華南理工大學

South China University of Technology

Intro to Computer Science and Software Engineering

Operating Systems

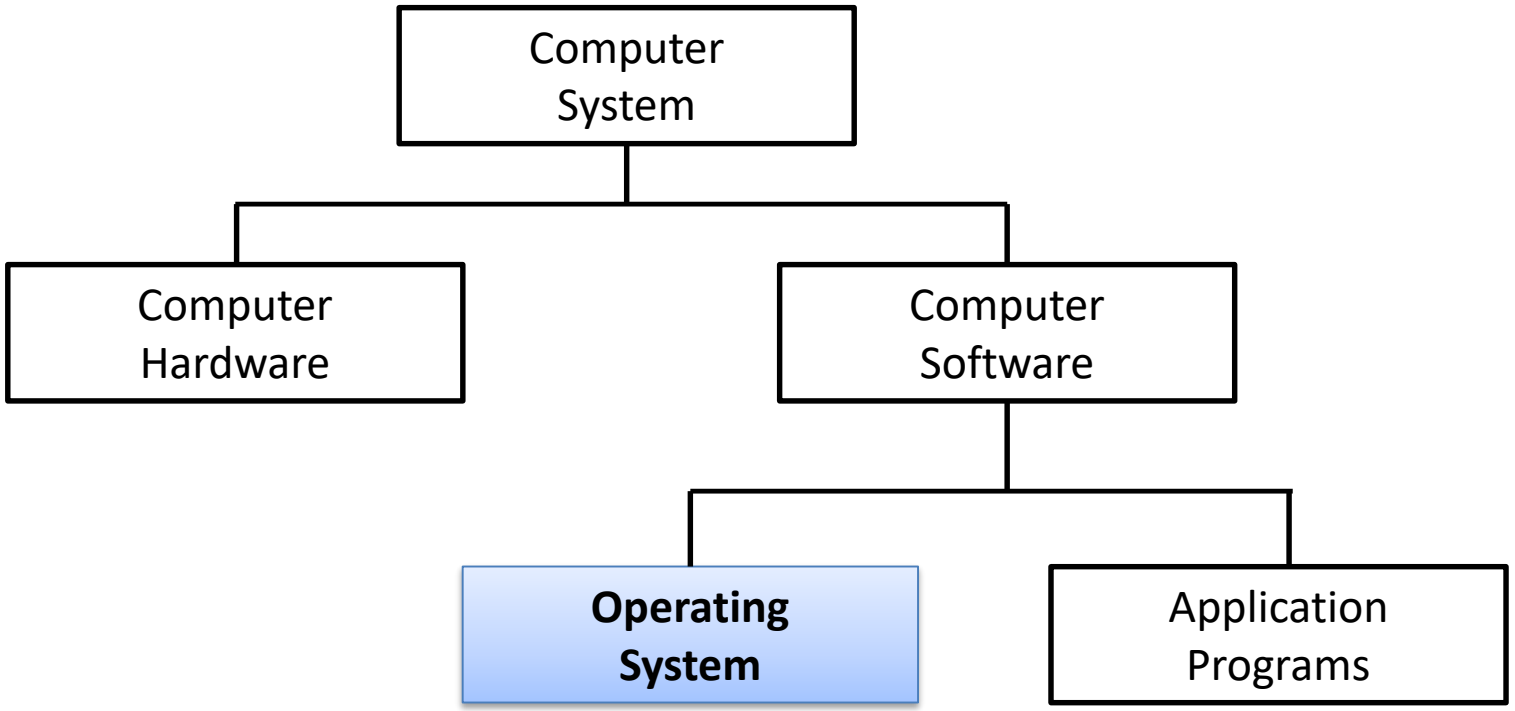
Dr Yubei Lin

yupilin@scut.edu.cn

School of Software Engineering

Computer System

- Software: the collection of programs that allows the hardware to do its job.

