華南理工大學

South China University of Technology

Intro to Computer Science and Software Engineering

Operating Systems

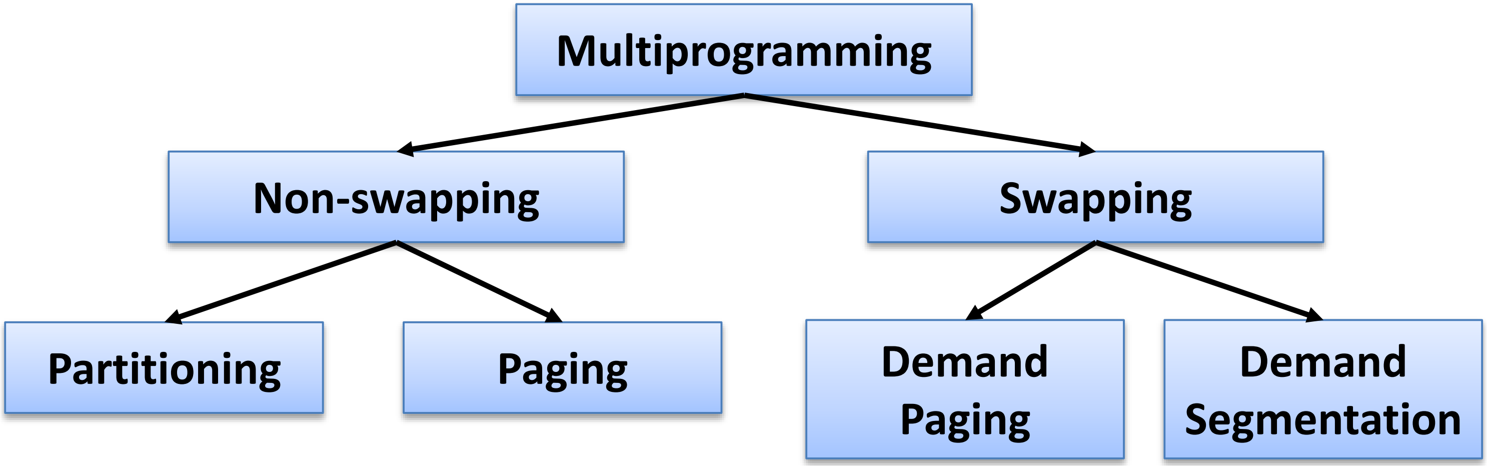
Dr Yubei Lin

yupilin@scut.edu.cn

School of Software Engineering

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

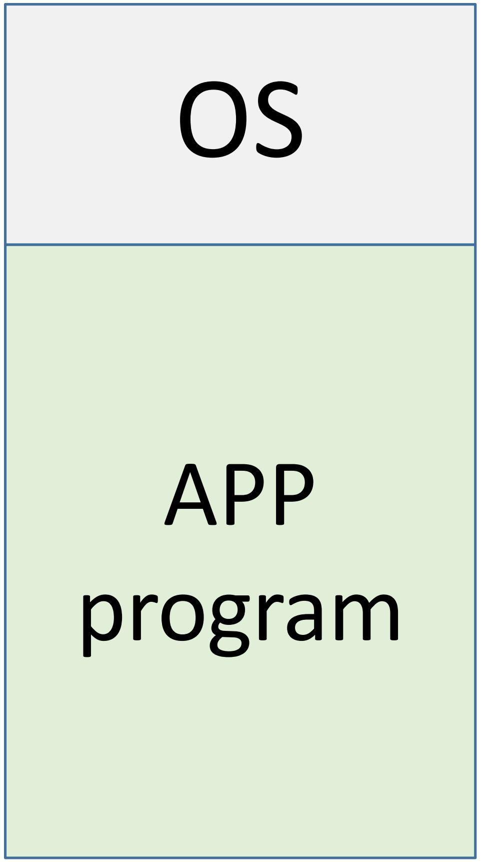
- **Partitioning**
  - Memory is divided into variable length sections
  - ‘holes’: waste of memory
- **Paging**
  - Memory is divided into equal size frames
  - Program is divided into equal size pages
  - Normally, frame size = page size
  - Why this is more efficient than partitioning?

