

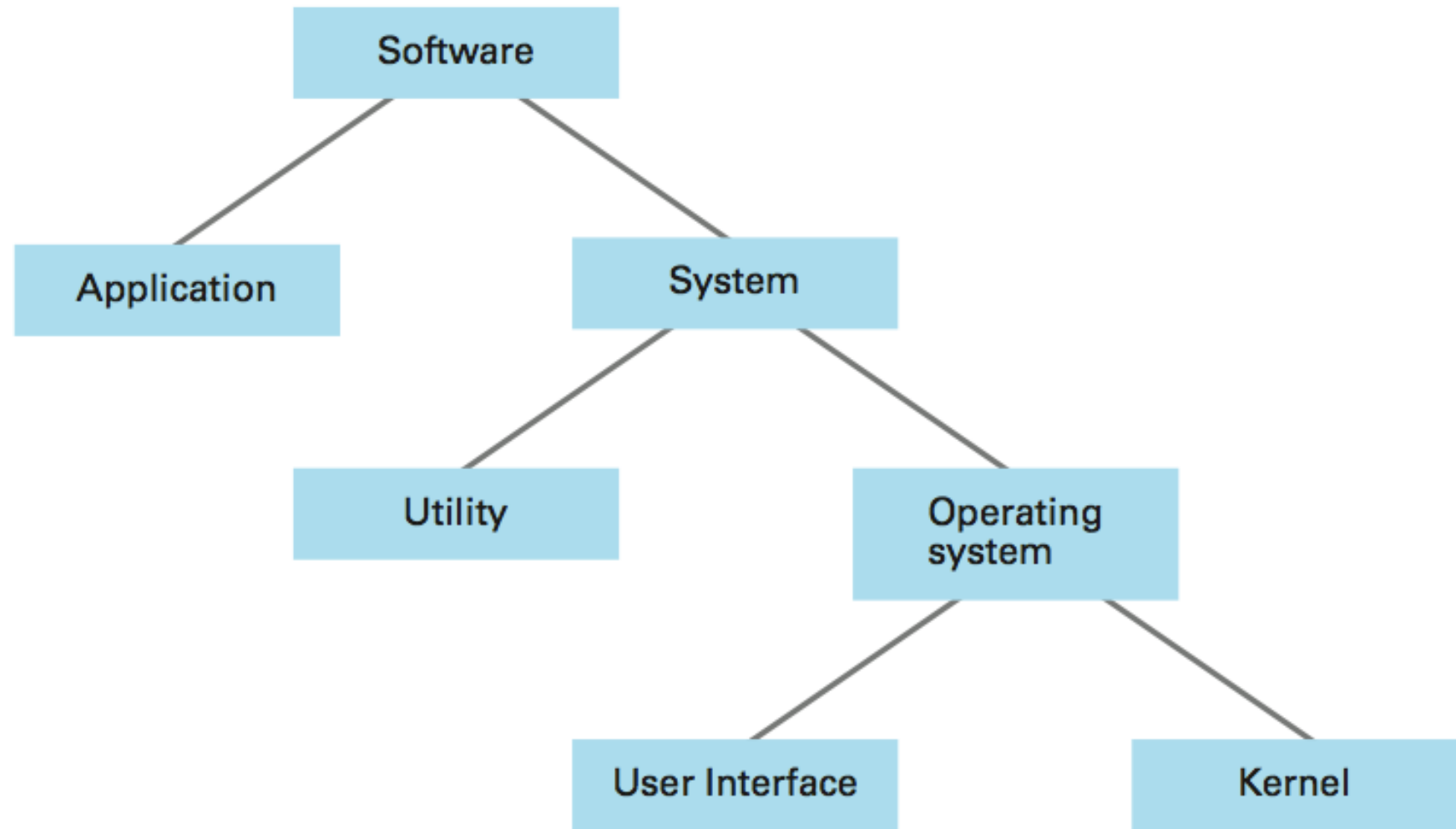


# Lecture 5: Operating System Layer

Lecturer: Phuong L. Vo

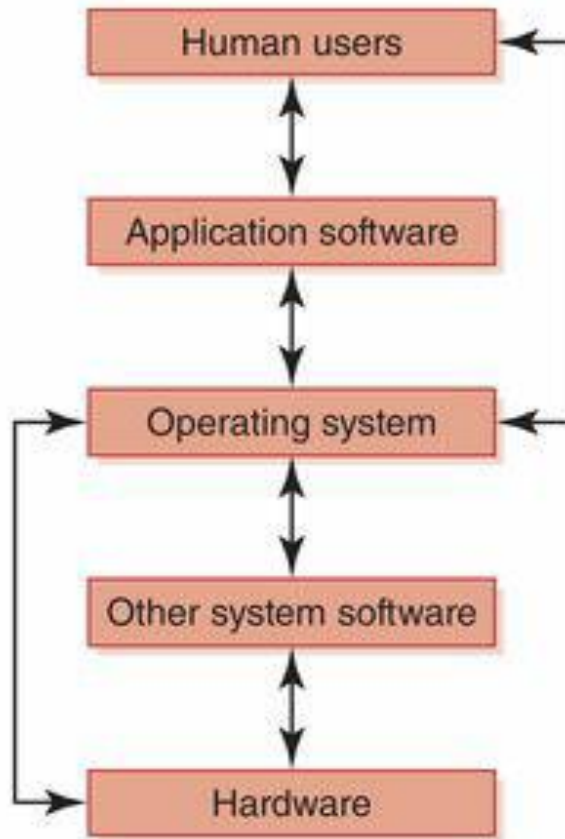
# 1. Operating system

## Software clasification



*This figure is from “CS: An overview”.*

# Roles of an Operating System



- ❖ **Application software:** Programs that help us solve real-world problems. Give some examples?
- ❖ **System software:** Programs that manage a computer system and interact with Hardware. E.g., ...
- ❖ **Operating system:** System software that manages computer resources and provides an interface for system interaction