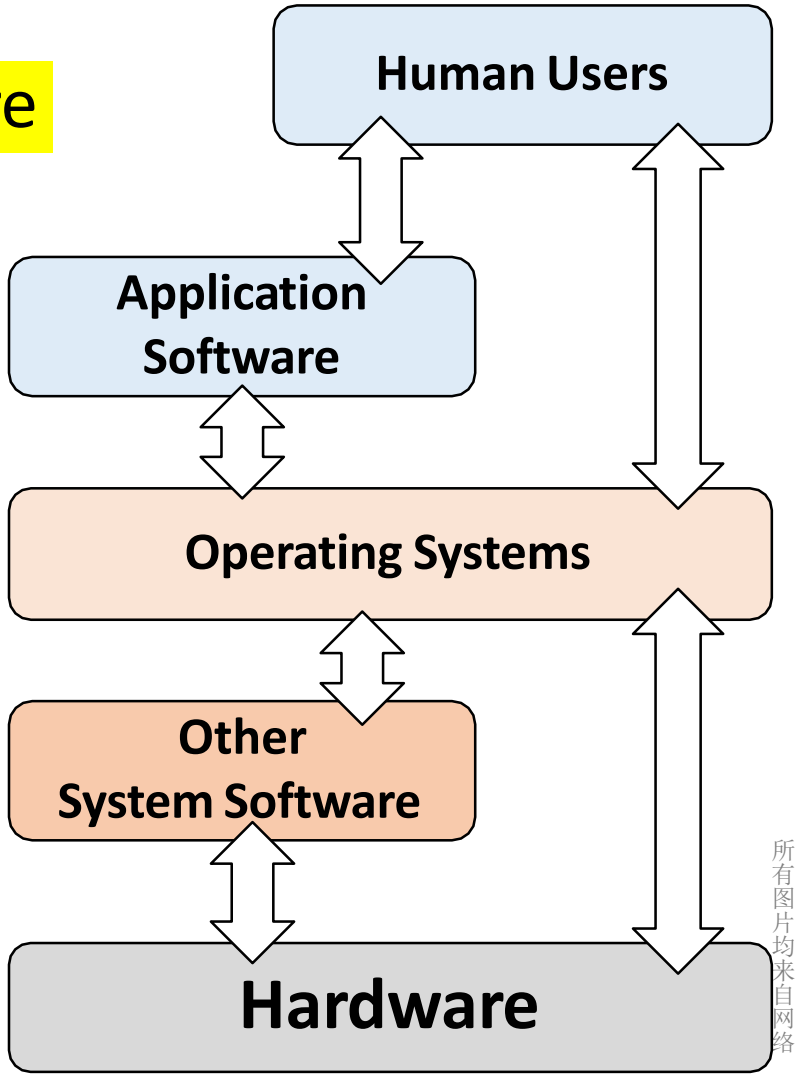


Operating Systems

Operating Systems: System software

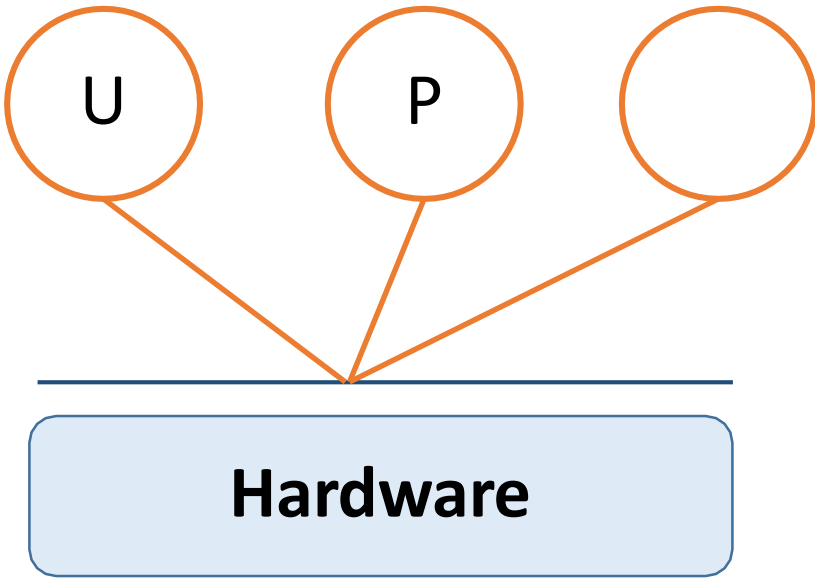
1. manages computer resources, such as memory and input/output devices
2. provides an interface through which a human can interact with the computer
3. allows an application program to interact with resources

What operating systems have you used?



Roles of Operating System

The various roles of an operating system generally revolve around the idea of "sharing nicely". An operating system manages resources, and these resources are often shared in one way or another among programs that want to use them.

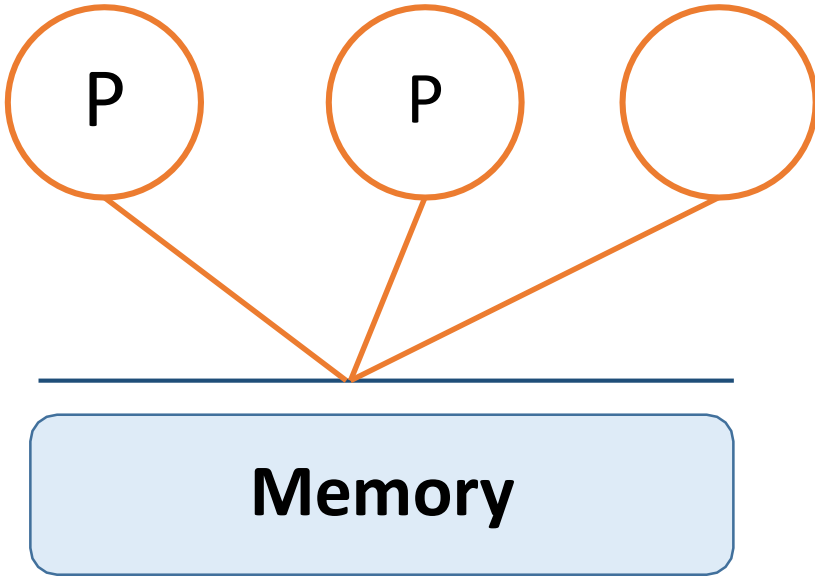


Why operating systems?

Resource Management

Multiprogramming

The technique of keeping multiple programs that compete for access to the CPU in main memory at the same time so that they can execute.



Memory management

The process of keeping track of what programs are in memory and where in memory they reside