所有图片均来自网络

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

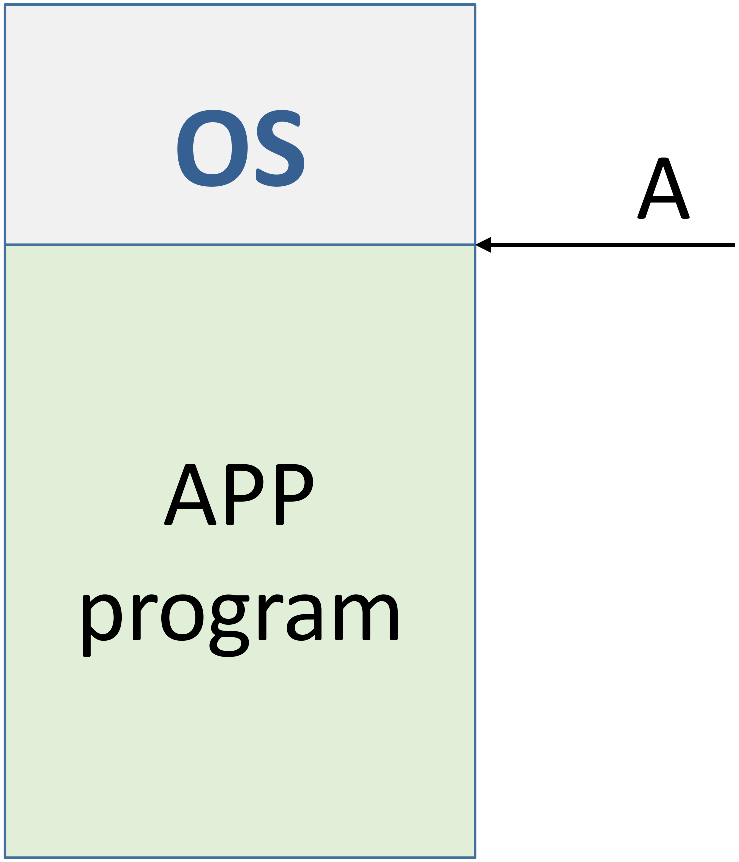


There are only two programs in memory

- 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

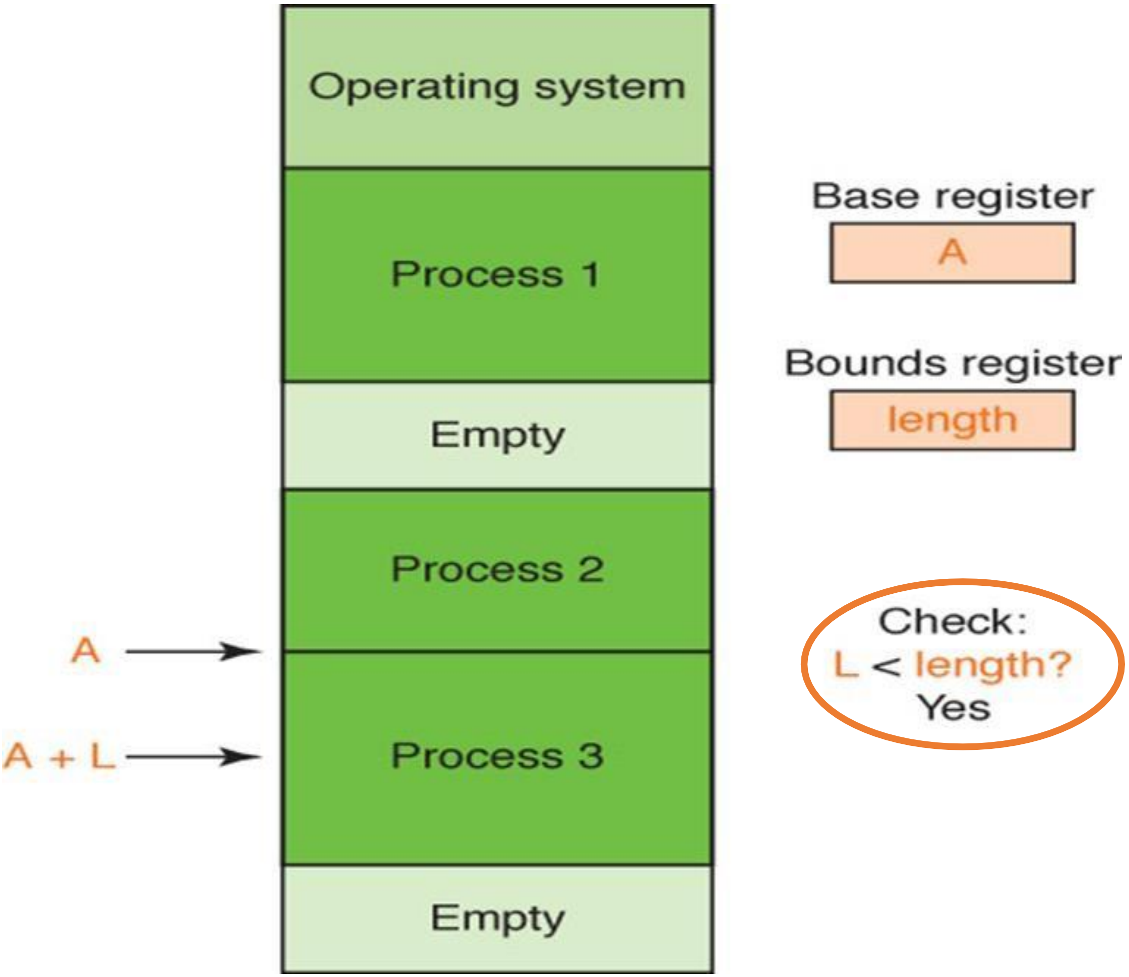
A register that holds the beginning address of the current partition (the one that is running)

Bounds register

A register that holds the length of the current partition

所有图片均来自网络

Partition Memory Management



Why check?