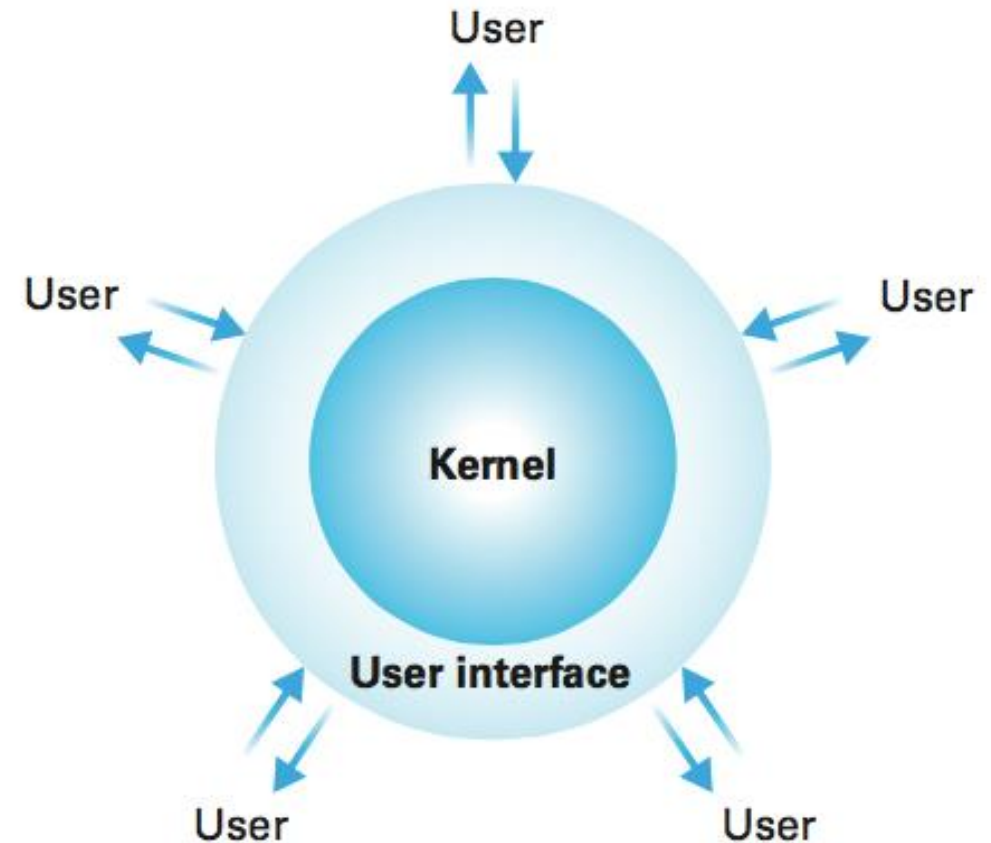
# Operating System Components

❖ **User Interface:** Communicates with users

- ❑ Text based (Shell), e.g. Bourne shell, Bash shell, C shell, and Korn shell
- ❑ Graphical user interface (GUI), e.g. GNOME, KDE , Unity, etc.

❖ **Kernel:** Performs basic functions

- ❑ File manager
- ❑ Device drivers
- ❑ Memory manager
- ❑ Scheduler and dispatcher