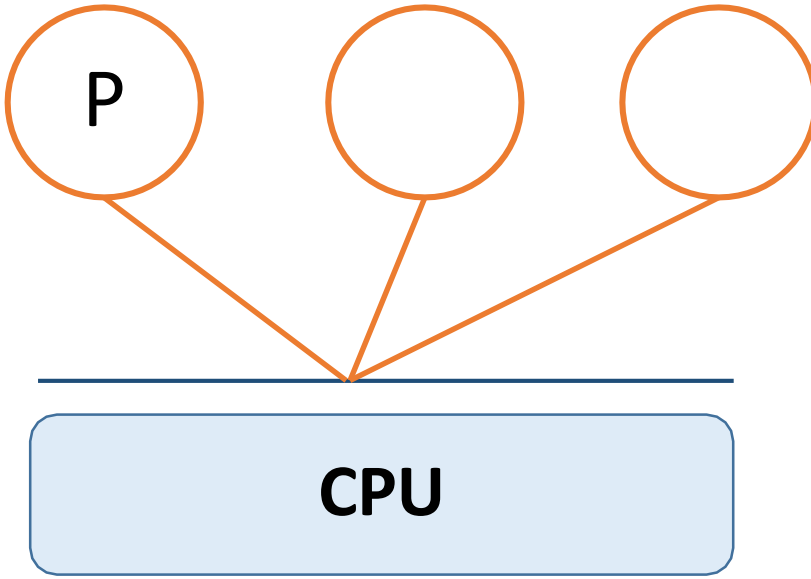
Resource Management

Process

A program in execution

Process management

The act of carefully tracking the progress of a process and all of its intermediate states

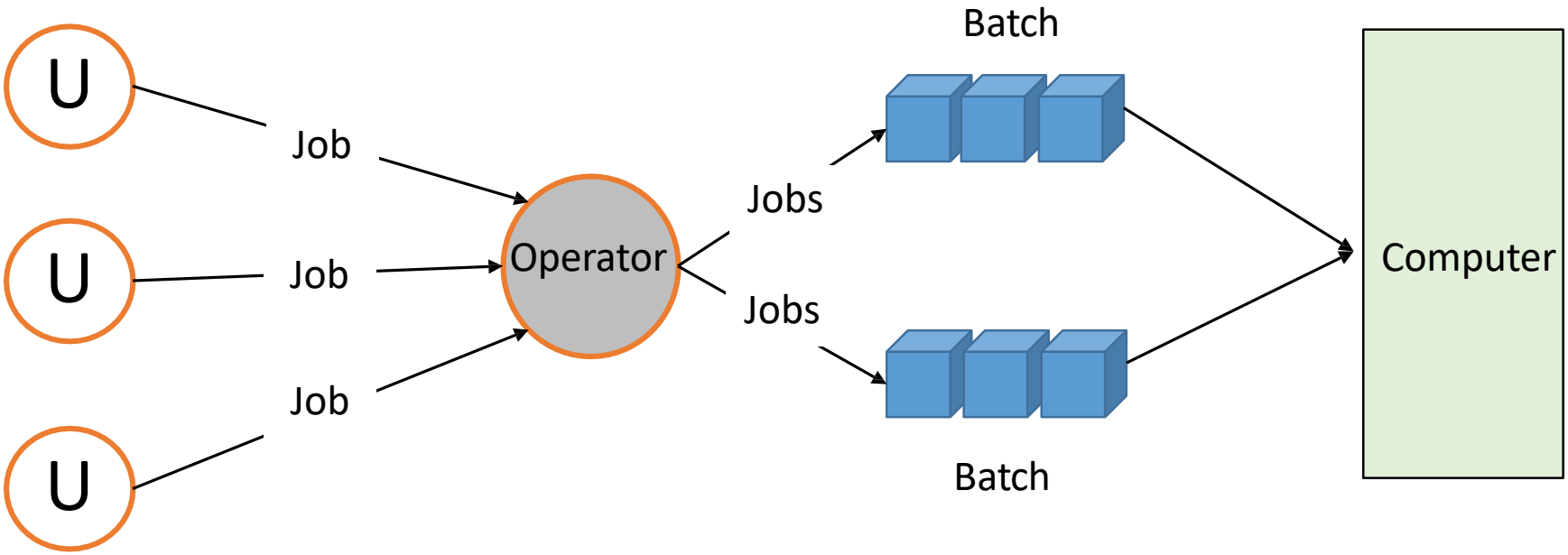


CPU scheduling

Determining which process in memory is executed by the CPU at any given point

History: Batch Processing

The first operating system was a human operator, who organized various jobs from multiple users into *batches* of jobs that needed the same resources.



As computer speed increased, the human operator became the bottleneck

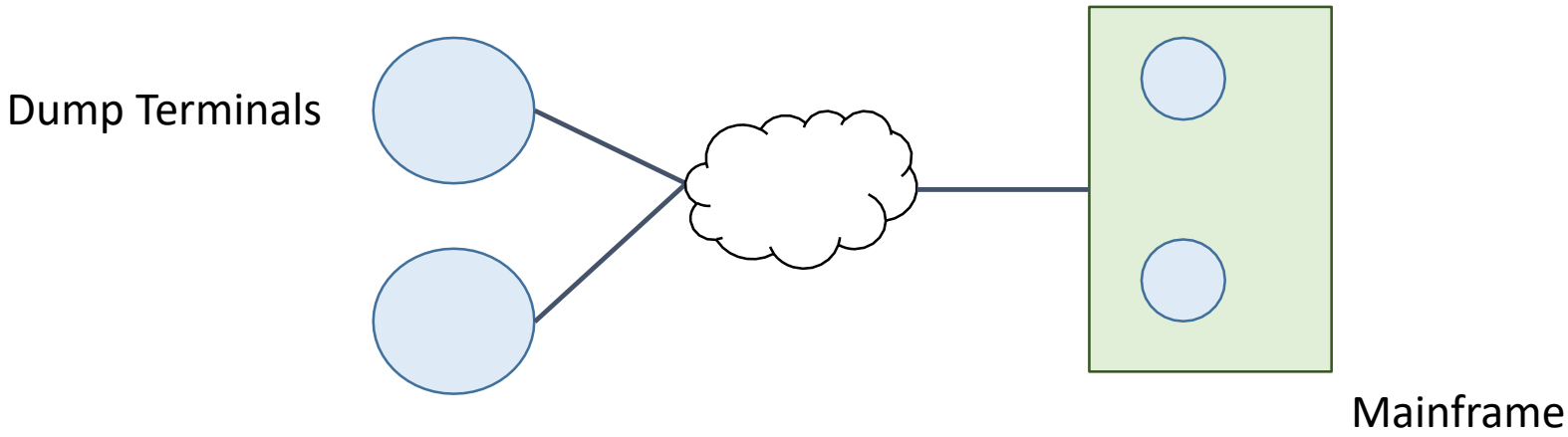
History: Timesharing

Timesharing system

A system that allows multiple users to interact with a computer at the same time

Virtual machine

The illusion created by a time-sharing system that each user has his/her own machine