所有图片均来自网络

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

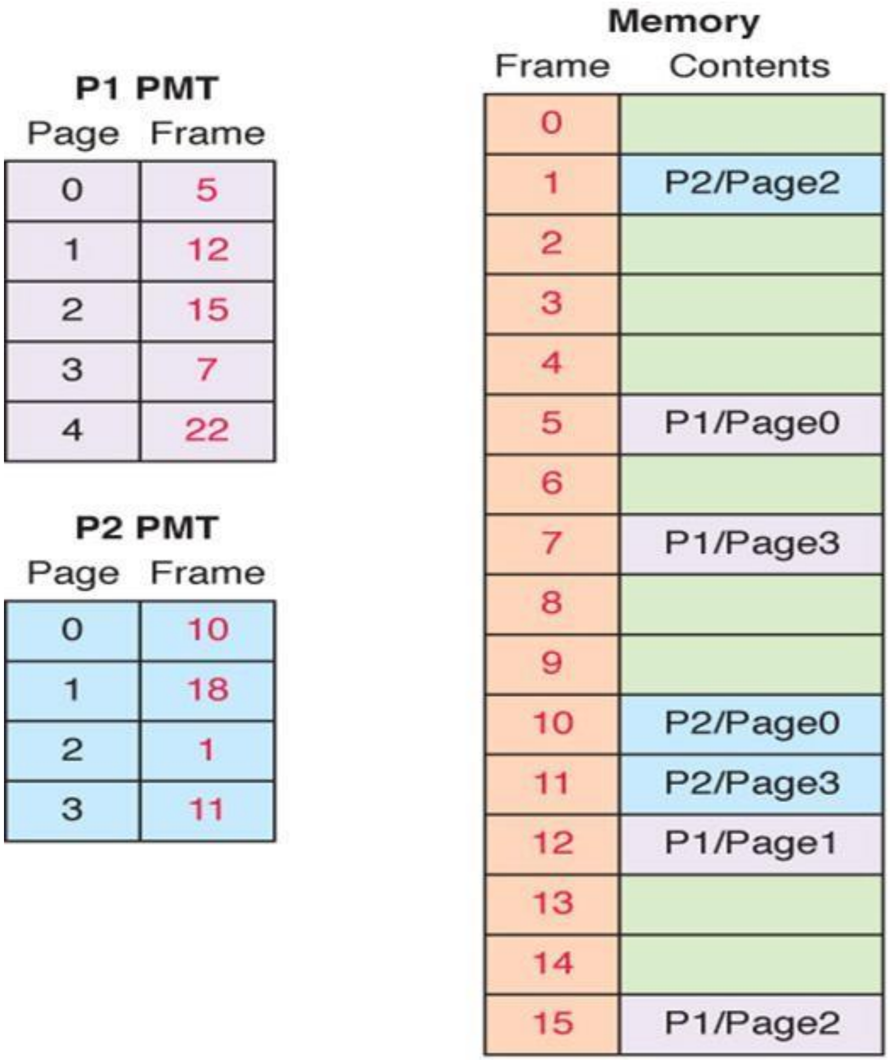
Page

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

所有图片均来自网络

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>

所有图片均来自网络

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

所有图片均来自网络

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

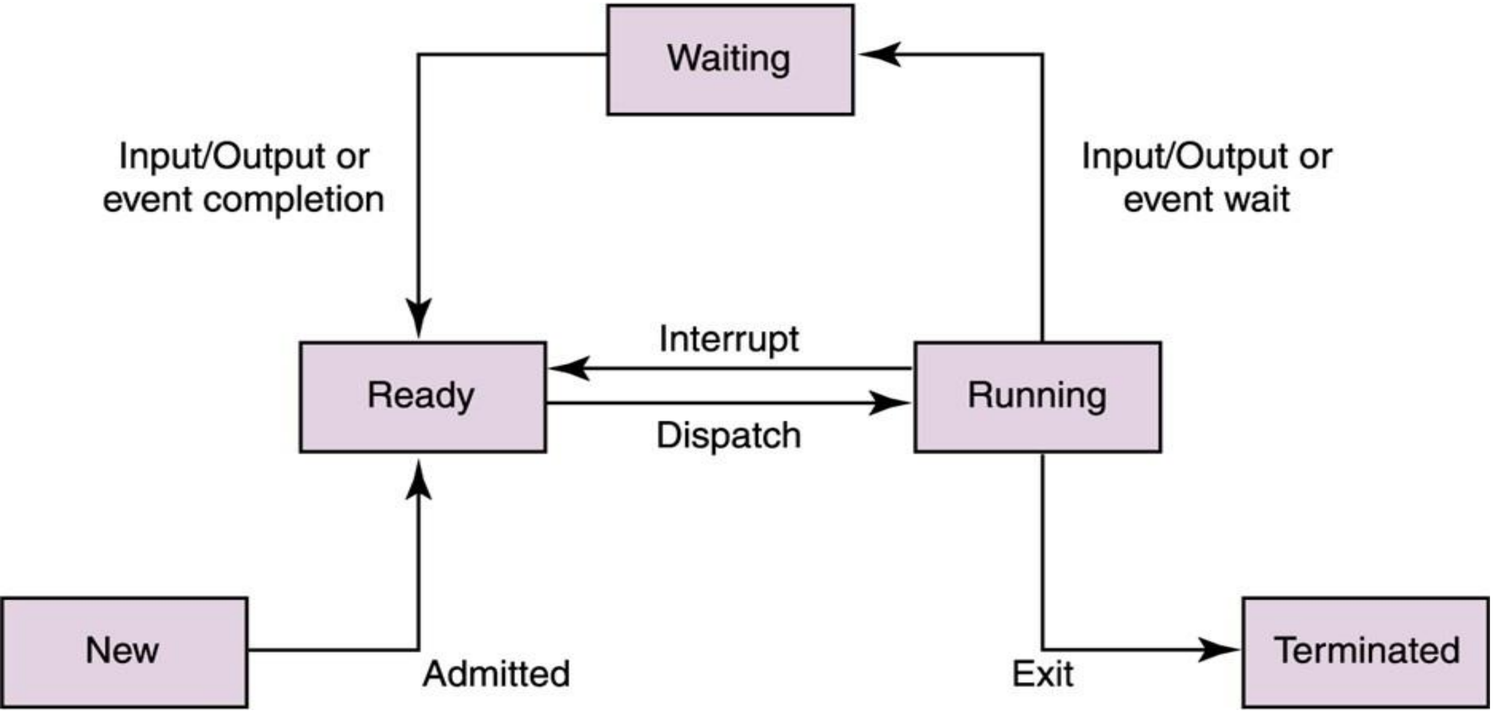
Thrashing

Inefficient processing caused by constant page swaps

Process Manager

A process is a program in execution.

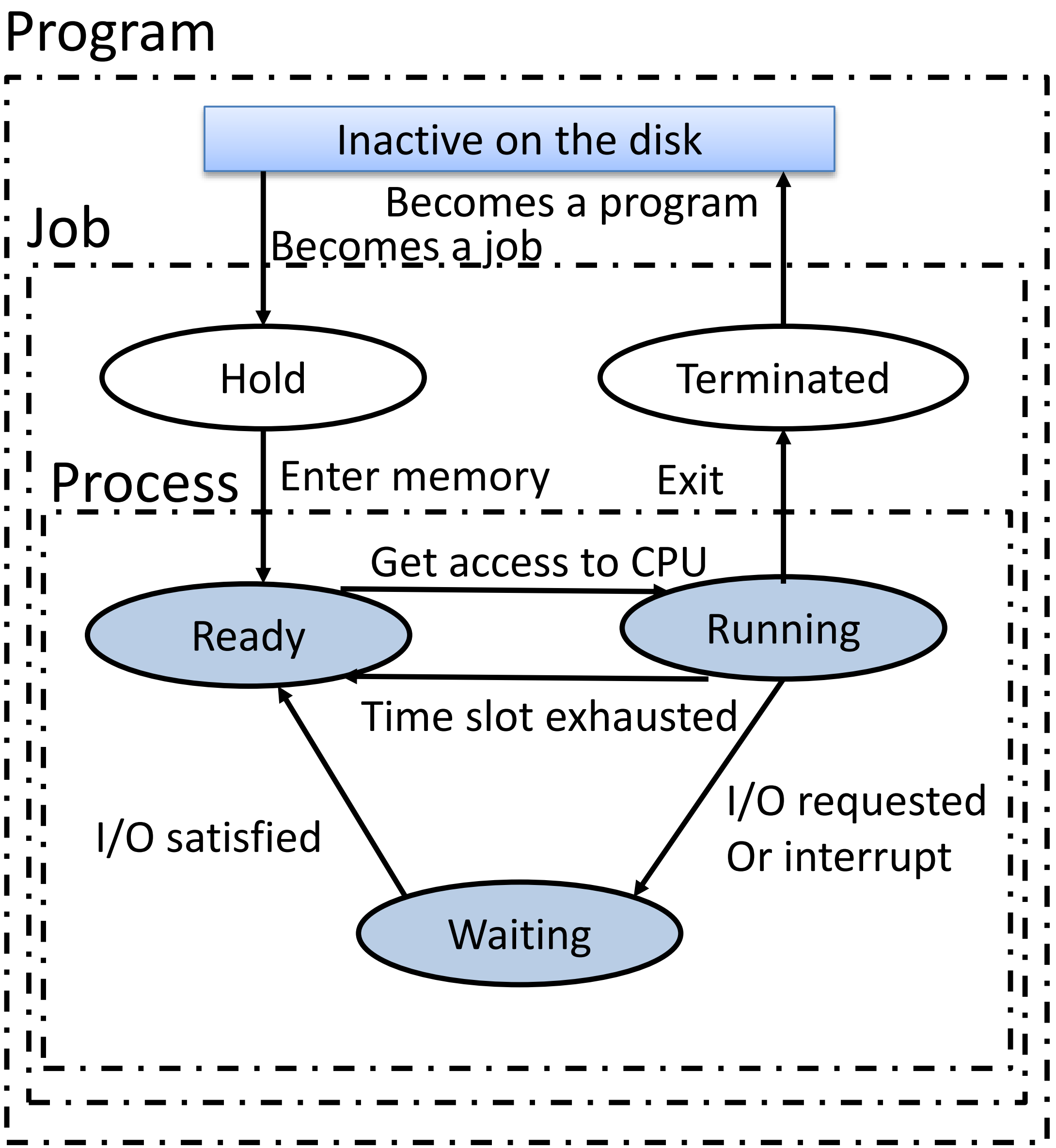
Process management is the act of managing the use of the CPU by individual processes.



所有图片均来自网络

What stages does a process go through?

Process Manager



所有图片均来自网络

Process Manager

Process control block (PCB)

A data structure used by the OS to manage information about a process, including

1. current value of the program counter
2. values of all CPU registers for the process
3. base and bound register values (or page tables)
4. accounting information

Each state is represented by a list of PCBs, one for each process in that state

Process Manager

There is only one CPU and therefore only one set of CPU registers, which contain the values for the currently executing process.