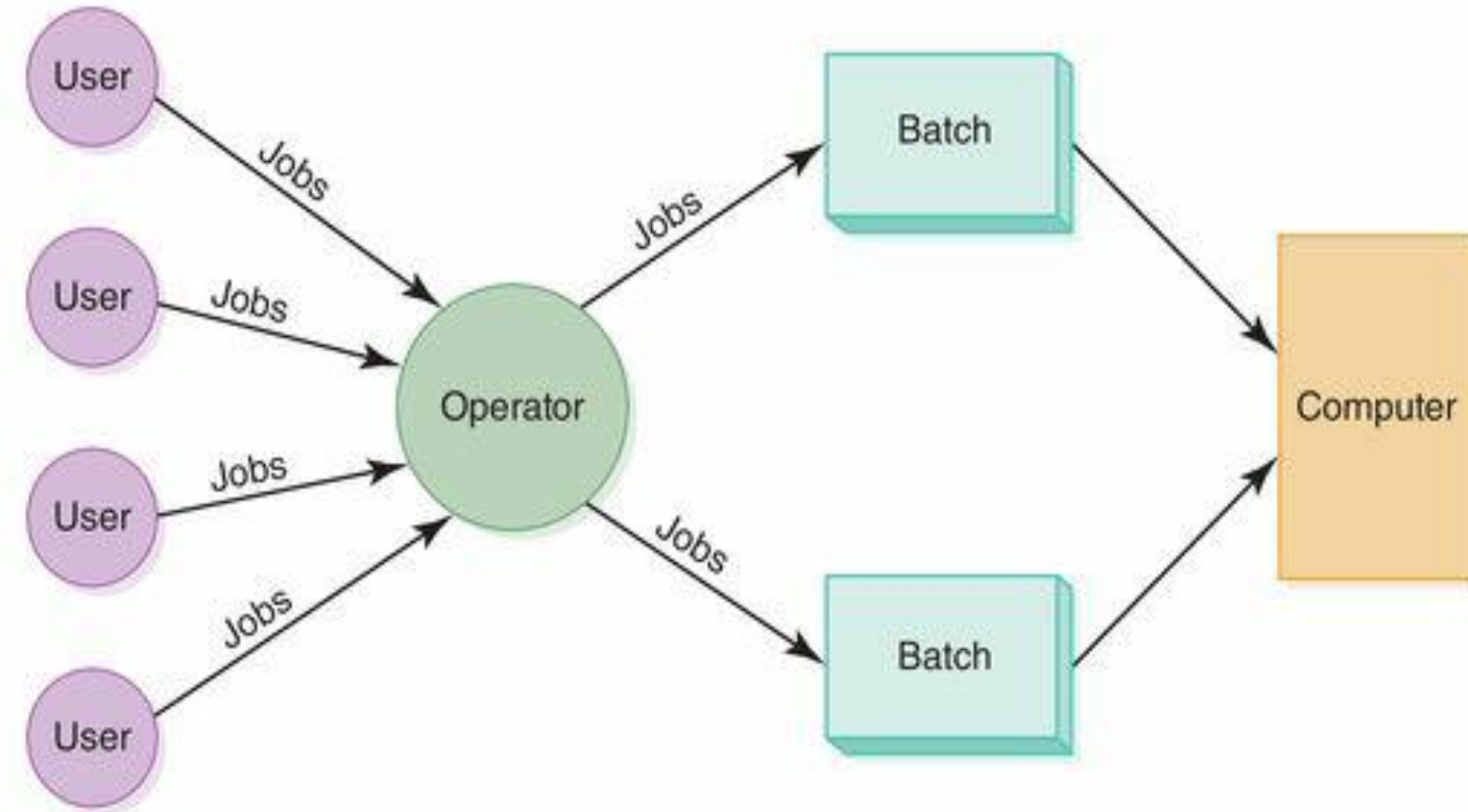




# Memory, Process, and CPU Management

- ❖ **Multiprogramming** The technique of keeping multiple programs in main memory at the same time, competing for the CPU
- ❖ **Memory management** The act of keeping track of how and where programs are loaded in main memory.
- ❖ **Process** The dynamic representation of a program during execution
- ❖ **Process management** The act of keeping track of information for active processes
- ❖ **CPU scheduling** The act of determining which process in memory is given access to the CPU so that it may execute

# Batch Processing



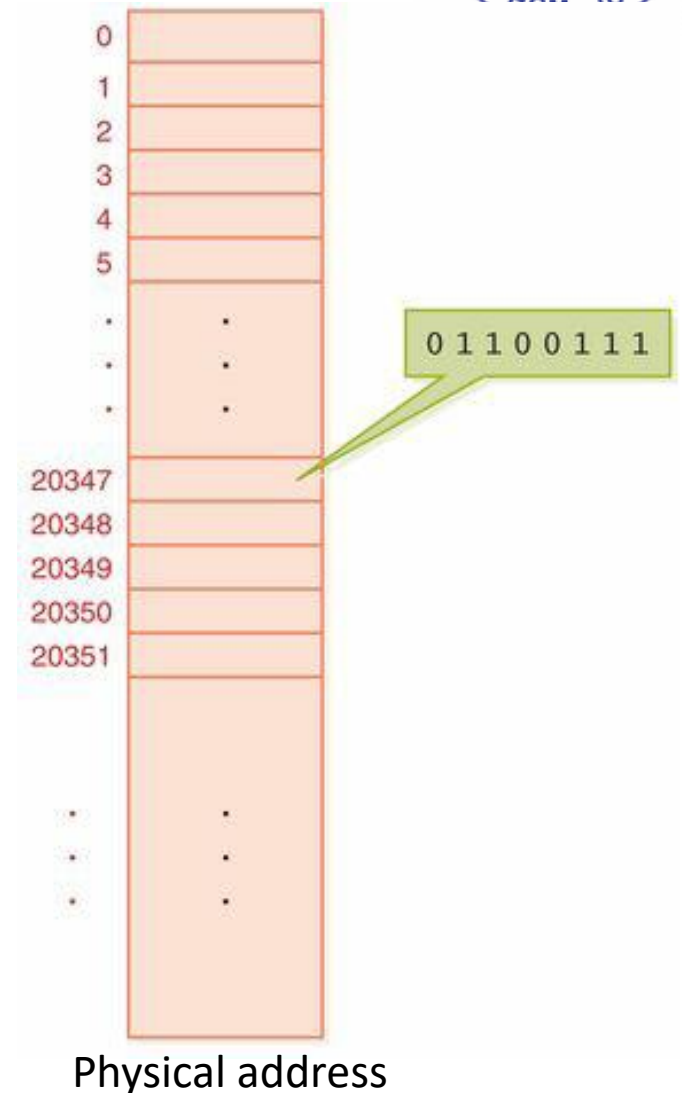
In early systems, human operators would organize jobs into batches

In modern computing system:

- ❖ .bat in Microsoft Windows
- ❖ Shell script in linux

## 2. Memory Management

- ❖ **Physical address** An actual address in the main memory device
- ❖ **Logical address**: A reference to a stored value relative to the program making the reference
- ❖ **Address binding** The mapping from a logical address to a physical address