Other Factors

Real-time System

A system in which response time is crucial given the nature of the application

Response time

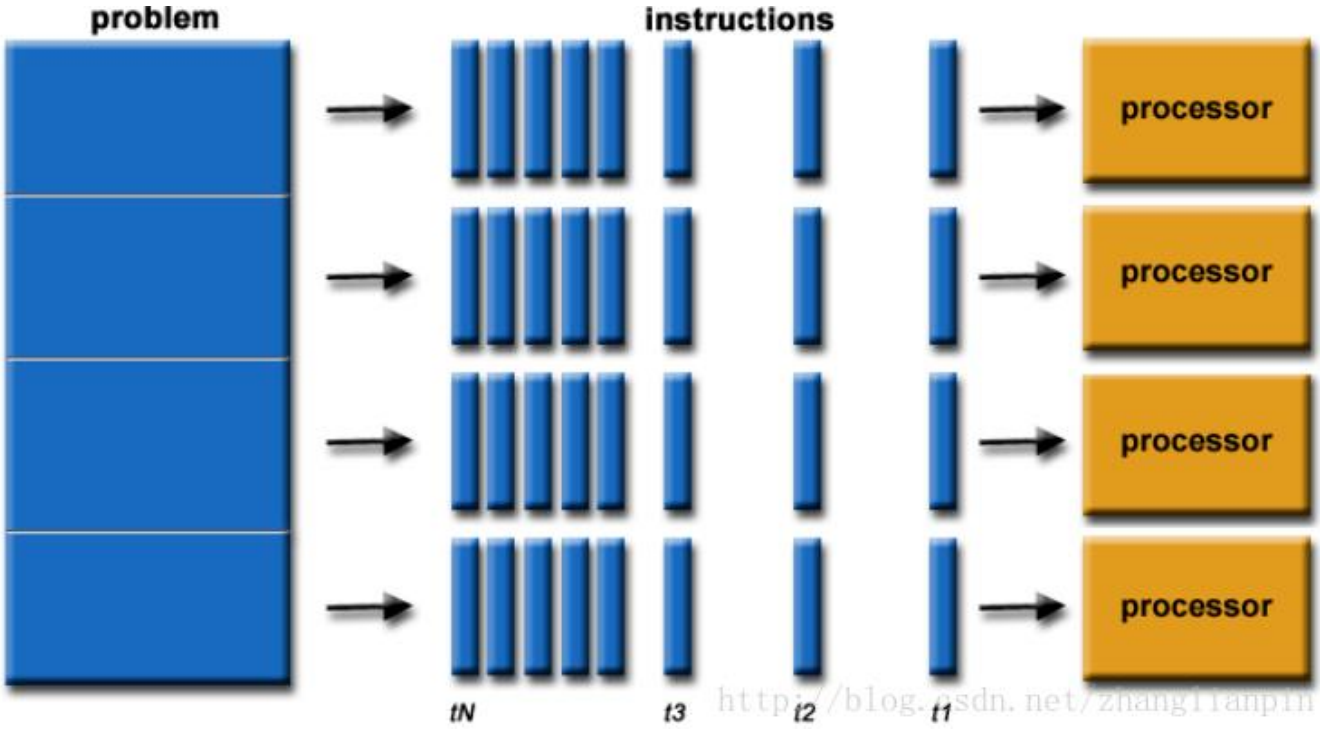
The time delay between receiving a stimulus and producing a response

Device driver

A small program that “knows” the way a particular device expects to receive and deliver information

