operating Systems

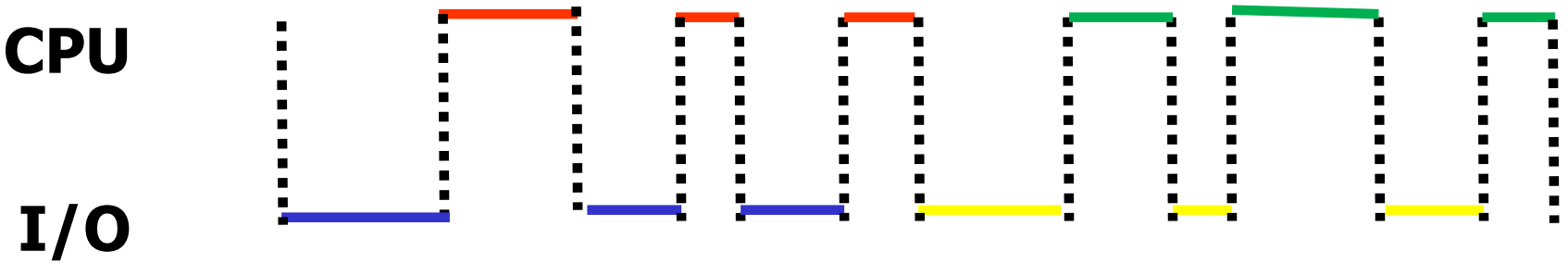
- Phase 2: 1965-1980（Multiprogrammed Batch Systems）
  - Several jobs are kept in main memory at the same time, and the CPU is multiplexed among them.
  - Multiprogramming increases CPU utilization by organizing jobs such that CPU always has one to execute.



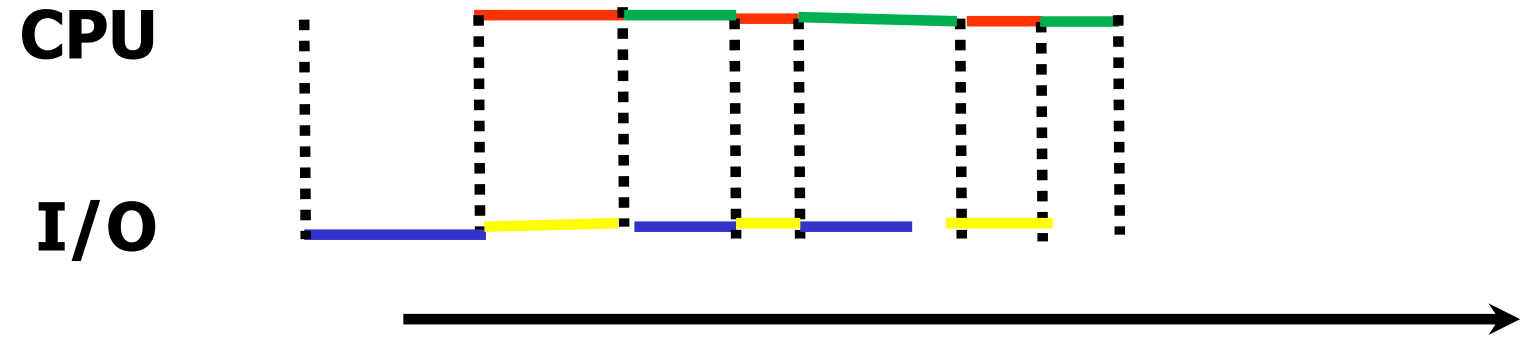
History of Operating Systems

- Phase 2: 1965-1980（Multiprogrammed Batch Systems）

Simple batch system: Job A (blue red) Job B (Yellow green）



Multiprogramming
batch system: Job A (blue red) Job B (Yellow green）



History of Operating Systems

- Phase 2: 1965-1980（Multiprogrammed Batch Systems）
  - OS Features Needed for Multiprogramming
  - ✓ I/O routine supplied by the system.
  - ✓ Memory management– the system must allocate the memory to several jobs.
  - ✓ CPU scheduling – the system must choose among several jobs ready to run.
  - ✓ Allocation of devices.