Each time a process is moved to the running state:

1. Register values for the currently running process are stored into its PCB
2. Its PCB is moved to the list of the state into which it goes
3. Register values of the new process moving into the running state are loaded into the CPU
4. This exchange of register information is called a context switch

Process Manager

- Limited number/size of CPU/memory vs. many job or processes
- To handle multiple processes and jobs, the process manager uses schedulers and queues

Process Manager

- Schedulers: controlling access to memory and CPU
  - Job scheduler
    - Creating/terminating a process
  - Process scheduler
    - Moving a process from one state to another (ready, running and waiting

Process Manager

- A queue are a waiting list of jobs or processes
  - Storing Job or process control block
  - E.g. the job queue, the ready queue and the I/O queue
- Policy is needed for selecting the next job or process from a queue.
  - E.g. FIFO, shortest length first, etc

所有图片均来自网络

Process Manager

- Process synchronization
  - – Managing multiple processes accessing share resources
  - – Two interesting situations
    - Deadlock
    - Starvation

Process Manager

- Deadlock
  - OS does not put resources restrictions on processes
- Starvation
  - OS put too many resource restrictions on processes

所有图片均来自网络

CPU Scheduling

The act of determining which process in the ready state should be moved to the running state

1. Many processes may be in the ready state
2. Only one process can be in the running state, making progress at any one time

Which one gets to move from ready to running?

CPU Scheduling

Nonpreemptive scheduling（非抢占式）

The currently executing process gives up the CPU voluntarily.

Preemptive scheduling（抢占式）

The operating system decides to favor another process, preempting the currently executing process.

CPU Scheduling

First-Come, First-Served

Processes are moved to the CPU in the order in which they arrive in the running state.

Shortest Job Next

Process with shortest estimated running time in the ready state is moved into the running state first.

Round Robin

Each process runs for a specified time slice and moves from the running state to the ready state to await its next turn if not finished.

CPU Scheduling

Turnaround time（周转时间）

The amount of time between when a process arrives in the ready state the first time and when it exits the running state for the last time.

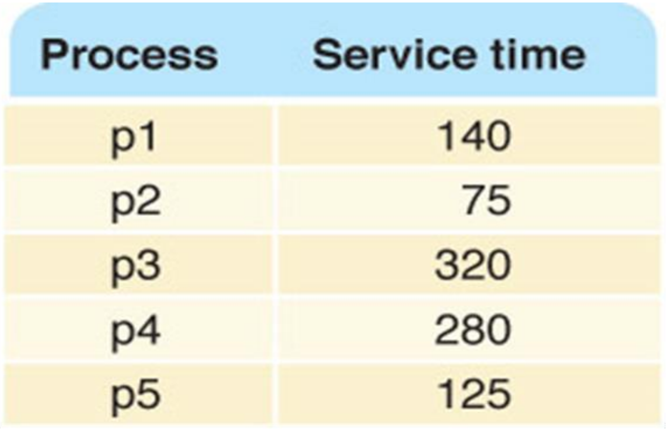
例：有3个进程P1、P2、P3先后到达，它们分别需要20、4、2个单位时间运行完毕。如果它们按P1、P2、P3的顺序执行，且不可抢占，则3个进程的周转时间分别是多少个单位时间？如果用时间片原则的抢占调度方式（假设时间片为2个单位时间），则3个进程的周转时间分别是多少个单位时间？

（1）20,24,26　　（2）26,10,6

所有图片均来自网络

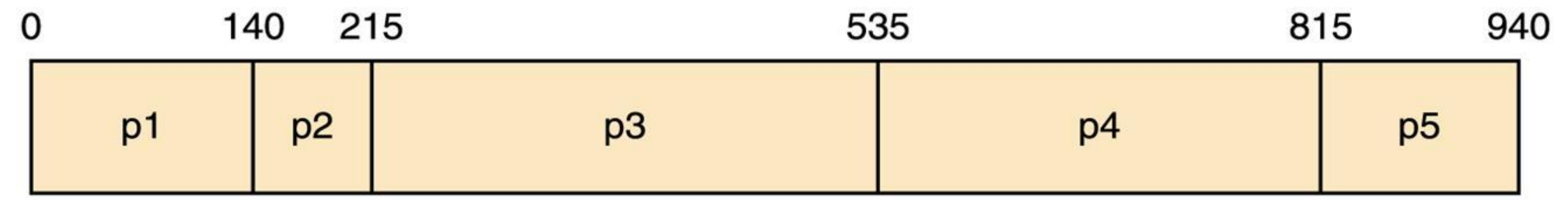
CPU Scheduling

First-Come, First-Served



Process	Service time
p1	140
p2	75
p3	320
p4	280
p5	125

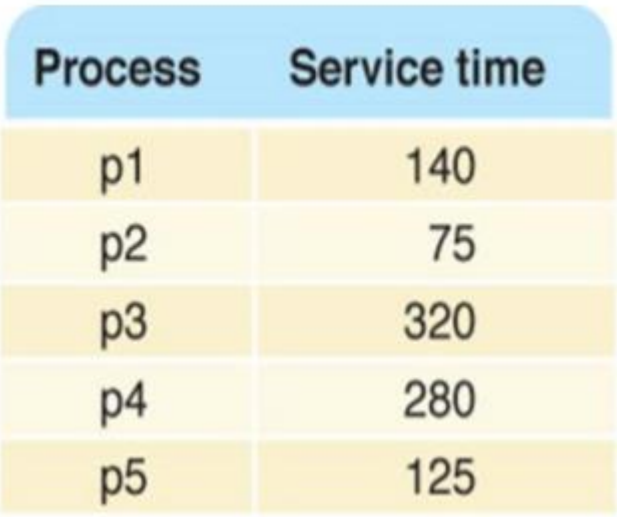
What is the average turnaround time?



所有图片均来自网络

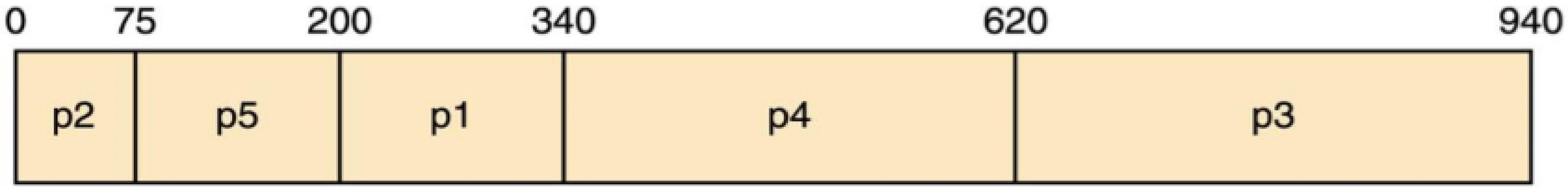
CPU Scheduling

Shortest Job Next



Process	Service time
p1	140
p2	75
p3	320
p4	280
p5	125

What is the average turnaround time?



所有图片均来自网络

CPU Scheduling

Round Robin

Every process is treated the same!

