CMPS1134
Fundamentals of Computing

Operating Systems

Computer Science: An Overview
Eleventh Edition
J. Glenn Brookshear
Chapter 3

Chapter 3: Operating Systems

- ☐ The History of Operating Systems
- ☐ Operating System Architecture
- ☐ Coordinating the Machine's Activities
- ☐ Handling Competition Among Processes
- ☐ Security

Functions of Operating Systems

An **Operating system** is the software that:

☐ Oversee operation of computer

☐ Store and retrieve files

☐ Schedule programs for execution

☐ Coordinate the execution of programs

Evolution of Shared Computing

☐ **Batch processing**

☐ **Interactive processing**

☐ **Time-sharing/Multitasking**

☐ **Multiprocessor machines**

Figure 3.1 **Batch processing**

- **40-50s**: Machines occupied entire rooms.
- No VDU – switches, punched cards, etc.
- **Jobs** are collected in a single <u>batch</u> (by operator) and then executed without further interaction with the user.
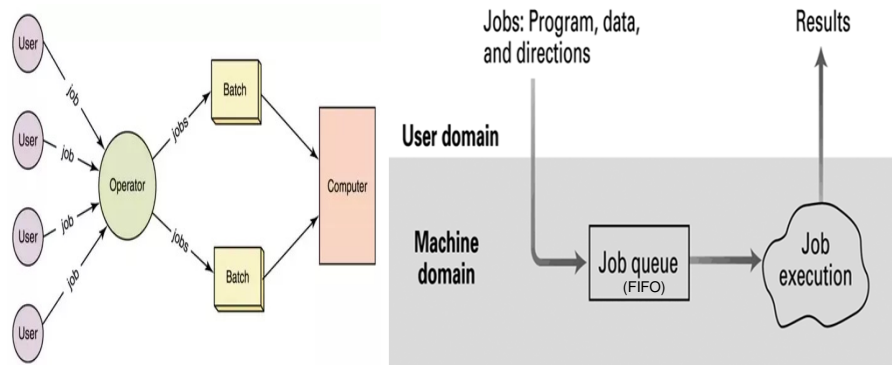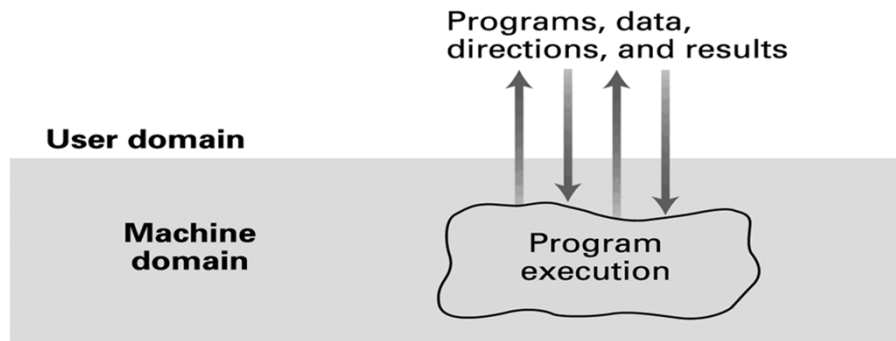
Figure 3.2 **Interactive processing**

- **60-70s**: **Real-time processing** (user actions occur in real-time)
- Dialog with users through remote terminals (typewriter/ printer)
- Worked well for one user but the constraint of executing only one job at a time presented obstacles for multiple users.
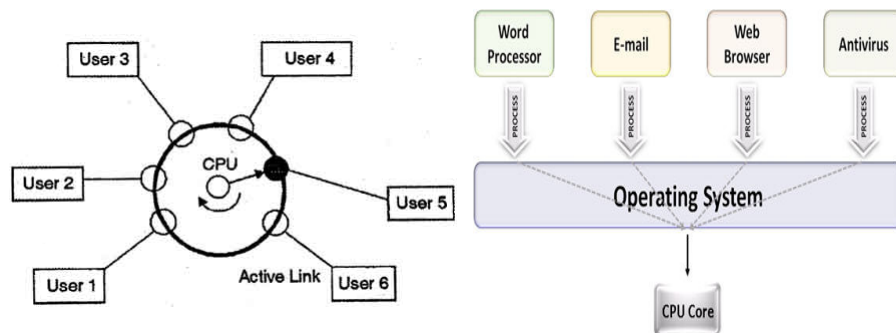
Time Sharing/ Multitasking

