Chapter Goals

- Describe the two main responsibilities of an operating system
- Define memory and process management
- Explain how timesharing creates the virtual machine illusion
- Explain the relationship between logical and physical addresses
- Compare and contrast memory management techniques

Chapter Goals

- Distinguish between fixed and dynamic partitions
- Define and apply partition selection algorithms
- Explain the stages and transitions of the process life cycle
- Explain the processing of various CPU scheduling algorithms

Software Categories

Application software

Software written to address specific needs—to solve problems in the real world

System software

Software that manages a computer system at a fundamental level

Can you name examples of each?

Roles of an Operating System

Operating system

System software that

– **manages** computer resources, such as memory and input/output devices

– **provides** an interface through which a human can interact with the computer

– **allows** an application program to interact with these other system resources

Roles of an Operating System

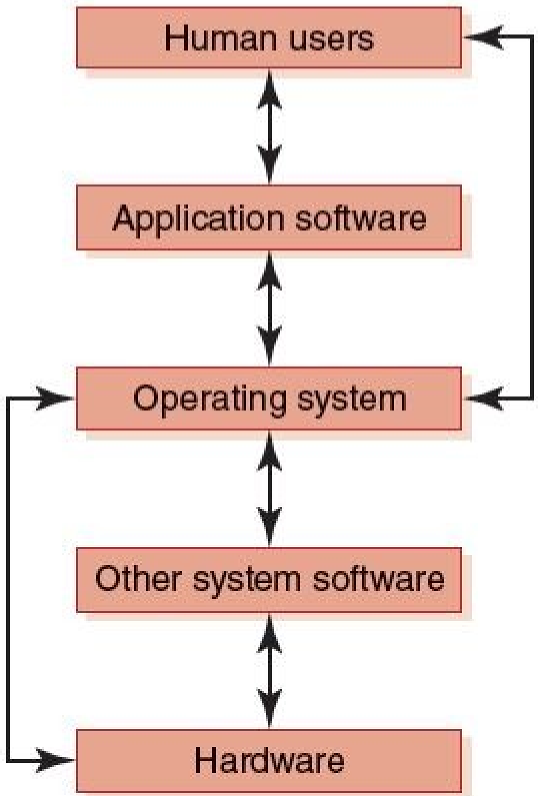


FIGURE 10.1 An operating system interacts with many aspects of a computer system.

What operating systems have you used?

Roles of an Operating System

Booting

Hardware is wired to initially load a small set of system instructions stored in ROM

These instructions load a larger set of instructions from hard disk

Dual boot - Multiboot

Roles of an 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

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

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

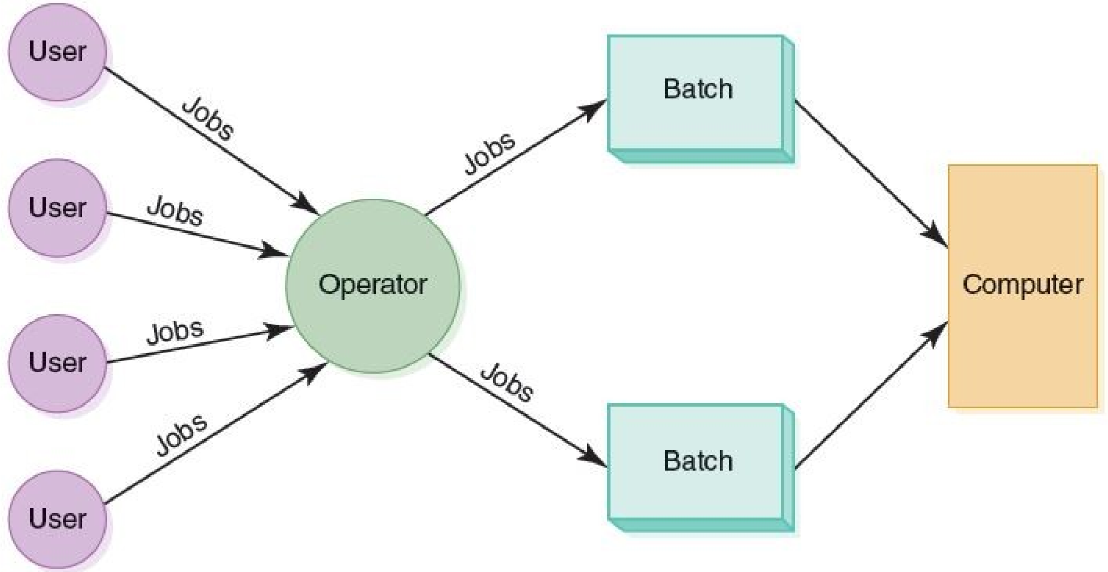


FIGURE 10.3 In early systems, human operators would organize jobs into batches

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

As computer speed increased, the human operator became the bottleneck

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

Memory Management

Operating systems must employ techniques to

- Track where and how a program resides in memory
- 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