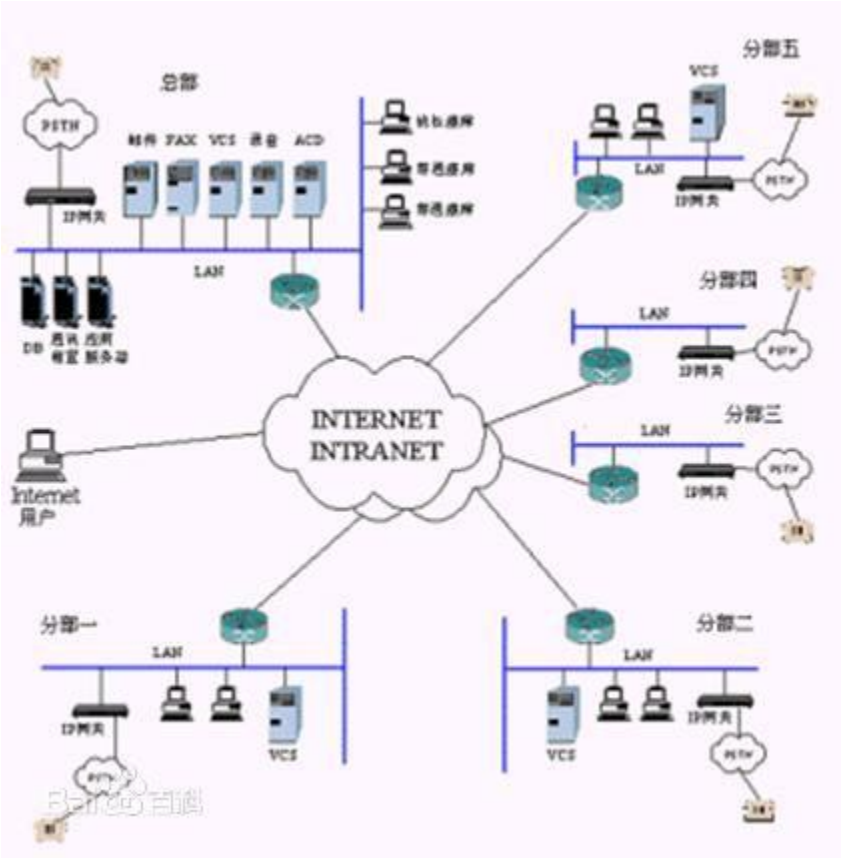
Parallel Systems

- Parallel systems
  - Multiple CPUS
  - Parallel instead of serially: one program uses one CPU



Distributed Systems

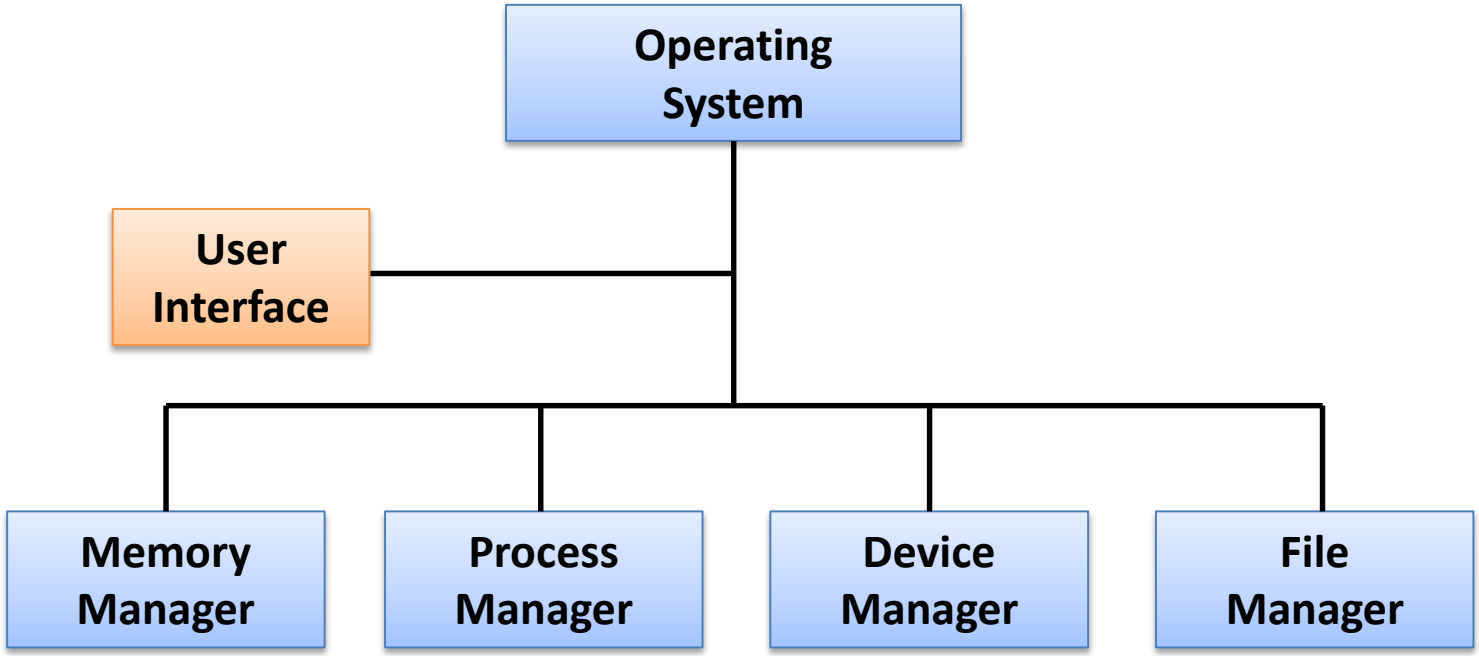
- Distributed systems
  - A job is run across multiple network-connected computers



所有图片均来自网络

Components of OS

- An operating system needs to manage different resources in a computer system.