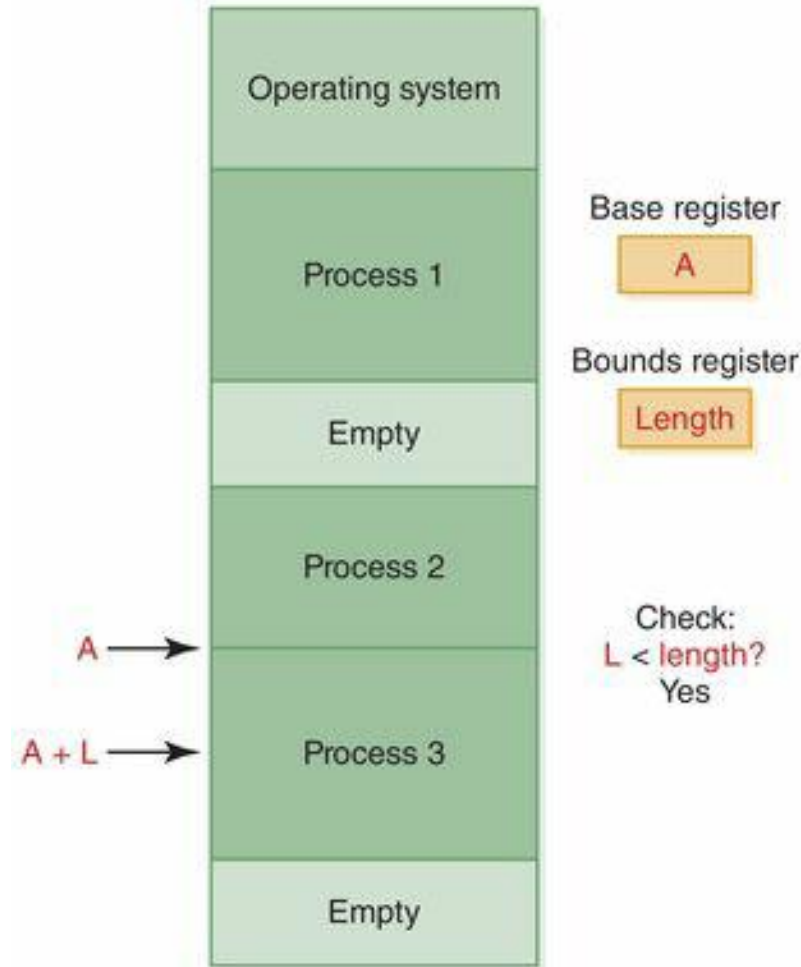




# Memory management



Single contiguous MM



Partition MM

**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

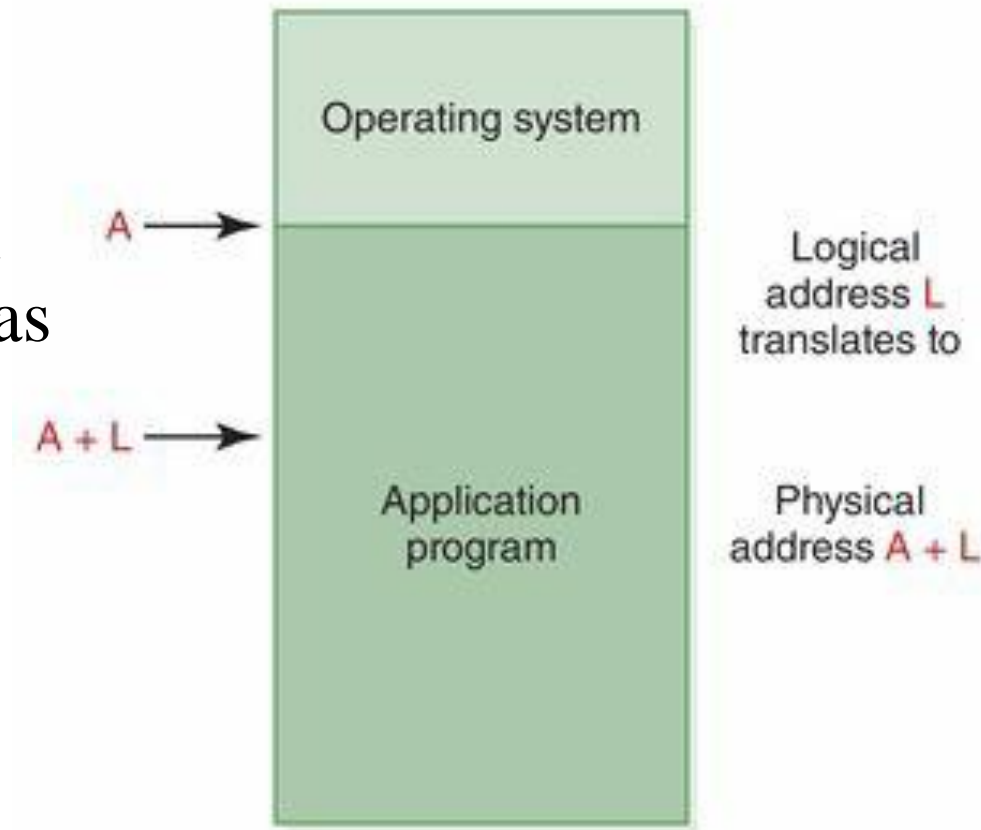
**Memory**

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

Paged MM

# Single Contiguous Memory Management

- ❖ The entire application program is loaded into one continuous area of memory
- ❖ Only one program other than the operating system can be processed at one time
- ❖ A logical address is simply an integer value relative to the starting point of the program as if the program loaded at location 0 of main memory.
- ❖ to produce a physical address, we add a logical address to the starting address of the program in physical main memory
- ❖ Simple but not multiprocessing