Time slice (quantum)

The amount of time each process receives before being preempted and returned to the ready state to allow another process its turn

所有图片均来自网络

CPU Scheduling

Round Robin

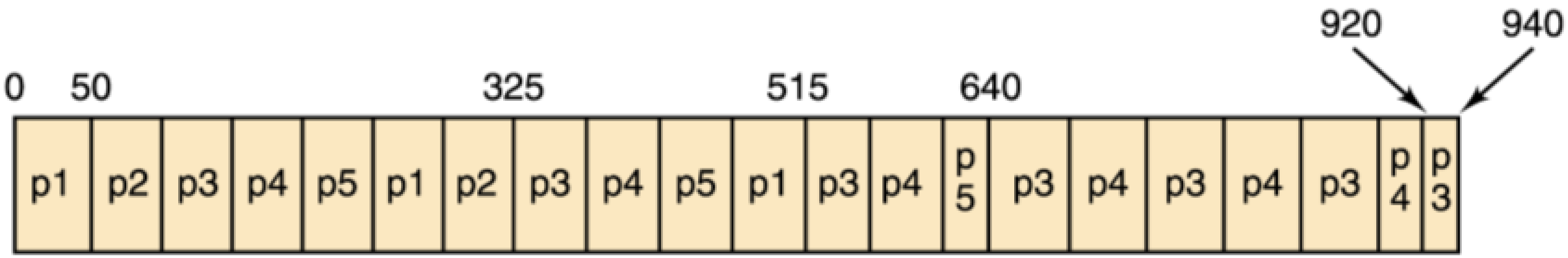
Every process is treated the same!

Time slice (quantum)

The amount of time each process receives before being preempted and returned to the ready state to allow another process its turn

Suppose the time slice is 50

What is the average turnaround time?



所有图片均来自网络

Device Manager

- “I/O devices”
  - Exclusive access, and limited number
  - slower in speed compared with the CPU and memory
- Device manager is responsible for the efficient use of I/O devices
  - Maintaining queues for available devices
  - Monitoring the devices’ status
  - Scheduling access to devices (policy needed)

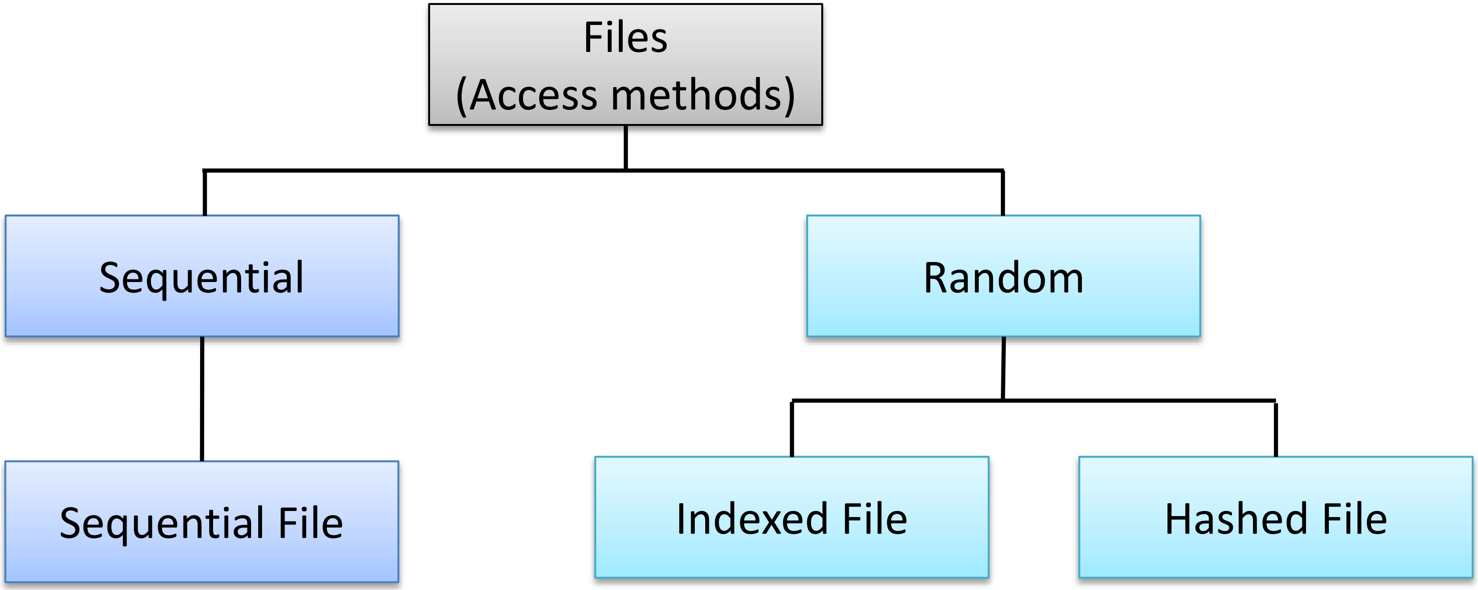
File: concepts

- File: an external collection of related data treated as a unit.
  - Physical view: storing in external storage devices, such as tape, disk, etc. (not in memory)
  - Logical view: a collection of data records with each record consisting of one or more fields.
- In Operating System, File system would look at its physical view.
- We only look at its logical view in this course, that is the File Structure!

所有图片均来自网络

Access methods

- We category file according to access methods:
  - How can you retrieve the information (a specific record) from the file.
- Two types: Sequential or random



Files
(Access methods)