History of Operating Systems

- Phase 2: 1965-1980（Multiprogrammed Batch Systems）
  - OS Features Needed for Multiprogramming
  - ✓ Improve CPU utilization
    7%--100% (in theory)
  - ✓ Improve memory and I/O devices utilization
  - ✓ Increase system throughput

Characteristics of Multiprogrammed Batch Systems:

多道性、无序性、调度性（两级）

History of Operating Systems

- Phase 2: 1965-1980（Multiprogrammed Batch Systems）
  - Can you tell me the concept of “Multiprogramming”?
  - ✓ 两个/两个以上的作业同时入主存
  - ✓ 处于宏观运行状态
  - ✓ 共享所有系统资源

History of Operating Systems

- Phase 3: 1975-1980（Timesharing Systems）
  - Requirements：User need interaction with computer
    - ✓ Response time < 1 sec
  - Solutions
    - ✓ Share CPU by time pieces（时间片）
    - ✓ Time-sharing (multitasking)
  - Users share Main frame
    - ✓ One main frame VS. Multi users & Multi terminal
  - Time-sharing system is a logical extension of multiprogramming
  - 经典操作系统：CTSS、MULTICS、UNIX

History of Operating Systems

- Phase 4: 1980-now（Multiprogrammed Batch Systems）
  - personal computers were developed after LSI (Large Scale Integration) circuits were invented.
  - First Microcomputer:
    - ✓ Intel 8080 CPU + attached 8-inch floppy disk
    - ✓ First disk based OS: CP/M (Control Program for Microcomputers)
  - Programming languages: C/C++, Java, ……
  - OS: DOS, Windows, MacOS, Linux ……

History of Operating Systems

- Phase 4: 1980-now（Multiprogrammed Batch Systems）
  - Bill Gates suggested IBM that they should look at CP/M (by Gary Kildall)
  - The biggest mistake of all:
  - ✓ Kindall refused to sign a non-disclosure agreement
  - ✓ IBM went back to Bill Gates and signed a contract with him to write an OS for their new home computer
  - MS-DOS was based on QDOS, the "Quick and Dirty Operating System" written by Tim Paterson of Seattle Computer Products
  - QDOS was based on Gary Kildall's CP/M
  - Microsoft bought the rights to QDOS for $50,000

Outline

- Computer System overview
- What is an Operating System?
- History of Operating Systems
- The Operating System Zoo
- Supplement Knowledge

The Operating System Zoo

- Mainframe operating systems
  - Room-size computers
  - High I/O capacity
  - Offers:
    - ✓ Batched OS(no interaction, such as large reports)
    - ✓ Multiprogrammed OS(large number of small requests)
    - ✓ Time-sharing OS(multiple users sitting in front of clients)

The Operating System Zoo

- Server operating systems
  - Offer services like print, file, or web
  - UNIX, Windows 2000, Linux
- Multiprocessor operating systems
  - Parallel computing
- Personal computer operating systems
  - Single user with a good GUI, such as Windows 98, Windows 2000, Macintosh OS, Linux

The Operating System Zoo

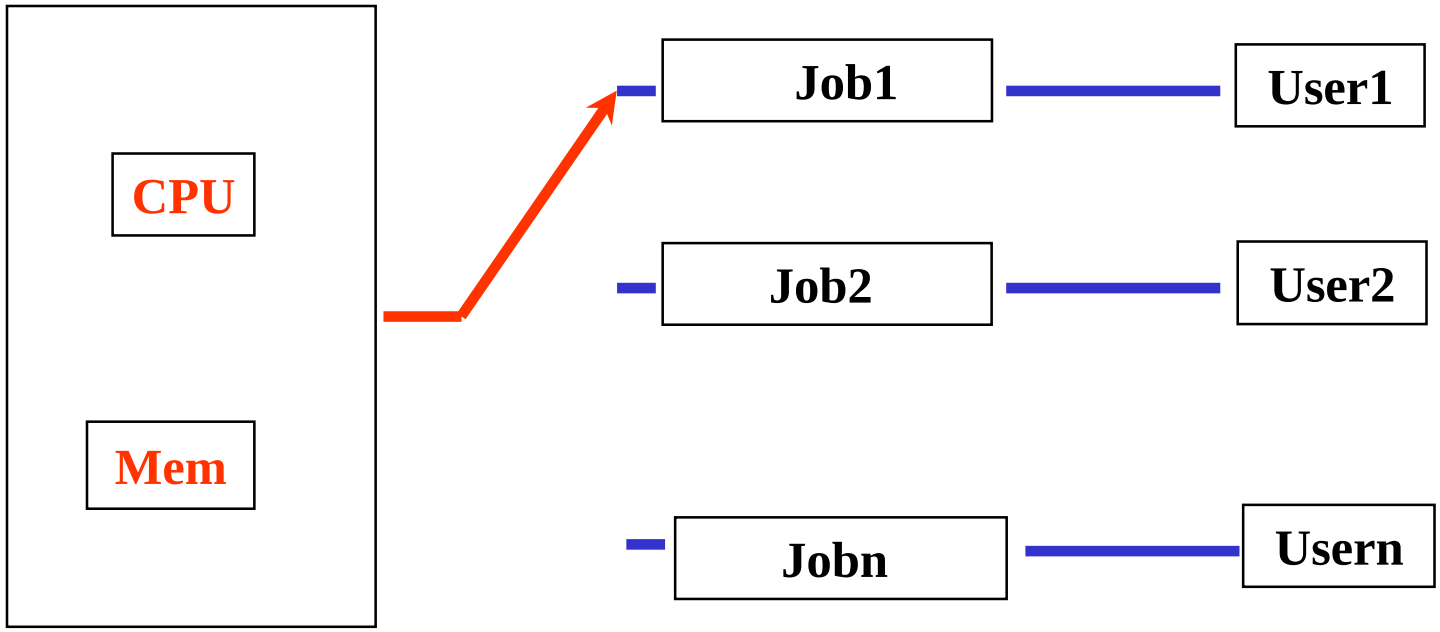
- Real-time operating systems
  - E.g. industrial process control systems where each job must be completed in the specified time.
  - Hard real-time (nuclear reactor control systems) or soft-real time systems (e.g multimedia systems) depending on the acceptance of missing deadlines
- Embedded operating systems
  - Real-time systems with some resource constraints like memory, CPU, power.
- Smart card operating systems
  - Extremely primitive OS running on credit card-sized devices with a CPU.