Memory Management

Operating systems must employ techniques to

1. Track where and how a program resides in memory
2. Convert logical addresses into actual addresses

Logical address

Reference to a stored value relative to the program making the reference

Physical address

Actual address in main memory

Address Binding

The mapping from a logical address to a physical address

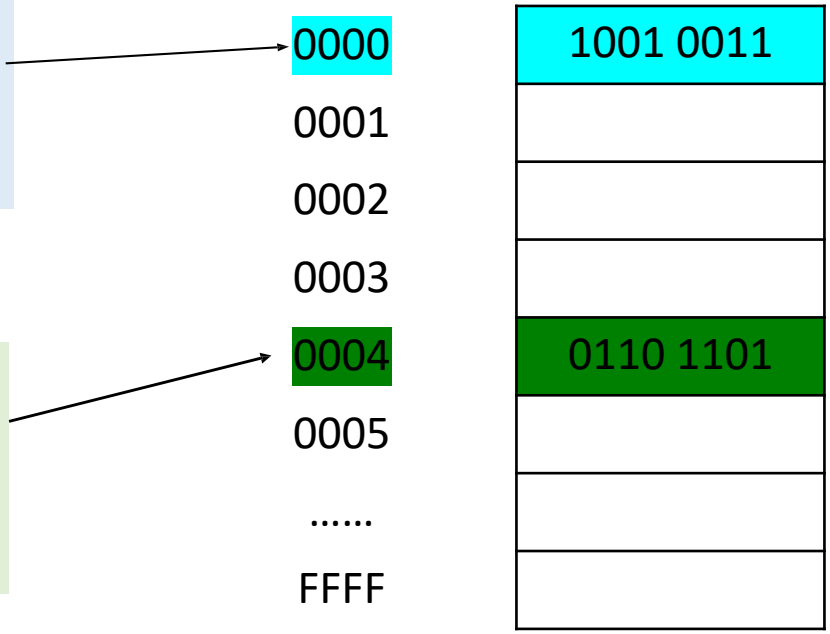
Memory Management

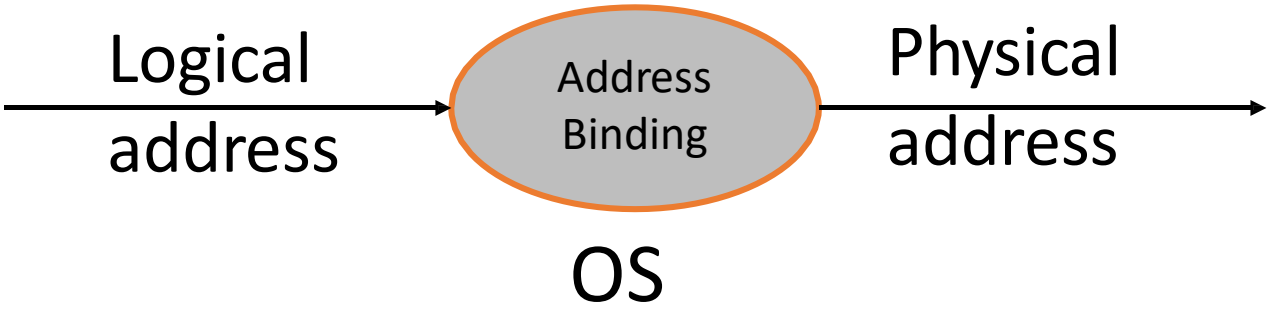
- The size of physical memory is fixed and limited!
- Can you run ‘unlimited’ number of programs?
  - Yes, in demand paging and segmentation
- Virtual memory
  - Logical view of memory: unlimited size!
- Virtual Address to Physical Address?

Memory Management

P1:
int a = -109;
a: reference ~ logical address = 0000

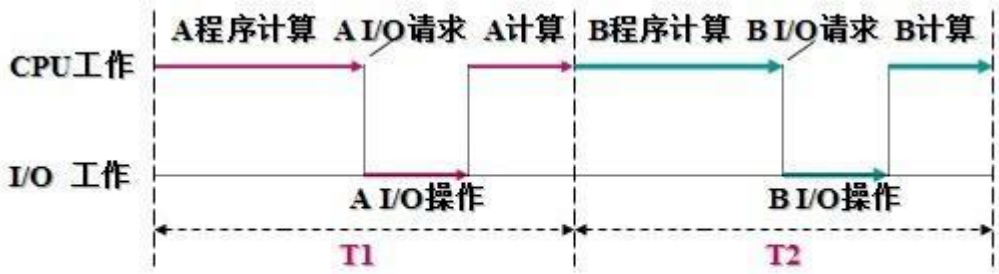
P2:
int a = 109;
a: reference ~ logical address = 0000





Memory Management

- Limited memory vs. many programs and data~
  - Running Out of Memory~~
- **Monoprogramming** (单道程序)
  - Not used today
  - One program a time!
  - 特点：
    - 资源独占性
    - 执行的顺序性
    - 结果的可再现性
    - 运行结果的无关性