# Partition Memory Management

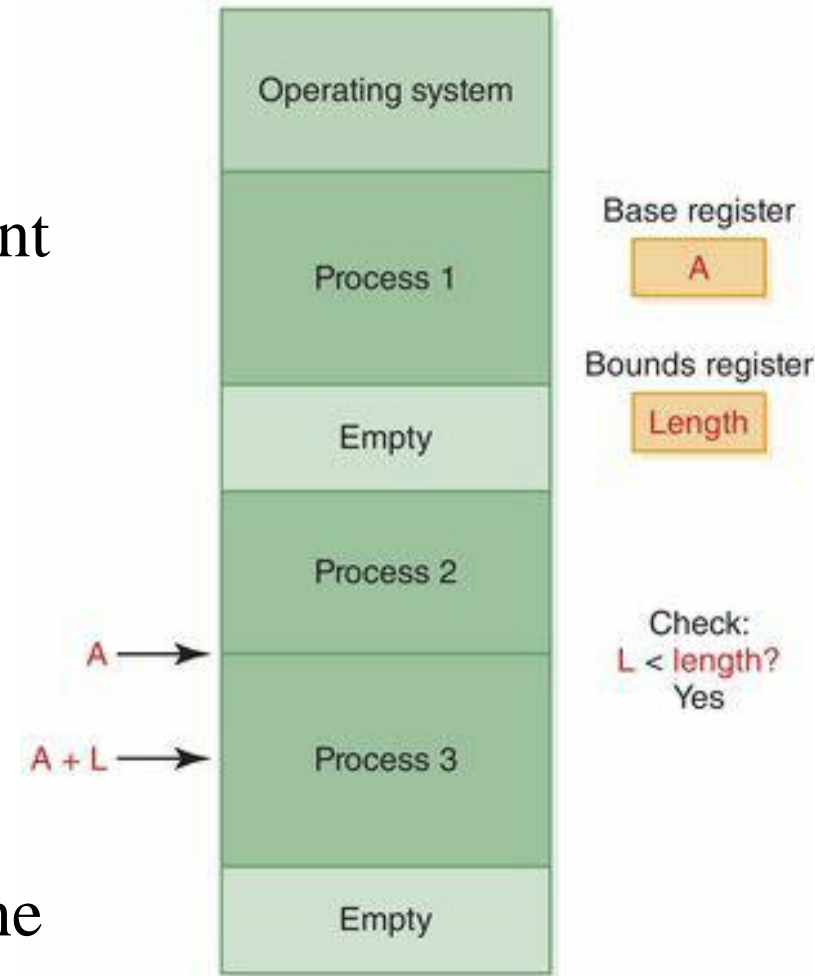
more than one application program in memory at the same time, sharing memory space and CPU time

❖ **Fixed-partition technique:** The memory management technique in which memory is divided into a specific number of partitions into which programs are loaded.

❖ **Dynamic-partition technique:** The memory management technique in which memory is divided into partitions as needed to accommodate programs.

❖ **Base register** A register that holds the beginning address of the current partition

❖ **Bounds register** A register that holds the length of the current partition



# Paged Memory Management

- ❖ **Paged memory technique:** A memory management 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
- ❖ **Page-map table (PMT):** The table used by the operating system to keep track of page/frame relationships