Mainframe operating systems

- The OS
  - was always resident in memory
  - automatically transferred control from one job to another
- Batching
  - Programmers submitted jobs in a job control language（shell…）
  - Operators batched together jobs with similar needs and run them as a group

The Operating System Zoo

- Time-Sharing Systems

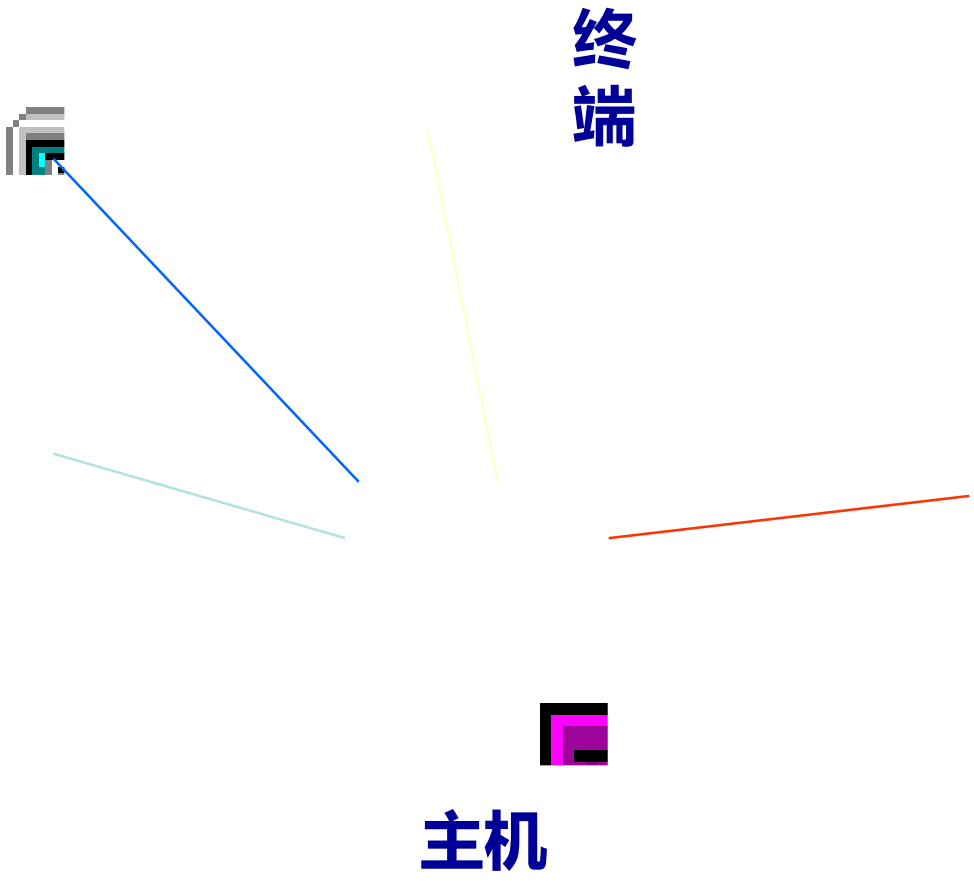


The Operating System Zoo

- Time-Sharing Systems
  - The CPU is multiplexed多路 among several jobs that are kept in memory and on disk (the CPU is allocated to a job only if the job is in memory).
  - A job is swapped in and out of memory to the disk.
  - On-line communication between the user and the system is provided; when the operating system finishes the execution of one command, it seeks the next “control statement” not from a card reader, but rather from the user’s keyboard.
  - On-line system must be available for users to access data and code.

The Operating System Zoo

- Time-Sharing Systems



The Operating System Zoo

- Real-Time Systems（专用系统）
- Often used as a control device in a dedicated application such as
  - ✓ controlling scientific experiments,
  - ✓ medical imaging systems,
  - ✓ industrial control systems, and some display systems.
- Well-defined fixed-time constraints.
  严格的时间限制

The Operating System Zoo

- Real-Time Systems
- Hard real-time system 硬实时系统
  - ✓ Secondary storage limited or absent, data stored in short-term memory, or read-only memory (ROM)
  - ✓ Conflicts with time-sharing systems, not supported by general-purpose operating systems.
- Soft real-time system 软实时系统
  - ✓ Limited utility in industrial control or robotics
  - ✓ Useful in applications (multimedia, virtual reality) requiring advanced operating-system features.

The Operating System Zoo

- Personal-Computer Systems
- Personal computers – computer system dedicated to a single user.
- I/O devices – keyboards, mice, display screens, small printers.
- User convenience and responsiveness.
- Can adopt technology developed for larger operating system, often individuals have sole use of computer and do not need advanced CPU utilization of protection features.

  可采用大型操作系统研发的技术，个人独占计算机不需要高级CPU使用的保护技术

The Operating System Zoo

- Parallel Systems
- Multiprocessor systems with more than one CPU in close communication.
- Tightly coupled system紧耦合系统 – processors share memory and a clock; communication usually takes place through the shared memory.
- Advantages of parallel system:

✓ Increased throughput 吞吐量