Memory Management

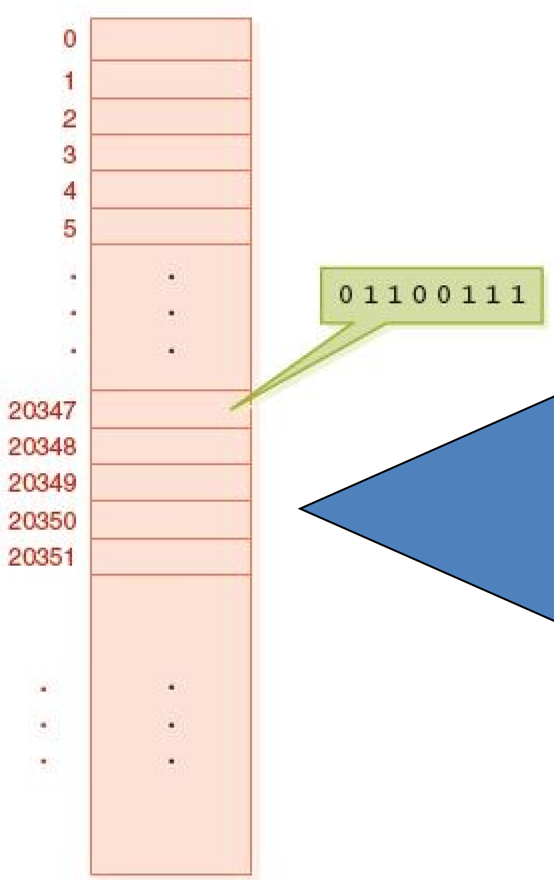


FIGURE 10.4 Memory is a continuous set of bits referenced by specific addresses

Program 1:
sum is assigned memory location 23, a location relative to Program 1

OS must map sum (relative location 23) to a specific physical address

Logical address for sum (23) is bound to a physical address in memory before the program runs

Single Contiguous MM

FIGURE 10.5 Main memory divided into two sections

There are only two programs in memory

- The operating system
- The application program

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

Single Contiguous MM

In concrete terms:

A logical address is simply an integer value relative to the starting point of the program

A physical address is a logical address added to the starting location of the program in main memory

Advantages – Disadvantages ? - Flexibility

Single Contiguous MM

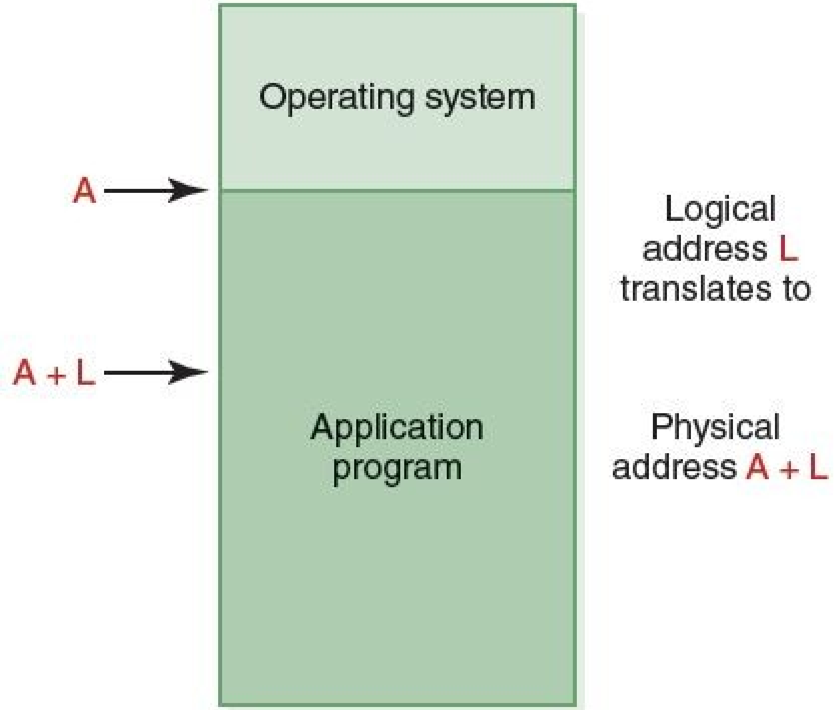


FIGURE 10.6 Binding a logical address to a physical address

If A is location 100, and the application program is Program 1, then sum is stored at location 123.