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

Page	Frame
0	10
1	18
2	1
3	11

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



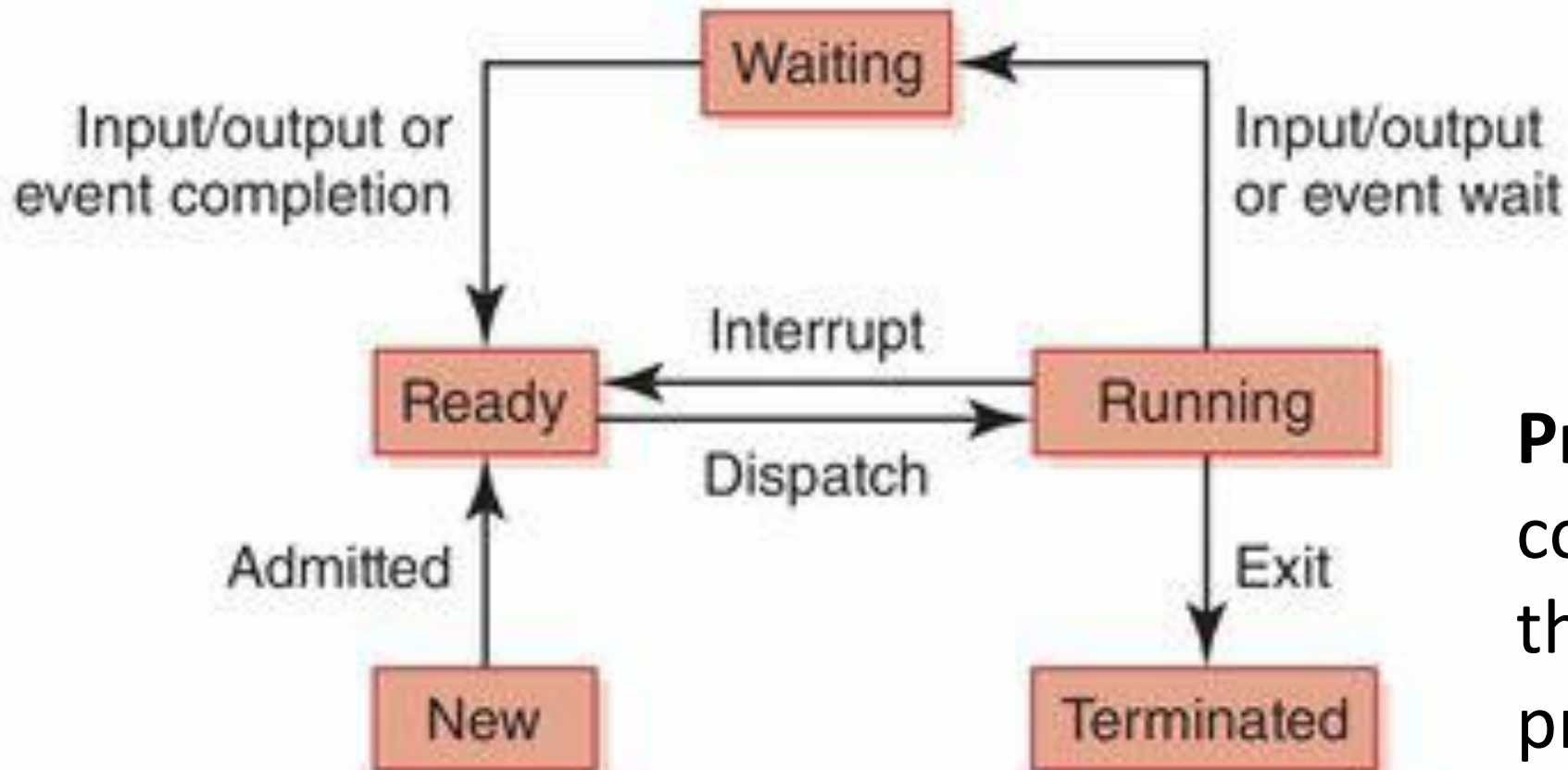
# Exercise

❖ P49 a-b page 356

❖ P51 a-b page 356

❖ P58 page 357

# Process Management



**Process states** The conceptual stages through which a process moves as it is managed by the operating system

# Process management - Process states

- ❖ In the new state, a process is being created.
- ❖ A process that has no barriers to its execution is in the ready state.
- ❖ A process in the running state is currently being executed by the CPU. Its instructions are being processed in the fetch–execute cycle.
- ❖ A process in the waiting state is currently waiting for resources (other than the CPU).
- ❖ A process in the terminated state has completed its execution and is no longer active.



# CPU Scheduling

- ❖ CPU scheduling is the act of determining which process in the ready state should be moved to the running state.
- ❖ Non-preemptive vs. preemptive scheduling

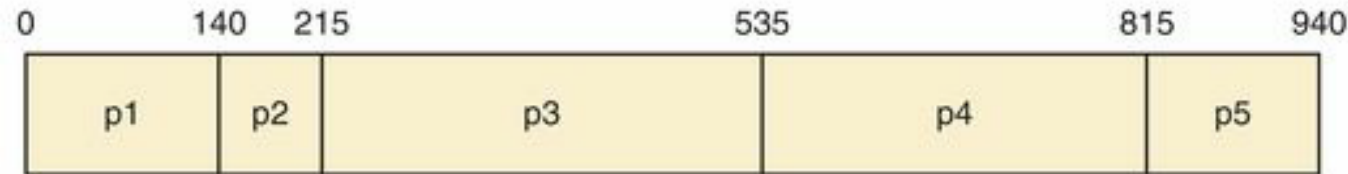
Process	Service Time
p1	140
p2	75
p3	320
p4	280
p5	125