单道程序工作示例

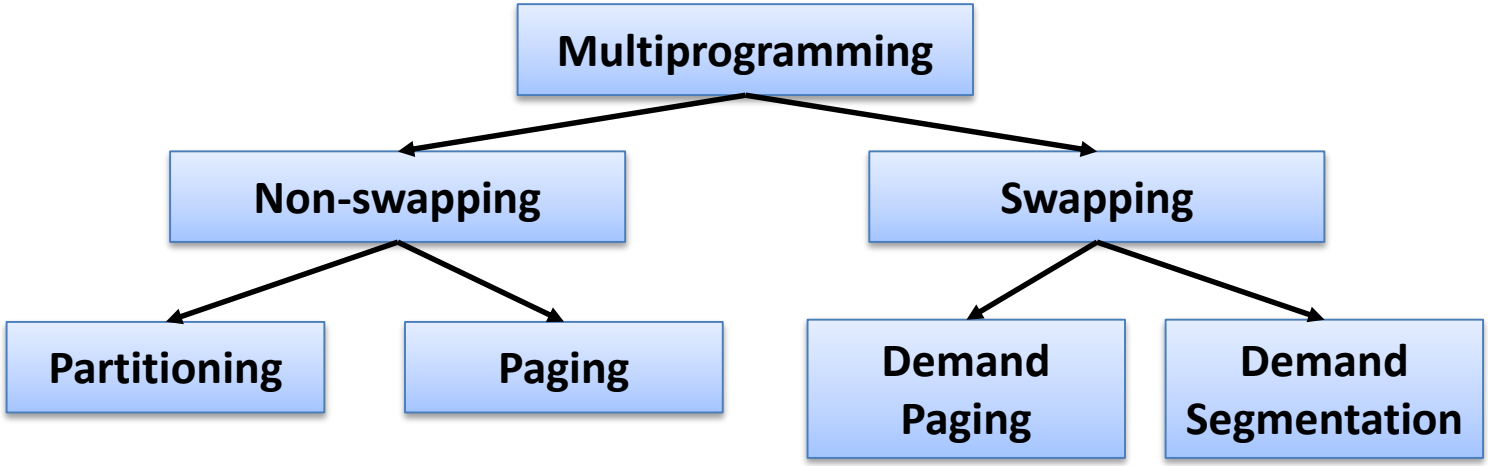
Memory Management

- Multiprogramming
  - More than one program is in memory at the same time, and they are executed **concurrently**!
  - Concurrent: CPU switches between the programs
    - I/O events
    - Time slot exhausted
    - Terminated
  - Context Switching
  - 特点：
    - 提高CPU的利用率
    - 提高内存与I/O的利用率
    - 增加系统的吞吐量

图片均来自网络

Memory Manager

- Memory management in multiprogramming



- Non-swapping
  - A program must be fully loaded into memory before executed
  - A program only be swapped out of memory when finished.

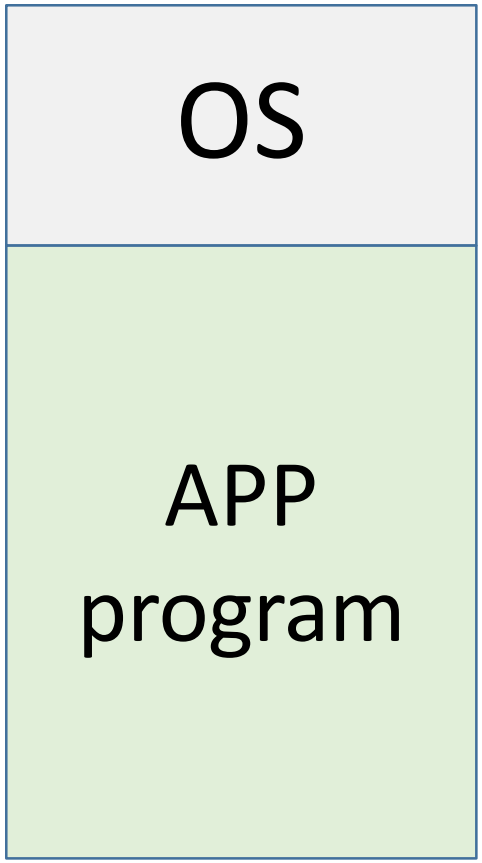
Memory Management

- **Partitioning**
  - Memory is divided into variable length sections
  - ‘holes’: waste of memory
- **Paging**
  - Memory is divided into equal size frames
  - Program is divided into equal size pages
  - Normally, frame size = page size
  - Why this is more efficient than partitioning?

Memory Management

- **Demand paging**
  - Only the current needed pages of a program must be loaded into memory for execution
- **Demand segmentation**
  - Logically, a program is divided into modules, not equally sized pages.
  - Memory is divided into segments.
  - A module is entirely loaded into a segment.
- **Demand paging and segmentation**

Memory Management

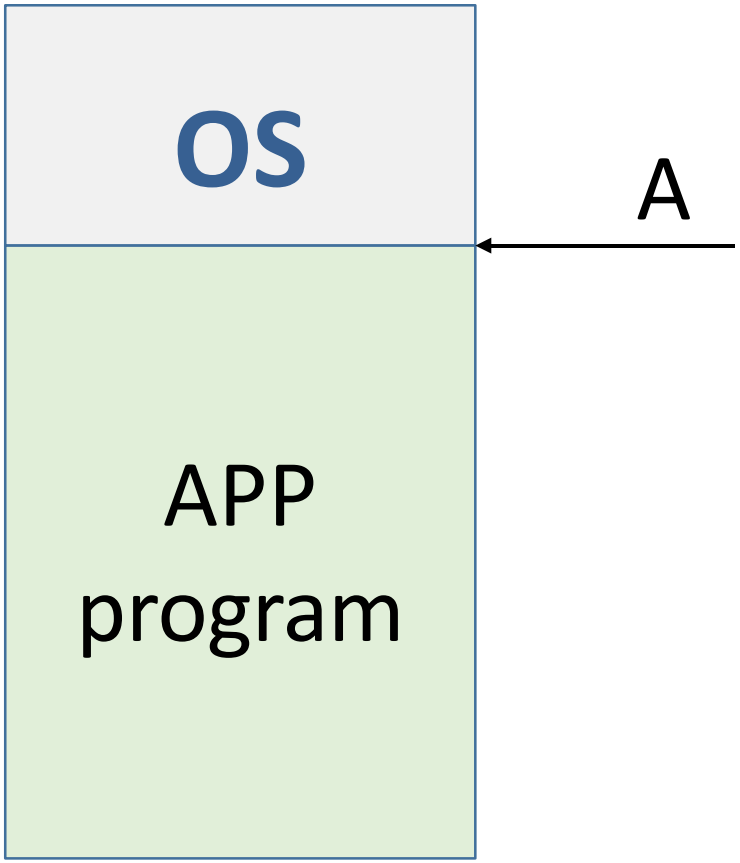


There are only two programs in memory

- The operating system
- The application program

This approach is called **single contiguous memory management**

Memory Management



Logical address: L

Physical address: A+L

Multiprogramming?