Sequential

Random

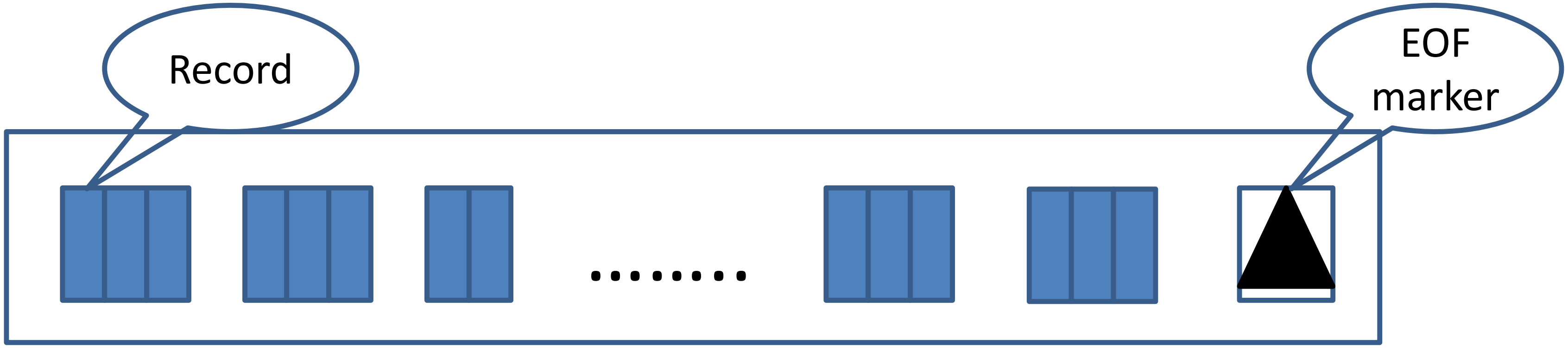
Sequential File

Indexed File

Hashed File

Sequential files

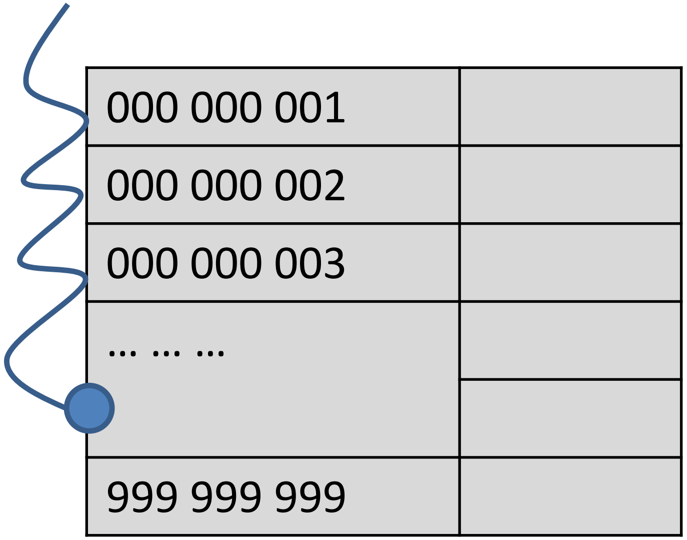
- A sequential file is one in which records can only be accessed sequentially, one after another, from beginning to end.



Sequential files

- Suitable cases?
  - Printing every employee's paycheck?
  - Checking one customer's balance?

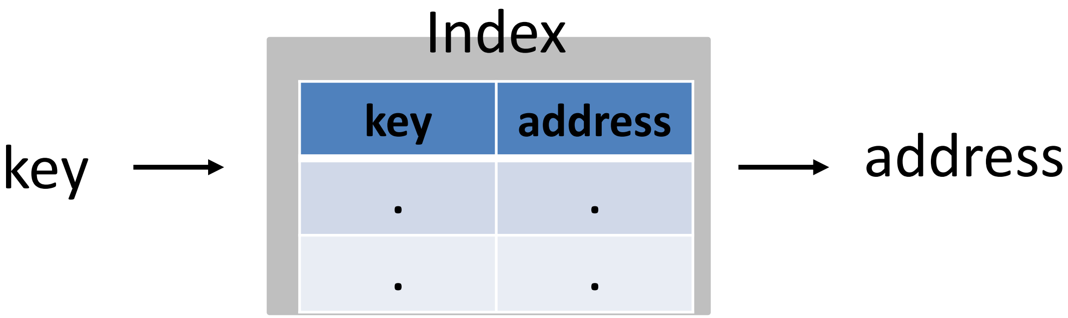
Locating one record in a large sequential file is very **inefficient**!



000 000 001	
000 000 002	
000 000 003	
… … …	
999 999 999	

Indexed files

- Accessing randomly, one must know the addresses of records.
  – How can you know the addresses?
- An indexed file:
  – Data file, which is a sequential file,
  – Indexed file, a very small file with only two fields:
    - The key of sequential file
    - The address of the corresponding record of the file



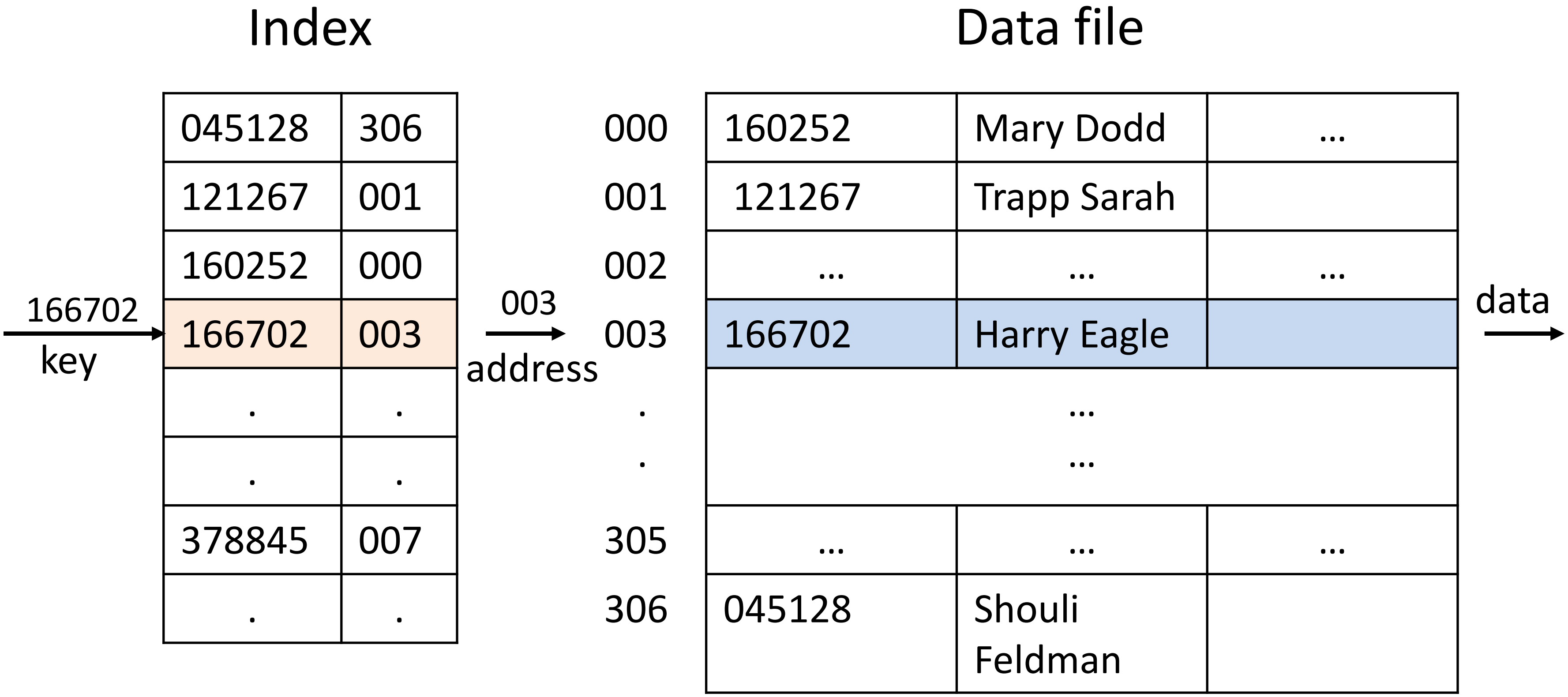
key →

Index

key	address
.	.
.	.

→ address

Indexed files



045128	306
121267	001
160252	000
166702	003
.	.
.	.
378845	007
.	.