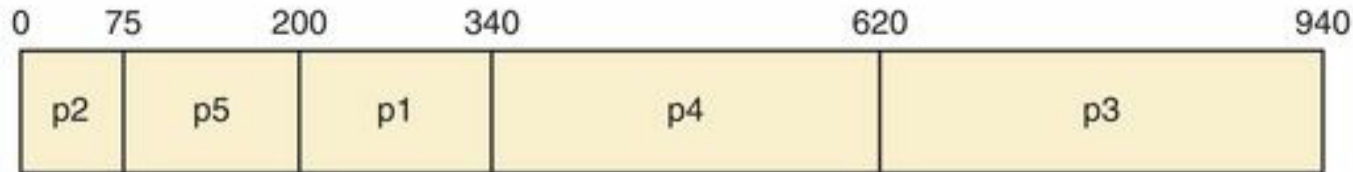
# CPU Scheduling

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

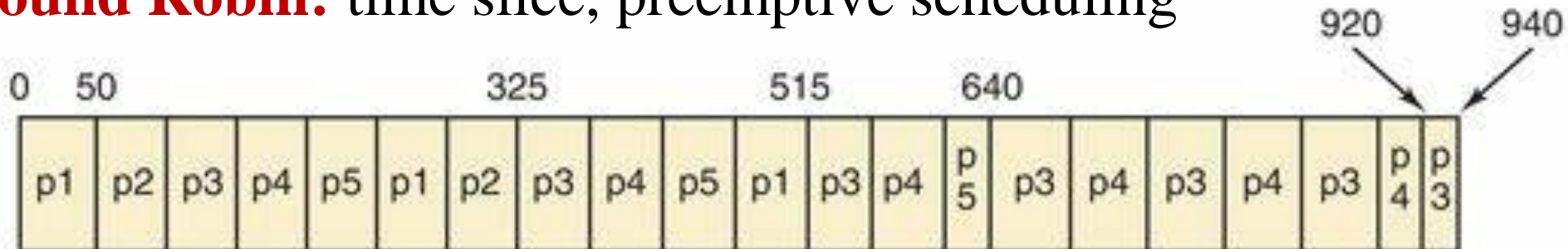
❖ **First Come, First Served (FCFS):** non-preemptive



❖ **Shortest Job Next:** non-preemptive



❖ **Round Robin:** time slice, preemptive scheduling



# Exercises

❖ Prob. 69, page 357 (FCFS, shortest job first)

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



# Quiz

- ❖ Exercise 1-18
- ❖ Prob. 49b, 51b-c, 57, 59 page 356-357
- ❖ Prob. 71, page 357

## 2. File system directory



# File Systems

- ❖ **File** A named collection of data, used for organizing secondary memory.
- ❖ **File system** The operating system's logical view of the files it manages.
- ❖ **Directory** A named group of files.

Extensions	File type
txt	text data file
mp3, au, wav	audio file
gif, tiff, jpg	image file
doc, wp3	word processing document
java, c, cpp	program source files



# File Operations

- ❖ Create a file
- ❖ Delete a file
- ❖ Open a file
- ❖ Close a file
- ❖ Read data from a file
- ❖ Write data to a file
- ❖ Reposition the current file pointer in a file
- ❖ Append data to the end of a file
- ❖ Truncate a file (delete its contents)
- ❖ Rename a file
- ❖ Copy a file

create / delete /  
open / close / write  
data / append data  
... to a file in **Python**

# File protection

	Read	Write/Delete	Execute
Owner	Yes	Yes	No
Group	Yes	No	No
World	No	No	No





# Directory Trees

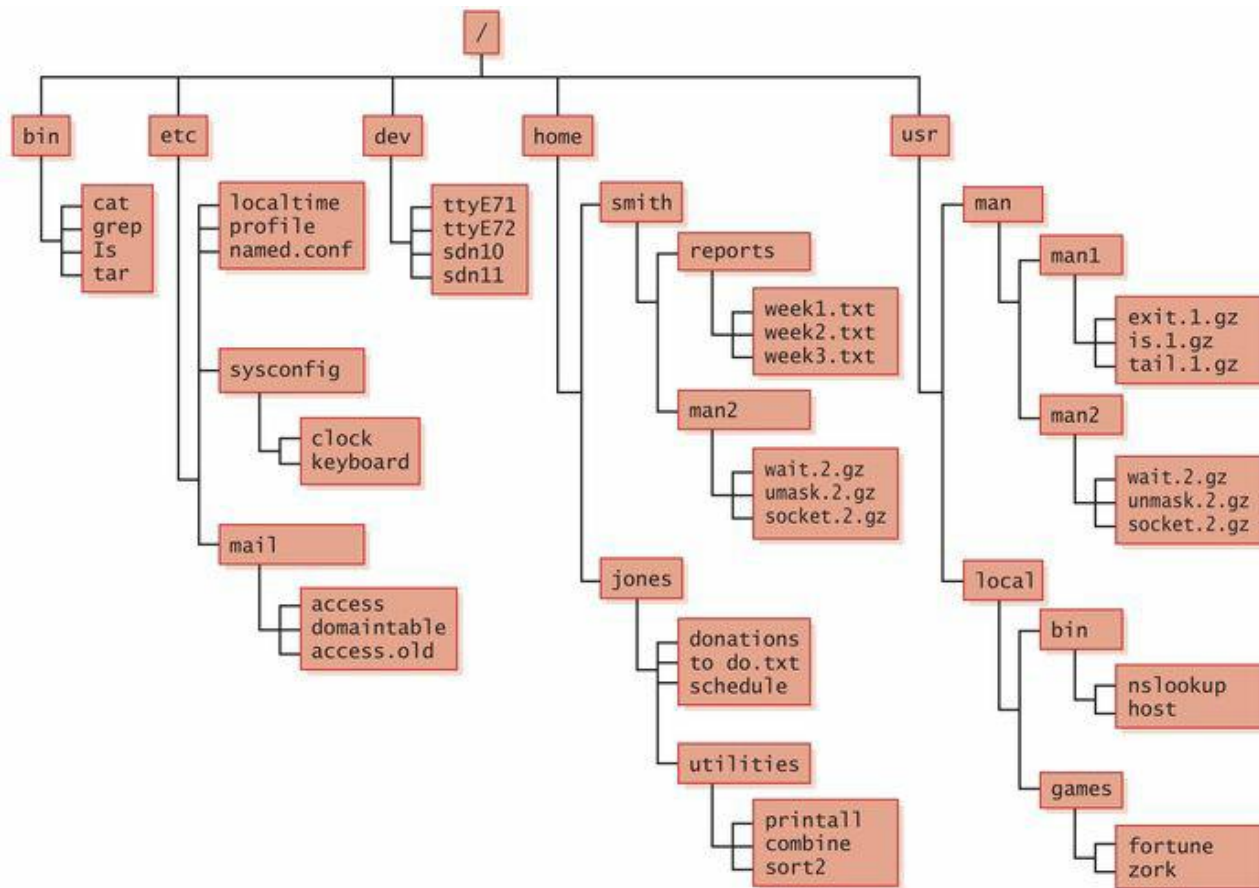
- ❖ **Directory tree** A structure showing the nested directory organization of the file system
- ❖ **Root directory** The topmost directory, in which all others are contained
- ❖ **Path Names:**