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** 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

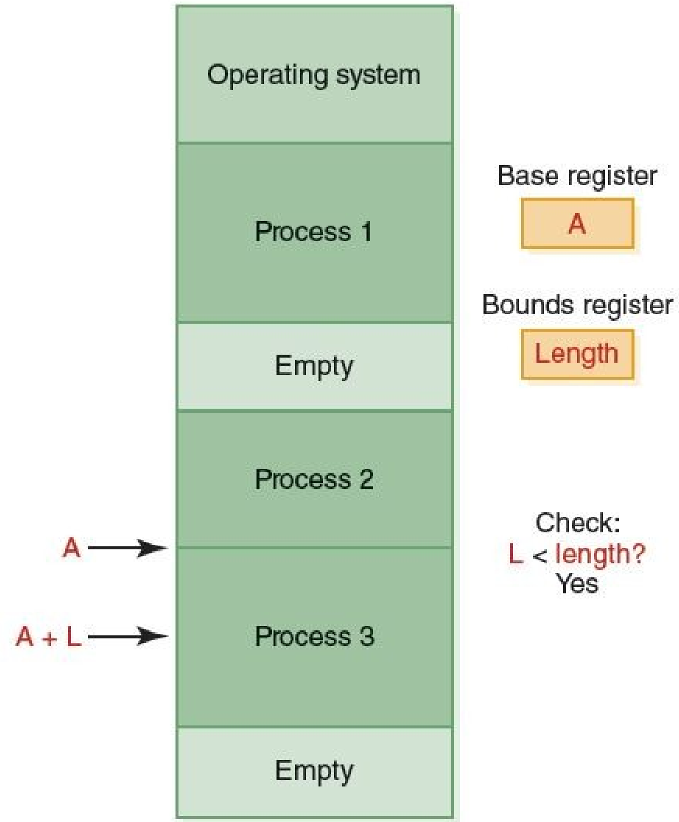
Base register

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?

FIGURE 10.7 Address resolution in partition memory management

Partition Selection Algorithms

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?

Partition Selection Algorithms

A: 1000
B: 700
C: 750
D: 1500
E: 300
F: 350

Requests come in for blocks of the following sizes:
1000, 25, 780, 1600, and 325

What block will be assigned to each request if the

- *first-fit algorithm is used?*
- *best-fit algorithm is used?*
- *worst-fit algorithm is used?*

(Treat each request as an independent event)

Process Management

Process management

The act of managing the use of the CPU by individual processes

Recall that a process is a program in execution

What stages does a process go through?

Process Management

The Process States

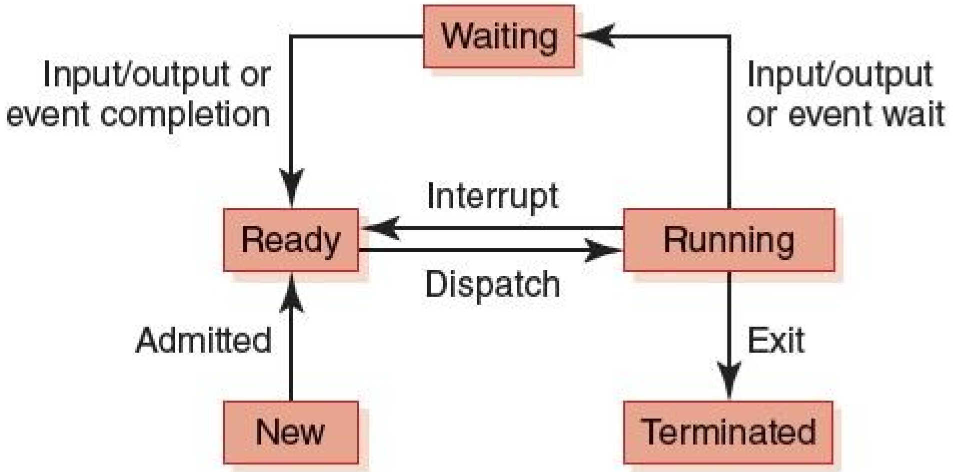


FIGURE 10.9 The process life cycle

What can cause a process to move to the Waiting state?