- **70s onward**: **Time-sharing** refers to multiple users sharing access to a common computer simultaneously
- One method of implementation – **Multiprogramming**
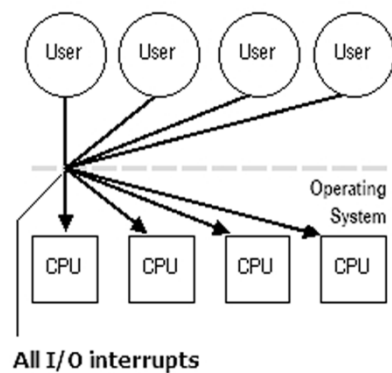- **Multi-tasking** is one user executing multiple tasks simultaneously

Multiprocessor Machines

Asymmetric Multiprocessing

Symmetric Multiprocessing
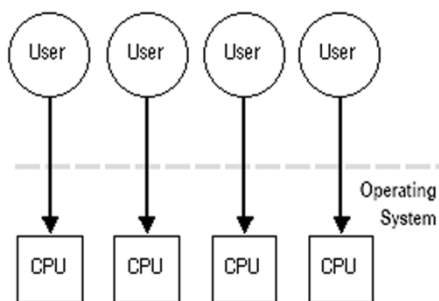
All I/O interrupts

Types of Software

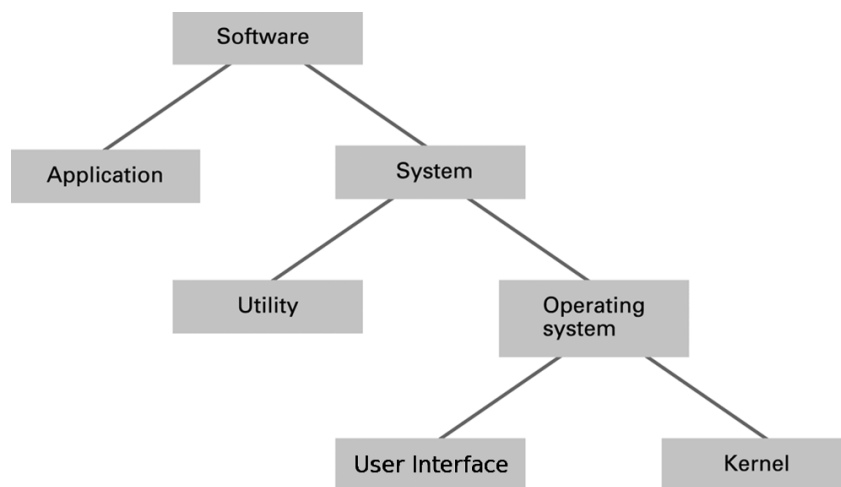- ☐ **Application software**
  - ■ Performs specific tasks for users

- ☐ **System software**
  - ■ Provides infrastructure for application software
  - ■ Consists of operating system and utility software

Figure 3.3 **Software classification**

Operating System Components

- □ **User Interface:** Communicates with users
  - ■ Text based (**Shell**)
  - ■ Graphical user interface (**GUI**)
- □ **Kernel:** Performs basic required functions
  - ■ **File manager**
  - ■ **Device drivers**
  - ■ **Memory manager**
  - ■ **Processor Manager**
    - □ Scheduler
    - □ Dispatcher
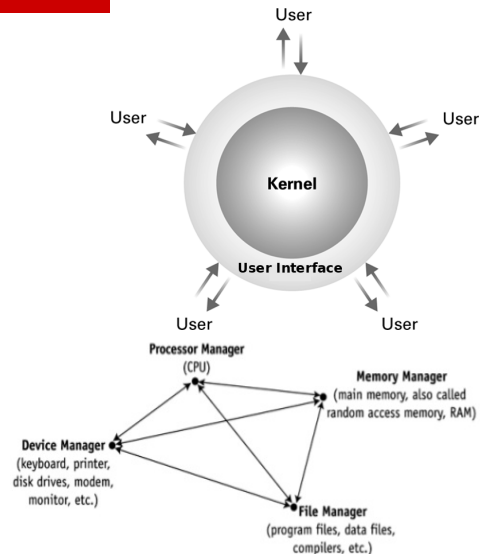
11

File Manager

Coordinates the use of the machine's mass storage facilities.

