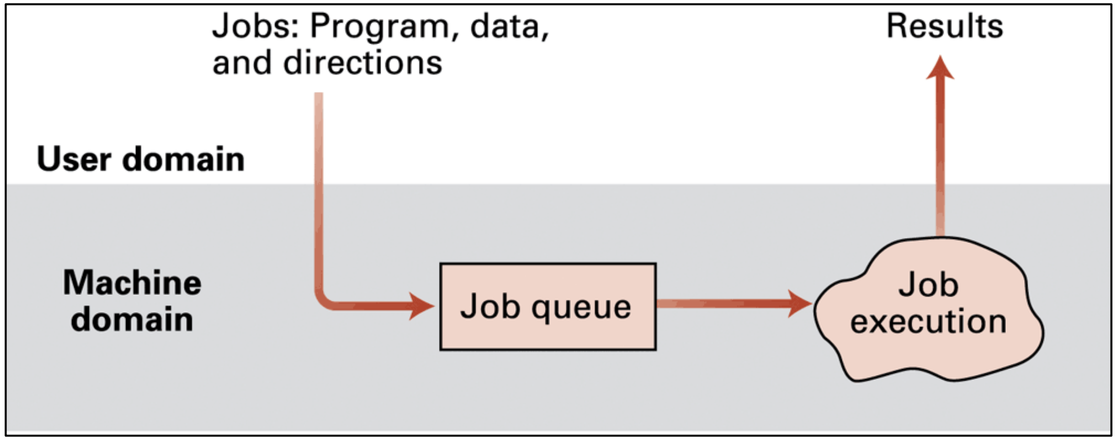
Operating Systems

National Chiao Tung University
Chun-Jen Tsai

3/23/2012

History of Operating Systems (1/3)

- ❑ No OS (Pre 1950's):
  - ■ Computer runs a program under direct operator control
- ❑ Batch processing systems (1950's – 1960's):
  - ■ The OS put user programs into a job queue in a first-in-first-out manner (FIFO)
  - ■ The OS select one job at a time to run under the computer until it finishes; then it selects the next job to run



History of Operating Systems (2/3)

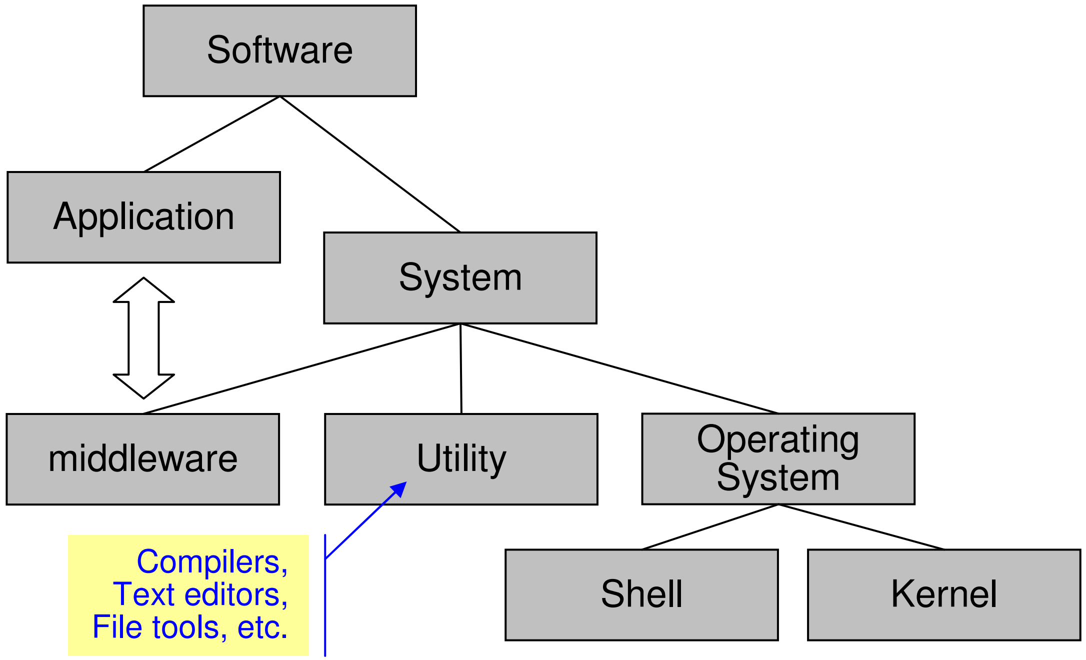
- ❑ Simple multi-tasking systems (1960’s – 1970’s)
  - ■ Several programs are arranged in the memory at the same time by the OS
  - ■ The OS runs one program until it has to do I/O (executed by some other small computers), then the OS selects another program to run
  - ■ When the first program finishes its I/O task, it will wait for the 2nd program finishes or tries to do I/O before it start running again