✓ Economical

✓ Increased reliability

graceful degradation 故障时性能降低不大

fail-soft systems 软失效系统

The Operating System Zoo

- Parallel Systems
- Symmetric multiprocessing (SMP)对称多处理
  - ✓ Each processor runs an identical copy of the operating system.
  - ✓ Many processes can run at once without performance deterioration.
  - ✓ Most modern operating systems support SMP
- Asymmetric multiprocessing非对称多处理器
  - ✓ Each processor is assigned a specific task; master processor schedules and allocates work to slave processors.
  - ✓ More common in extremely large systems

The Operating System Zoo

- Distributed Systems
- Distribute the computation among several physical processors.
- Loosely coupled system – each processor has its own local memory; processors communicate with one another through various communications lines, such as high-speed buses or telephone lines.
- Advantages of distributed systems.
- ✓ Resources Sharing
- ✓ Computation speed up
- ✓ Reliability
- ✓ Communications

The Operating System Zoo

- Distributed Systems
- Network Operating System
  - ✓ provides file sharing
  - ✓ provides communication scheme
  - ✓ runs independently from other computers on the network
- Distributed Operating System
  - ✓ less autonomy between computers
  - ✓ gives the impression there is a single operating system controlling the network.

Outline

- Computer System overview
- What is an Operating System?
- History of Operating Systems
- The Operating System Zoo
- Hardware features required by OS

（Supplement Knowledge）

Hardware features required by OS

- 任何系统软件都是对硬件系统的延伸，都是建立在硬件基础上的，离不开硬件设施的支持，而操作系统更是直接依赖于硬件，与硬件的关系尤为密切

Hardware features required by OS

- Hardware protection
- System call
- Memory protection
- Interrupt architecture
- I/O systems
- Timer systems

Hardware features required by OS

- Hardware protection
- A properly designed OS must ensure that an incorrect (or malicious) program cannot cause other programs to execute incorrectly.
  - ✓ When in dead loop
  - ✓ When sharing recourses
  - ✓ When one erroneous program might modify the program or data of another program, or even the OS
- Hardware must provide protection
  - ✓ Dual-Mode Operation
  - ✓ I/O protection
  - ✓ Memory protection
  - ✓ CPU protection

Hardware features required by OS

- Protected instruction（特权指令- **privileged instruction**）
- 有些指令只有操作系统才有权使用，比如：
  - ✓ 访问某些硬件资源的指令，这些硬件资源禁止用户程序直接访问；
  - ✓ 对I/O设备的直接访问指令，如磁盘、打印机等；
  - ✓ 对内存管理状态进行操作的指令（页表指针、刷新TLB等）；
  - ✓ 某些特殊的状态位的设置指令；
  - ✓ 停机指令

如何从硬件上实现OS的这个要求?