- □ **Directory** (or **Folder**):
  A user-created bundle of files and other directories (subdirectories)

- □ **Directory Path:**
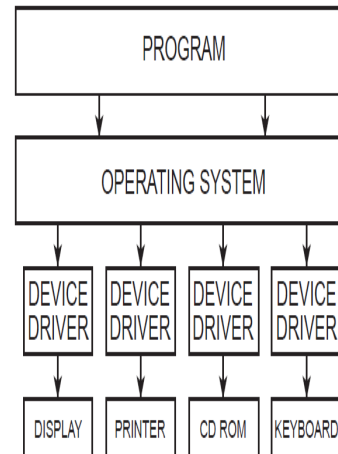 A sequence of directories within directories
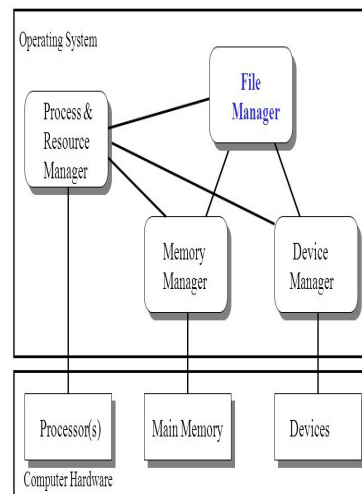
12

Device Drivers

Software units that communicate with:

- the device controllers
- or directly with the peripheral devices

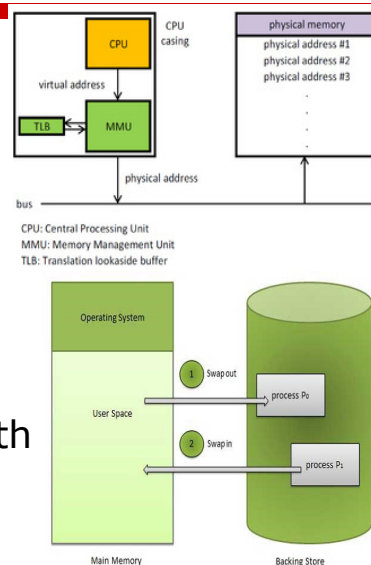to carry out operations on the peripheral devices attached to the machine.

PROGRAM

OPERATING SYSTEM

DEVICE DRIVER | DEVICE DRIVER | DEVICE DRIVER | DEVICE DRIVER

DISPLAY | PRINTER | CD ROM | KEYBOARD

Memory Manager

- ☐ Allocates space in main memory

- ☐ May create the illusion that the machine has more memory than it actually does (**virtual memory**) by playing a "shell game" in which blocks of data (**pages**) are shifted back and forth between main memory and mass storage (**paging**)

CPU casing

CPU

virtual address

TLB ⇄ MMU

physical address

bus

physical memory
physical address #1
physical address #2
physical address #3

CPU: Central Processing Unit
MMU: Memory Management Unit
TLB: Translation lookaside buffer

Operating System

User Space

1 Swap out → process P₀

2 Swap in → process P₁
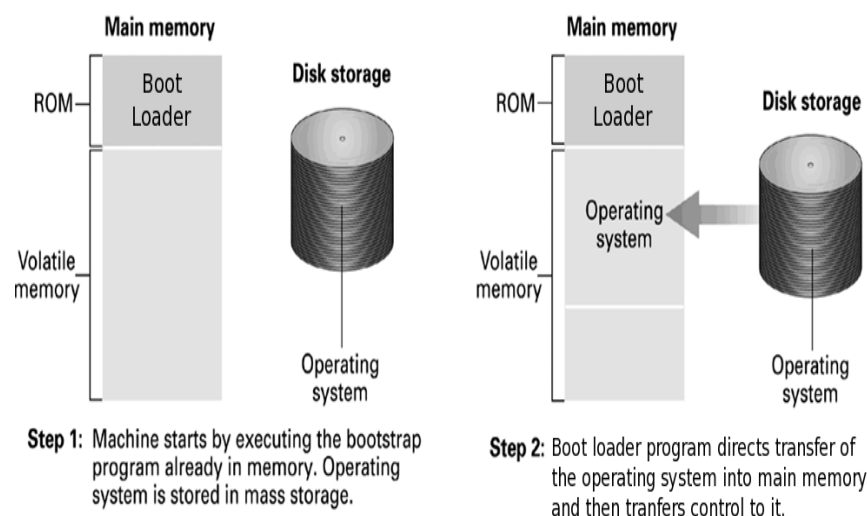
Main Memory   Backing Store

Getting it Started

- ☐ **Bootstrapping** (often shortened to **booting**) is the process that transfers the operating system from mass storage into main memory.