absolute

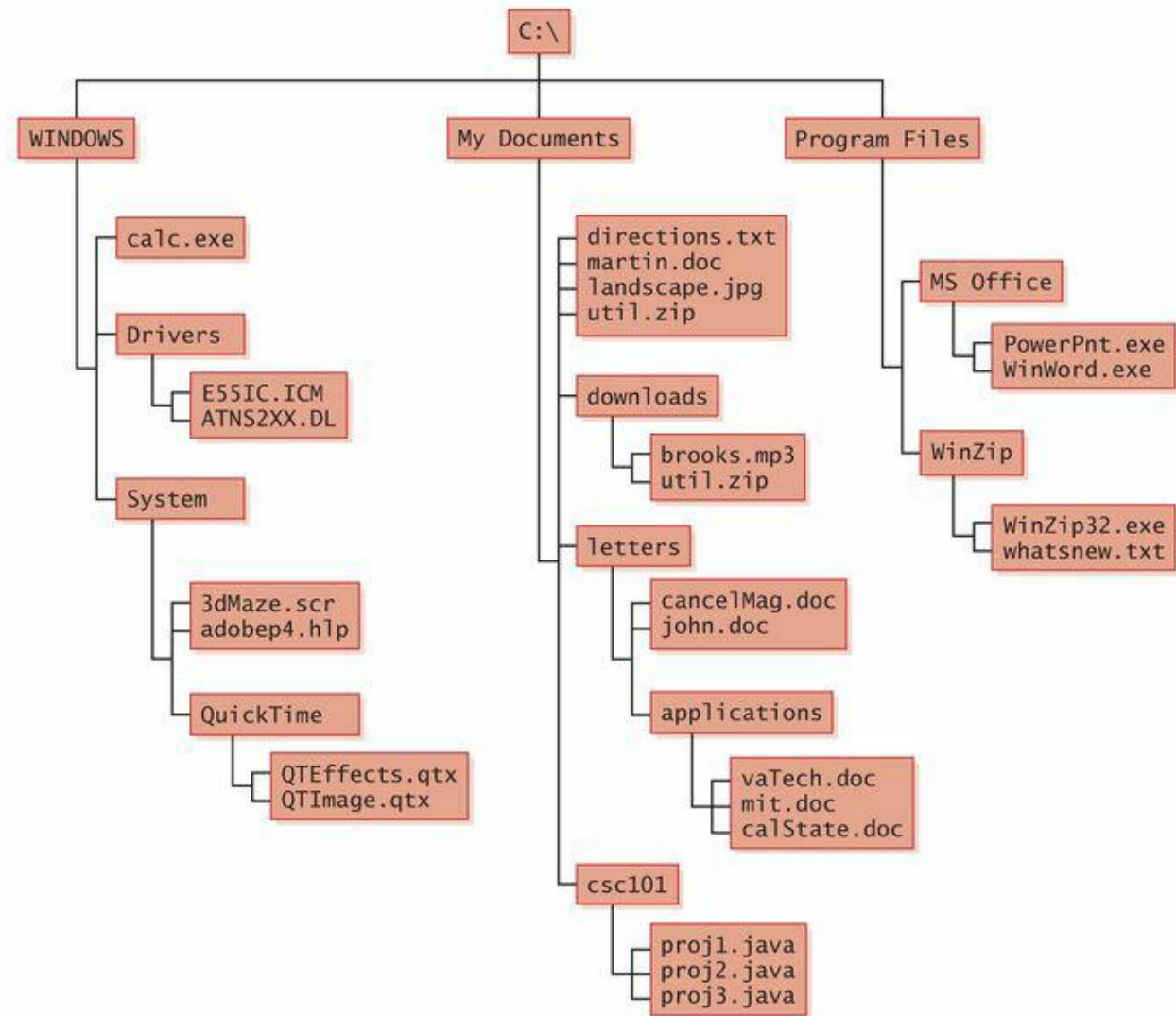
```
C:\Program Files\MS Office\WinWord.exe  
C:\My Documents\letters\applications\vaTech.doc  
C:\Windows\System\QuickTime
```

relative

```
..\landscape.jpg  
..\csc111\proj2.java  
..\..\WINDOWS\Drivers\E55IC.ICM  
..\..\Program Files\WinZip
```



A **Unix** directory tree



A **Windows** directory tree

# Summary

- ❖ An operating system is the part of the system software that manages resources on a computer.
- ❖ Multiprogramming is the technique for keeping multiple programs in memory at the same time, contending for time on the CPU.
- ❖ Batch processing organizes jobs into batches that use the same or similar resources.
- ❖ An operating system must manage memory to control and monitor where processes are loaded into main memory.
- ❖ CPU scheduling algorithms determine which process gets priority to use the CPU next.
- ❖ A file system defines the way our secondary memory is organized.
- ❖ Directories are used to organize files on disk.

Q&A