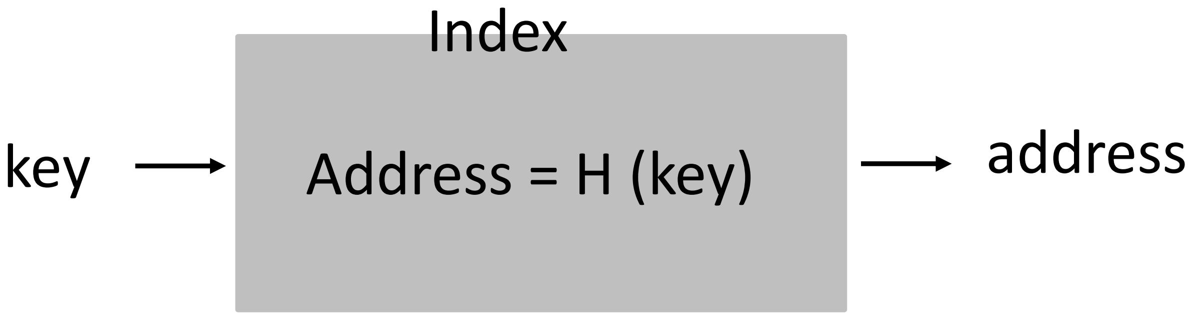
000	160252	Mary Dodd	…
001	121267	Trapp Sarah	
002	…	…	…
003	166702	Harry Eagle	
.		…	
.		…	
305	…	…	…
306	045128	Shouli Feldman	

Logical view of an indexed file

所有图片均来自网络

Hashed files

- A hashed file uses a function to map the key to the address!



Index

key

Address = H (key)

address

- Hash methods
  - Direct method
  - Modulo division method
  - Digit extraction method

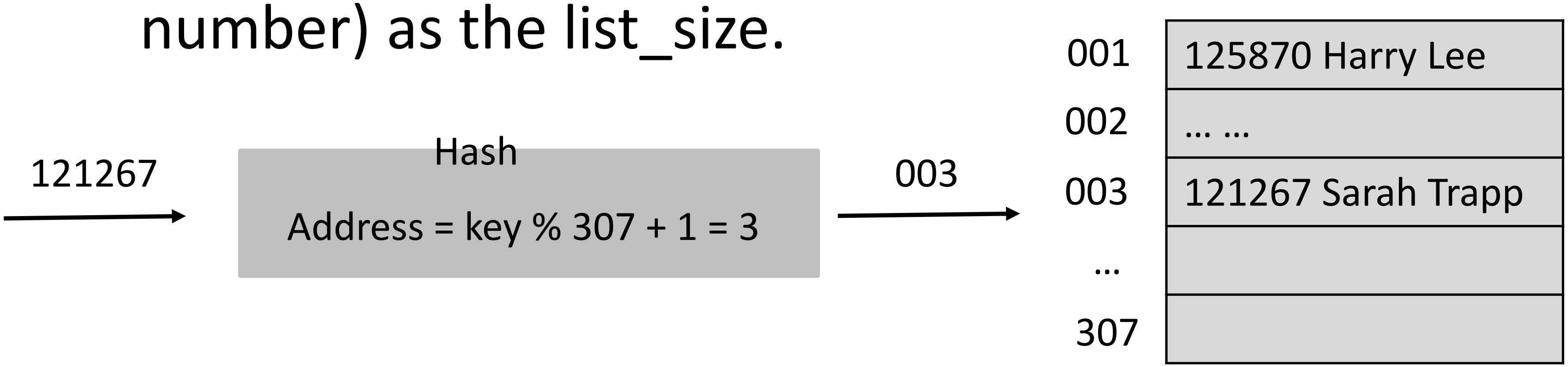
所有图片均来自网络

Direct hashing

- Key = address!
- Pros and Cons
  - – No synonyms or collisions
  - – The data file need space to containing a record for every possible key.
    - Considering: 100 employee with social security number as key (9 digits), result in data file of 999, 999, 999 records. (waste of data space!)
- How about map a large key space to a small data space?

Modulo division hashing

- Address = key % list_size + 1, where list_size is the number of records in the data file.
- List_size is preferred to be prime number!
- Considering the employee case:
  - key space 9 digits: 999, 999, 999
  - Planning for 300 employee, then use 307 (prime number) as the list_size.


121267

Hash

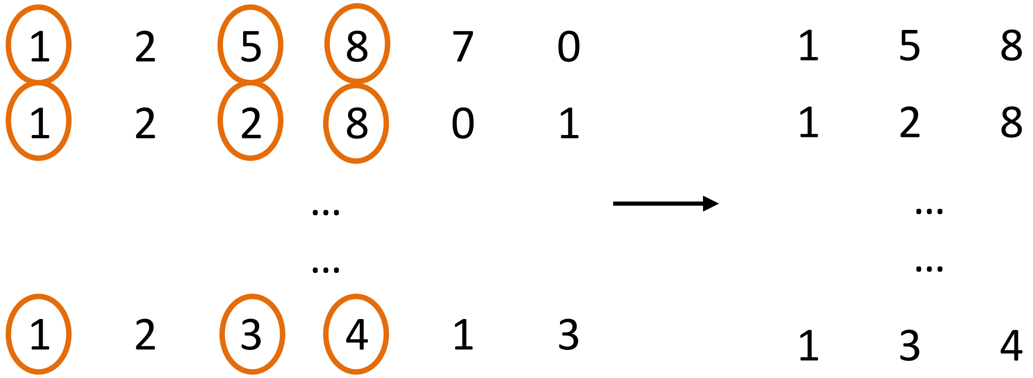
Address = key % 307 + 1 = 3

003

001	125870 Harry Lee
002	… …
003	121267 Sarah Trapp
…	
307	

Digit extraction hashing

- Selected digits are extracted from the key and used as address.



1 2 5 8 7 0 1 5 8

1 2 2 8 0 1 1 2 8

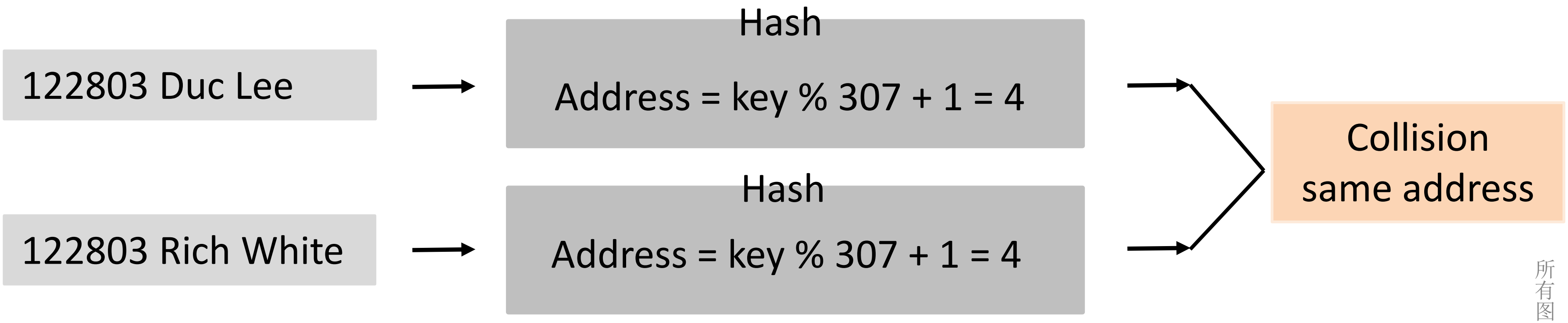
… …

… …

1 2 3 4 1 3 1 3 4

Collision

- Mapping a large key space to a small data space would result in collision!
- Collision: a set of keys are mapped to the same address.