- ☐ **Boot loader:** Program in **read-only memory (ROM)** (example of firmware)
  - ■ Run by the CPU when power is turned on
  - ■ Transfers operating system from mass storage to main memory
  - ■ Executes jump to operating system

Figure 3.5 **The booting process**

Step 1: Machine starts by executing the bootstrap program already in memory. Operating system is stored in mass storage.

Step 2: Boot loader program directs transfer of the operating system into main memory and then tranfers control to it.

Processes

☐ **Process:** The activity of executing a program

☐ **Process State:** Current status of the activity

- Current position in the program (<u>Program counter</u>)
- Values in the CPU registers and related portions of main memory
- A snapshot of the machine at a particular time

Process Administration

☐ **Scheduler**: Adds new processes to the process table and removes completed processes from the process table

☐ **Dispatcher**: Controls the allocation of **time slices** (time segments of milli or microseconds) to the processes in the process table
  - ☐ The end of a time slice is signaled by an **interrupt**.
  - ☐ An **interrupt handler** describes how the dispatcher should respond to an interrupt signal

☐ The **Process Table** is a block of main memory maintained by the scheduler with entries for each process (assigned memory, priority, ready/waiting, etc.)

☐ A process is **ready** if it is in a state in which its progress can continue

☐ A process is **waiting** if its progress is currently delayed until some external event occurs

Figure 3.6 **Time-sharing between process A and process B**
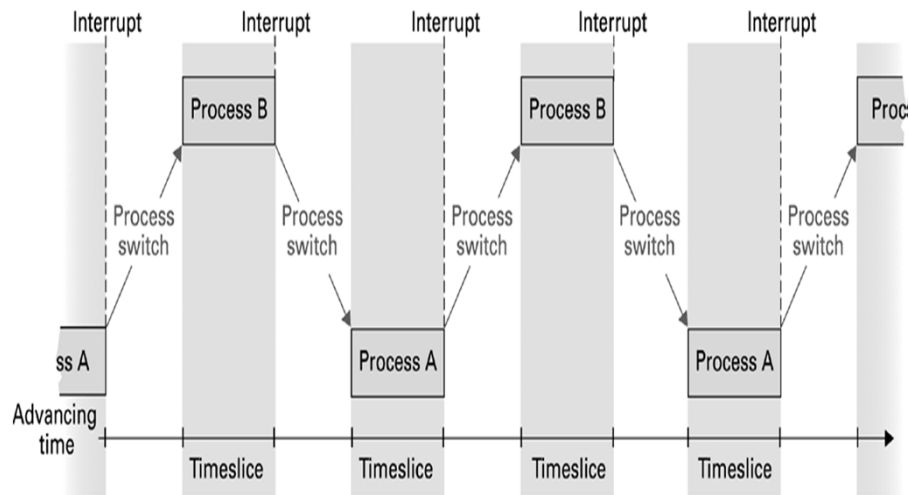
Handling Competition for Resources

☐ **Semaphore:** A "control flag" that controls access to a common resource by multiple processes

☐ **Critical Region:** A group of instructions that should be executed by only one process at a time

☐ **Mutual exclusion:** Requirement for proper implementation of a critical region

Deadlock

- ☐ Processes block each other from continuing because they are waiting for resources allocated to another
- ☐ Processes need to create new processes (**forking**) to complete but the <u>process table</u> is full
- ☐ Conditions required for deadlock
  1. Competition for non-sharable resources
  2. Resources are requested on a partial basis and processes return to request more
  3. An allocated resource can not be forcibly retrieved
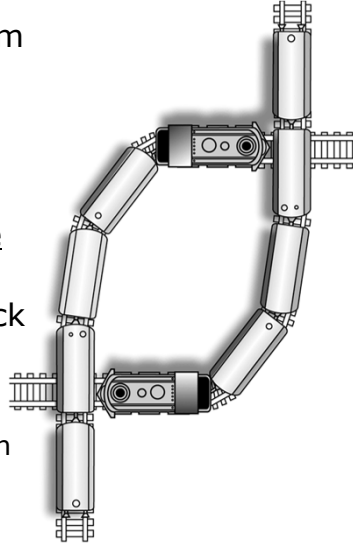
21

Security

- ☐ **Attacks from outside**
  - ■ <u>Problems</u>
    - ☐ Insecure passwords
    - ☐ Sniffing software
  - ■ <u>Counter measures</u>
    - ☐ Auditing software
- ☐ **Attacks from within**
  - ■ Problem: Unruly processes
  - ■ Counter measures: Control process activities via **privileged levels** and **privileged instructions**

22