Partition Memory Management

Single contiguous MM has only the OS and one other program in memory at one time

Partition MM has the OS and any number of other programs in memory at one time

There are two schemes for dividing up memory for programs:

Fixed partitions Main memory is divided into a fixed number of partitions into which programs can be loaded

Dynamic partitions are created as needed to fit the programs waiting to be loaded

Partition Memory Management

Memory is divided into a set of partitions, some empty and some allocated to programs

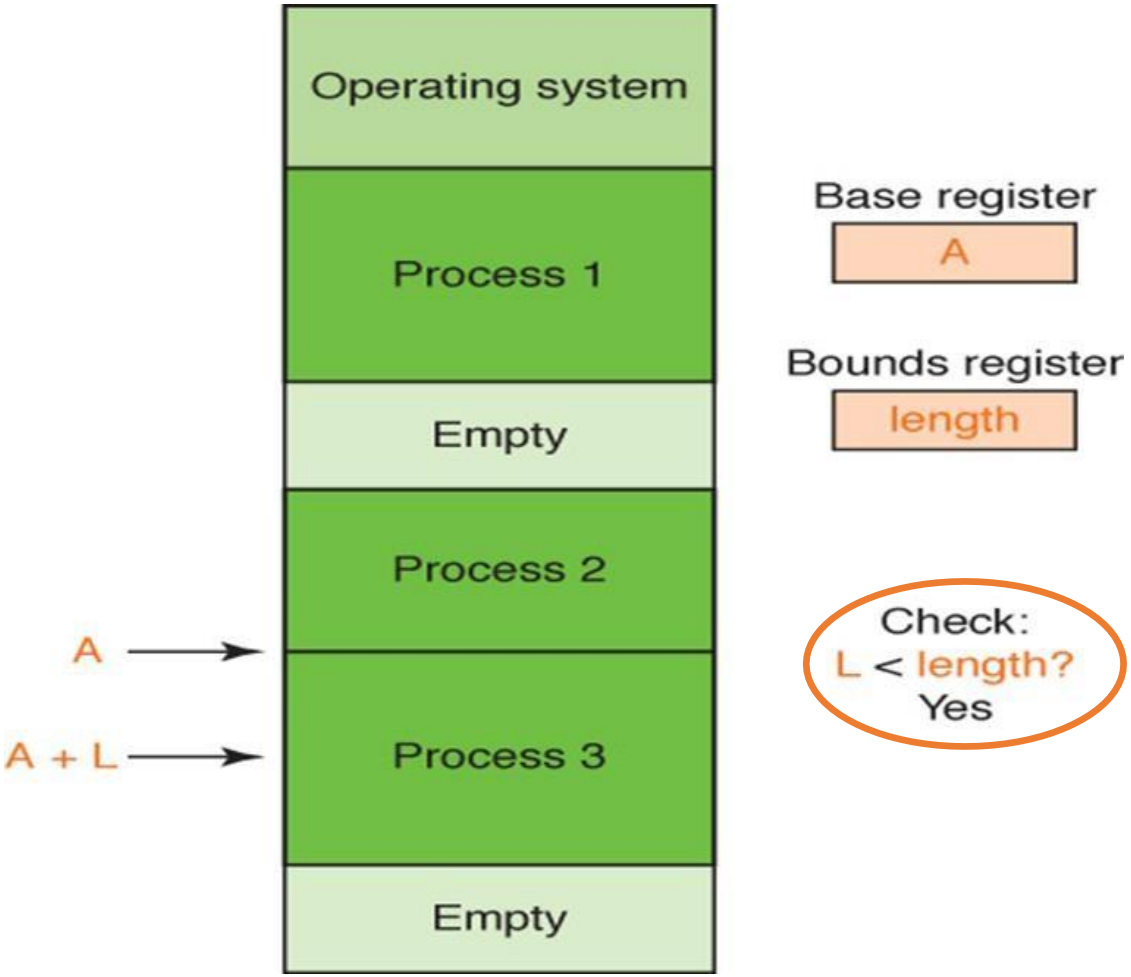
Base register

A register that holds the beginning address of the current partition (the one that is running)

Bounds register

A register that holds the length of the current partition

Partition Memory Management



Why check?

Partition Memory Management

Which partition should we allocate to a new program?

First fit Allocate program to the first partition big enough to hold it

Best fit Allocated program to the smallest partition big enough to hold it

Worst fit Allocate program to the largest partition big enough to hold it

Can you give a rationale for each?

Paged Memory Management

Paged memory technique

A technique in which processes are divided into fixed-size **pages** and stored in memory **frames** when loaded

Frame

A fixed-size portion of *main memory* that holds a process page

Page

A fixed-size portion of a *process* that is stored into a memory frame

We assume that a frame and a page are the same size

Paged Memory Management

P1 PMT

Page	Frame
0	5
1	12
2	15
3	7
4	22

P2 PMT

Page	Frame
0	10
1	18
2	1
3	11

Memory

Frame	Contents
0	
1	P2/Page2
2	
3	
4	
5	P1/Page0
6	
7	P1/Page3
8	
9	
10	P2/Page0
11	P2/Page3
12	P1/Page1
13	
14	
15	P1/Page2

Integer logical address is mapped into a logical address : <page number, offset>

Page number

Address divided by the page size (say 1024)

Offset

The remainder of the address divided by the page size

2566 DIV 1024 = 2

2566 MOD 1024 = 518 ==> <2, 518>

Page Management Table for each program.

Paged Memory Management

Demand paging

An extension of paged memory management in which pages are brought into memory on demand

Page swap

The act of bringing in a page from secondary memory, which often causes another page to be written back to secondary memory

Paged Memory Management

Virtual memory

The illusion that there are no restrictions on the size of a program because an entire process doesn't have to be in memory at the same time

Thrashing

Inefficient processing caused by constant page swaps