122803 Duc Lee

Hash

Address = key % 307 + 1 = 4

122803 Rich White

Hash

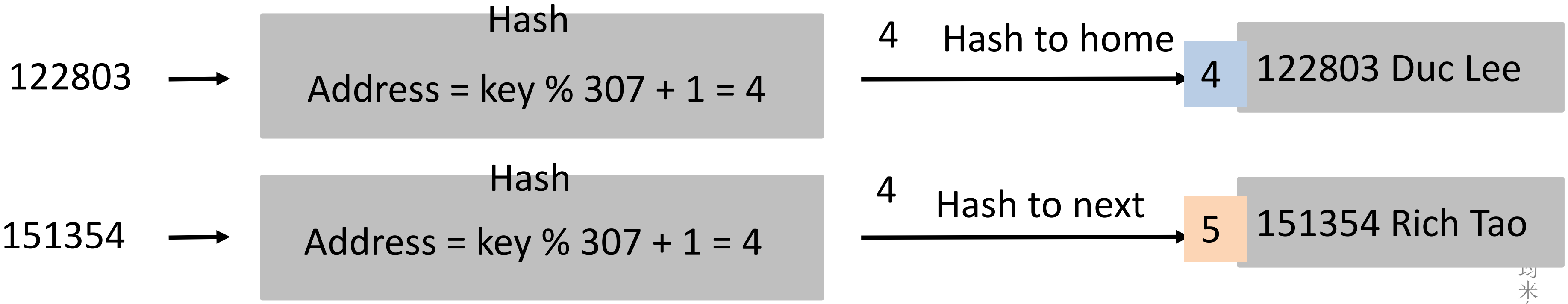
Address = key % 307 + 1 = 4

Collision
same address

所有图片均来自网络

Collision resolution

- Open addressing resolution:
  - When a collision occurs, the prime area addresses are searched for an open or unoccupied record where the new data can be placed.
  - increases the probability of future collisions

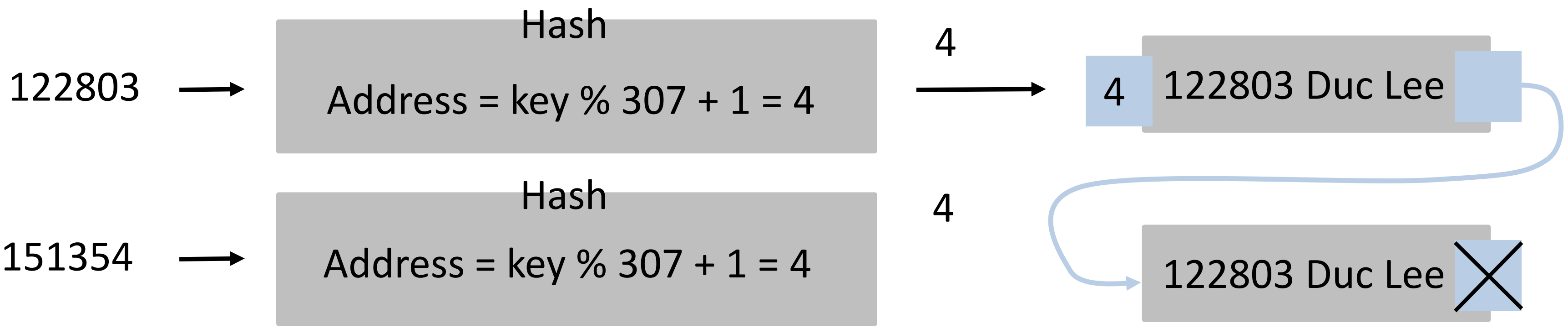


122803 → Hash
Address = key % 307 + 1 = 4
4 Hash to home → 4 | 122803 Duc Lee

151354 → Hash
Address = key % 307 + 1 = 4
4 Hash to next → 5 | 151354 Rich Tao

Collision resolution

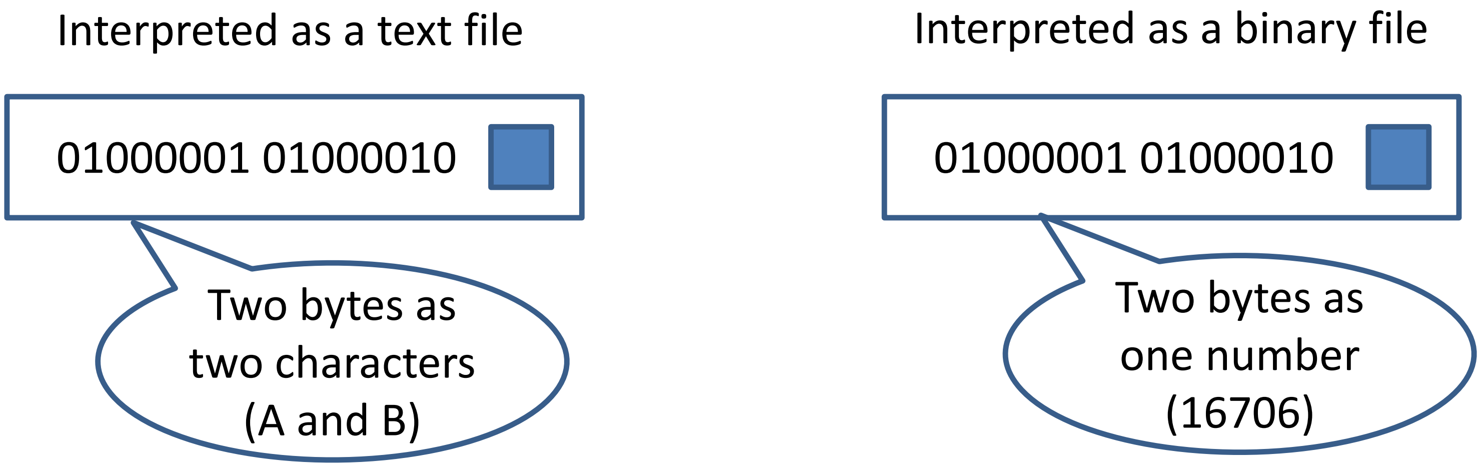
- Linked list resolution:
  - The first record is store in the home address, but it contains a pointer to the second record, and so on.



所有图片均来自网络

Text versus Binary

- A file is a collection of records.
- A record is a sequence of bits.
- How you interpret bit patterns is critically important! ~ to be meaningfull


Interpreted as a text file

01000001 01000010

Two bytes as two characters (A and B)

Interpreted as a binary file

01000001 01000010

Two bytes as one number (16706)

所有图片均来自网络

Text versus Binary

- A text file is a file of characters.
  - ASCII code or others encoding
- A binary file is a collection of data stored in the internal format of the computer.
  - Records can be an integer, a floating-point number, a character, or any other structure data (expect a file)

所有图片均来自网络

File Manager

- File is a logical concept for storing info in storage devices, such as Disk.
- File manager is responsible for controlling access to the files.
  - Access permission, regarding to a process
  - Supervising the creation, deletion and modification of files
  - Supervising the storage of files
  - And so on

所有图片均来自网络

User interface

- A program that accepts requests from users and interprets them for the rest of the operating system.
- Text-based interface: UNIX/LINUX shell
- Graphical User Interface: Windows XP …

所有图片均来自网络