- 处理器的状态
- 根据运行程序对资源和机器指令的使用权限，把处理器设置为不同的状态；
- 多数系统将处理器工作状态分为管态（**kernel mode**）和目态（**User mode**）：
  - ✓ 管态：操作系统的管理程序运行时的状态，较高的特权级别，又称为特权态、系统态、内核态
  - ✓ 处理机处于管态时：可以执行所有的指令（包括特权指令）、使用所有的资源，并具有改变处理机状态的能力。
  - ✓ 目态：用户程序运行时的状态，较低的特权优先级，又称为普通态（普态）、用户态
  - ✓ 在此状态下，禁止使用特权指令，不能直接使用系

如何从硬件上实现OS的这个要求?

统资源与改变CPU状态，并且只能访问用户程序所在的存储空间；

✓ 有些系统将处理器状态划分为：核心状态、管理状态和用户程序状态三种。

Examples：x86系列处理器

- ✓ 386、486、Pentium系列都支持4个处理器特权级别（特权环R0、R1、R2和R3）；
- ✓ 从R0到R3特权能力一次降低；
- ✓ R0相当于双状态系统的管态；
- ✓ R3相当于目态；
- ✓ R1和R2介于两者之间，他们能够运行的指令集合具有包含关系：

$$I_{R3} \subseteq I_{R2} \subseteq I_{R1} \subseteq I_{R0}$$

Examples：x86系列处理器

- 各个级别有保护性检查（地址校验、I/O限制）
- 特权级别之间的转换方式不尽相同
- 四个级别运行不同类别的程序：
  - ✓ R0-运行OS核心代码；
  - ✓ R1-运行关键设备驱动程序和I/O处理例程；
  - ✓ R2-运行其他受保护共享代码，如语言系统运行环境
  - ✓ R3-运行各种用户程序
- 现有基于x86系列处理器的操作系统，多数UNIX、Linux以及Windows系列大都只用了R0和R3两个特权级别

Question 1

- CPU怎么来判断当前运行的程序是系统程序还是用户程序？

程序状态字（PSW）

- 一个专门的寄存器，用来指示处理机的状态，PSW（Program Status Word），通常包括：
- CPU的工作状态码-指明管态还是目态，用来说明当前在CPU上执行的是操作系统还是一般用户，从而决定其是否可以使用特权指令或拥有 其他的特殊权利；
- 条件码-反映指令执行后的结果特征；
- 中段屏蔽码-指出是否允许中断
- ……

Question：如果CPU只有两个状态：系统态、内核态，用几位来描述CPU状态？

Question 2

- CPU状态之间如何转换？

Question 2

- 管态到目态

  通过设置PSW（修改程序状态字）来实现
- 目态到管态

  用户程序无法直接修改程序状态字
- 那么用户程序如何才能去做一些带有“特权”的事情（如I/O）呢？

解决方法--

Hardware features required by OS

- Protected instruction
- System call
- Memory protection
- Interrupt architecture
- I/O systems
- Timer systems