Process Management

Process control block (PCB)

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

- current value of the program counter
- values of all CPU registers for the process
- base and bound register values accounting information

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

Process Management

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:

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

CPU Scheduling

CPU Scheduling

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

- Many processes may be in the ready state
- 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

The currently executing process gives up the CPU voluntarily

Preemptive scheduling

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

Turnaround time

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

CPU Scheduling Algorithms

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

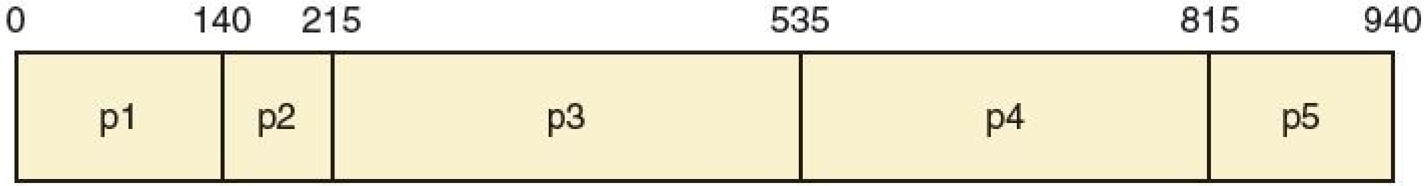
Round Robin

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

First-Come, First-Served

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

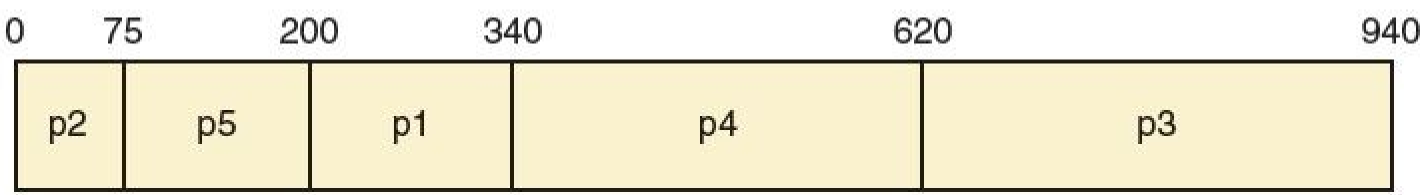
What is the average turn-around time?



Shortest Job Next

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

What is the average turn-around time?



Round Robin

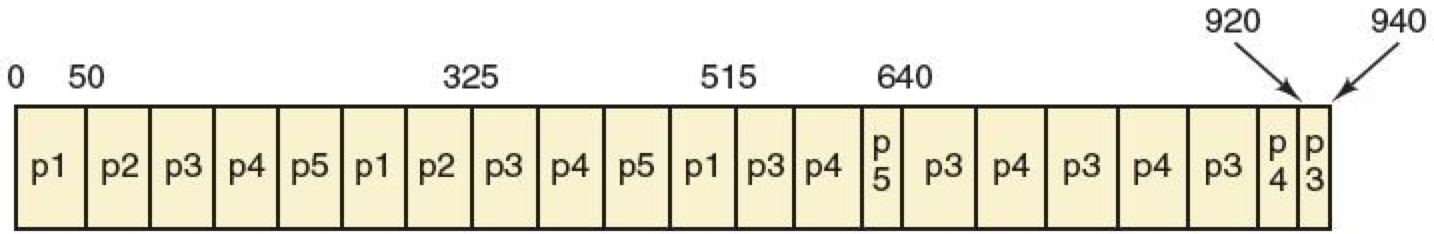
Every process is treated the same!

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

Round Robin

Suppose the time slice is 50



What is the average turnaround time?

CPU Scheduling Algorithms

Are these scheduling algorithms preemptive or non-preemptive? Explain

First-Come, First-Served?

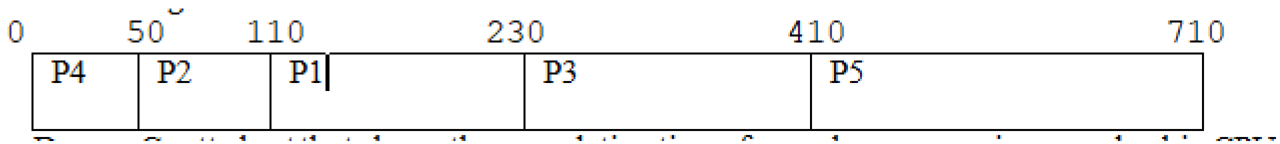
Shortest Job Next?

Round Robin?

Activity 1:

Use the following table of processes and service time. Draw Gantt charts that show turnaround time for each process using <u>SJN (shortest job next) and RR (round robin)</u> CPU Scheduling. Also find average turnaround time for each.

Process	P1	P2	P3	P4	P5
Service time	120	60	180	50	300



302

Activity 2:

Apply <u>First-Fit and Best-Fit</u> Algorithm on the following jobs given along the memory set available (fixed partitions). Fill Job Number column for your answer.

	First-Fit (Job No)	Best-Fit (Job No)	
8 KB	J1 = 4KB	J3 = 6KB	
100 KB	J2 = 30KB	J6 = 100KB	
1024 KB	J3 = 6KB	J7 = 7	
56 KB	J4 = 9KB	J5 = 10KB	
2 KB	J8 = 2KB	J8=2KB	
48 KB	J5 = 10KB	J2 = 30KB	
2048 KB	J6 = 100KB		
4 KB	J10 = 1KB	J1 = 4KB	
8 KB	J9 = 5KB	J9 = 5KB	
12 KB		J4 = 9KB	
56 KB		J10 = 1KB	

MEMORY MAP

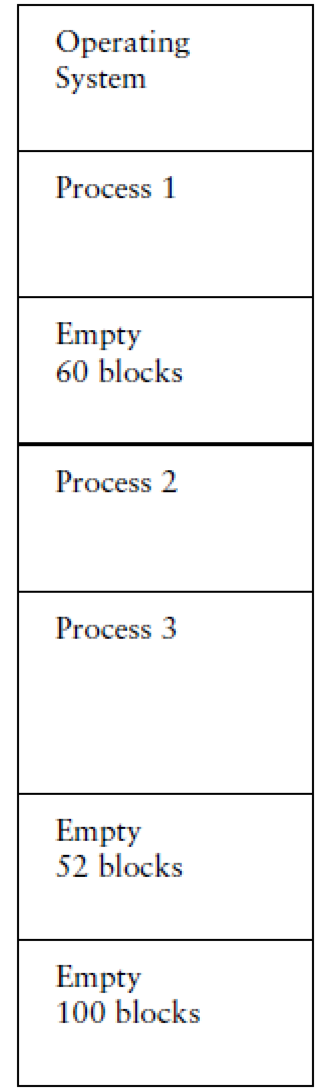
	First-Fit (Job No)	Best-Fit (Job No)
8 KB		
100 KB		
1024 KB		
56 KB		
2 KB		
48 KB		
2048 KB		
4 KB		
8 KB		
12 KB		
56 KB		

Jobs

J1	J2	J3	J4	J5	J6	J7	J8	J9	J10
4 KB	30 KB	6 KB	9 KB	10 KB	100 KB	200 KB	2 KB	5 KB	1 KB

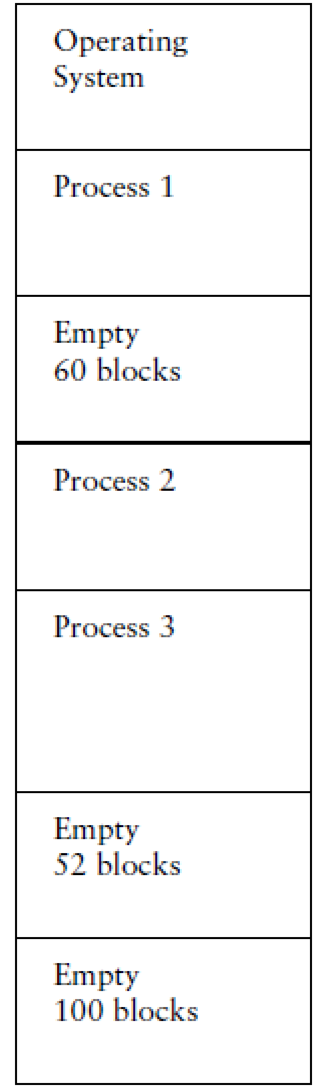
Activity 3:

- If the partitions are fixed and a new job arrive
  requiring 52 blocks of main memory, show m
  after using each of the following partition sel
  approaches:
- **a.** first fit (60 Block)
- **b.** best fit (52 Blocks)
- **c.** worst fit (100 Blocks)



Activity 4:

- If the partitions are dynamic and a new job ar requiring 52 blocks of main memory, show m after using each of the following partition sel approaches:
- **a.** first fit
- **b.** best fit
- **c.** worst fit



Activity 5:

- Why shouldn’t we use worst-fit partition selection in a fixed-partition memory management scheme?