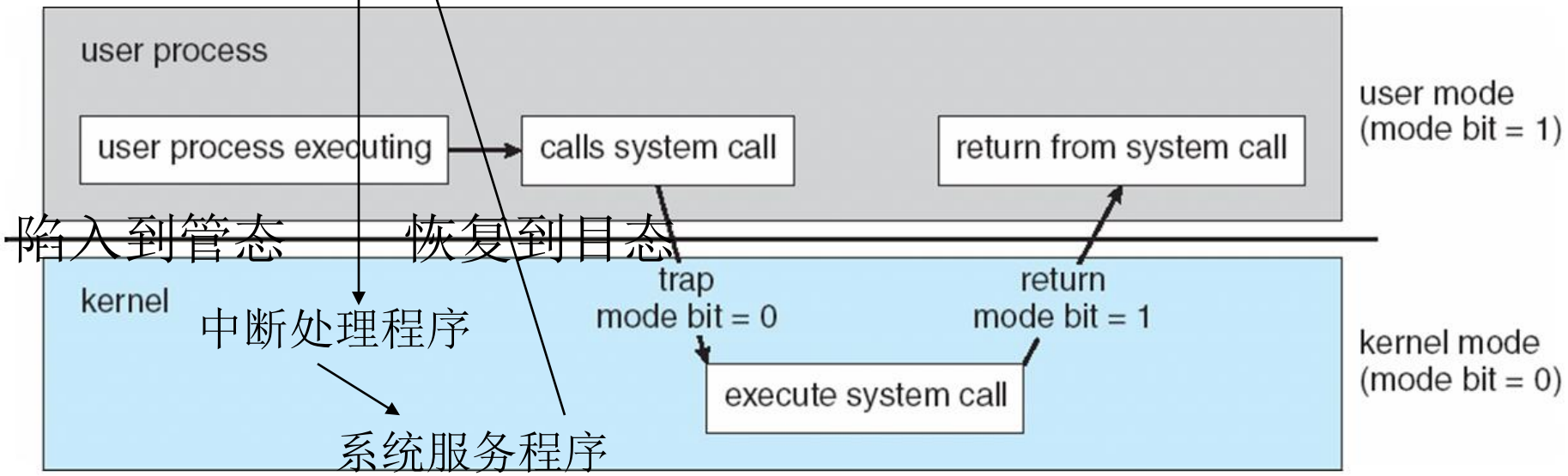
System call

- 用户程序通过特殊的访管指令，来请求操作系统为其提供某种功能的服务，系统调用指令实现的一般过程是
- 当CPU执行访管指令时，即引起访管中断；
- 处理器保存中断点的程序执行上下文环境（PSW、PC和其他的一些寄存器），CPU切换到管态；
- 中断处理程序开始工作，调用相应的系统服务；
- 中段处理结束后，恢复被中断程序的上下文环境，CPU恢复为目态，回到中断点继续执行。

System call

用户程序

系统调用（汇编指令 INT）



OS内核　　管态（系统态）

Hardware features required by OS

- Protected instruction
- System call
- Memory protection（存储管理讲）
- Interrupt architecture
- I/O systems
- Timer systems

中断机制

- 用中断对于操作系统的重要性
  就像机器中驱动齿轮一样
- 当有人把操作系统称为是由
  “ 中断驱动”或者“中断事件驱动”

中断机制

- 什么是中断？
- 指的是由于某个事件的发生，改变了正在CPU上执行的指令顺序；
- 这种事件对应于CPU芯片内部或者外部的硬件电路所生成的电信号
- 中断处理的过程：
- 当中断事件发生时，CPU暂停正在执行的程序，保留现场后自动转去执行相应事件的处理程序，带处理完成后返回断点，继续执行被打断的程序。

中断类型

- Intel公司的文档，中断可以分为两大类：同步中断和异步中断
- 同步中断，指当CPU正在执行指令的时候，由CPU的控制单元所发出的指令，也称为“异常”
- ✓ CPU检测到的异常包括：错误Fault、陷阱Trap、中止Abort，例如：算数溢出、被零除用户态下使用了特权指令等
- ✓ 程序设定的异常，即程序通过int、int3等指令来发出的中断请求，也称为软中断，主要用来实现系统调用服务

中断类型

- 异步中断，指其他的硬件设备在任意时刻发出的中断，简称为“中断”
  - ✓ 可屏蔽中断，即I/O中断，它是当外部设备或通道操作正常结束或发生错误时所发生的的中断。比如，打印机完成、缺纸、读磁盘时驱动器中没有磁盘等
  - ✓ 不可屏蔽中断，比如由掉电、存储器校验错等硬件故障引起的硬件中断
- 每一个中断或者异常都用一个0-255之间的整数来标识，称为中断向量，系统根据中断向量来为每一个中断或异常指定相应的处理程序

中断类型

- I/O系统，完成计算机系统中信息的输入输出
- 时钟
- 时钟是操作系统运行时必不可少的硬件设施，在操作系统中需要时钟支持的工作有：
- 在分时系统中，间隔时钟实现进程间按时间片轮转
- 在实时系统中，按要求的间隔输出正确的时间信号给实时的控制设备
- 记录用户和系统所需的绝对时间（年月日时分秒）

Hardware features required by OS

- Protected instruction
- System call
- Memory protection（存储管理讲）
- Interrupt architecture
- I/O systems
- Timer systems

End of Chapter 1