History of Operating Systems (3/3)

- ❑ Time-sharing systems (1970’s - present)
  - The OS divide the CPU computation time into intervals (a.k.a. time slices or time quantum)
  - Each program entered in the job queue run for one time slice at a time
  - At the end of the time slice, the program will be paused and another program will be selected for execution

Types of Software

- ❑ Application software: perform user tasks
- ❑ System software: perform tasks to control computers
- ❑ Middleware: system interface software for applications

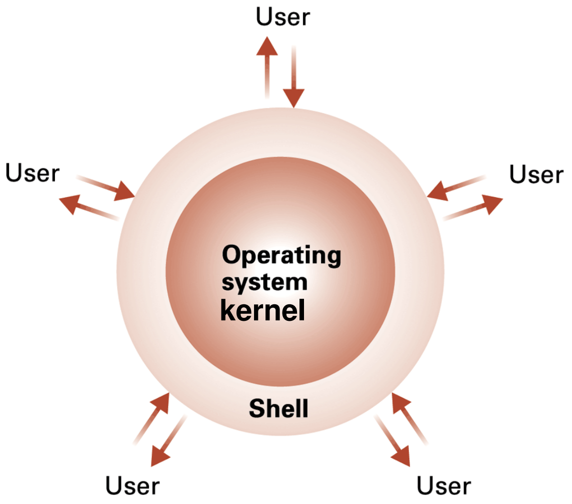


What is an Operating System (OS)?

- ❑ Main Functions of an OS:
  - ■ Schedule programs for execution
  - ■ Manage main memories
  - ■ Store and retrieve files
  - ■ Provide interfaces of input/output devices to programs
- ❑ Program development environment (libraries, compilers, linkers, debuggers, etc.) are often tied to the OS, especially in the early days of computers
  - ■ An OS provides an unified program model and standard libraries that makes debugging easier

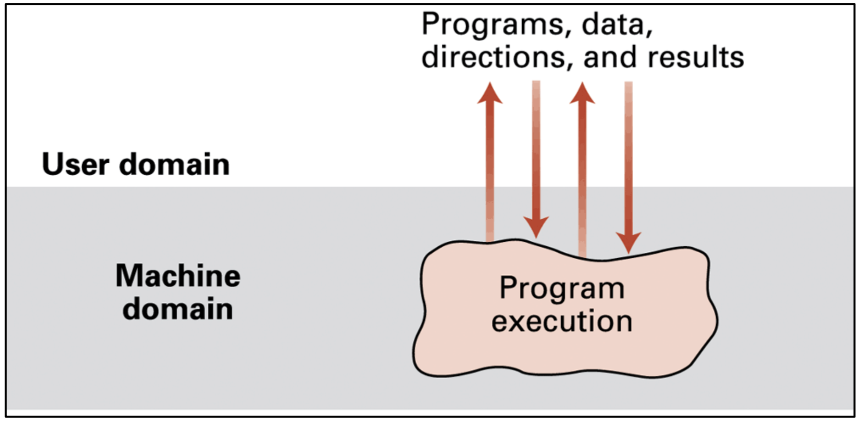
Components of an Operating System

- Shell: an interface between a user and the kernel of an OS
  - There are text-based and window-based shells
    - Command line interface: e.g., DOS prompt, Linux bash
    - Graphical user interface (GUI): e.g., window manager,
      GUIs are often called WIMP – Windows, Icons, Menus, and Pointers
  - Shell is not an essential part of an OS → Shells are replaceable
- Kernel: key components performing basic required functions
  - Scheduler and dispatcher (process manager)
  - Memory manager
  - File manager
  - Device drivers (I/O subsystem)



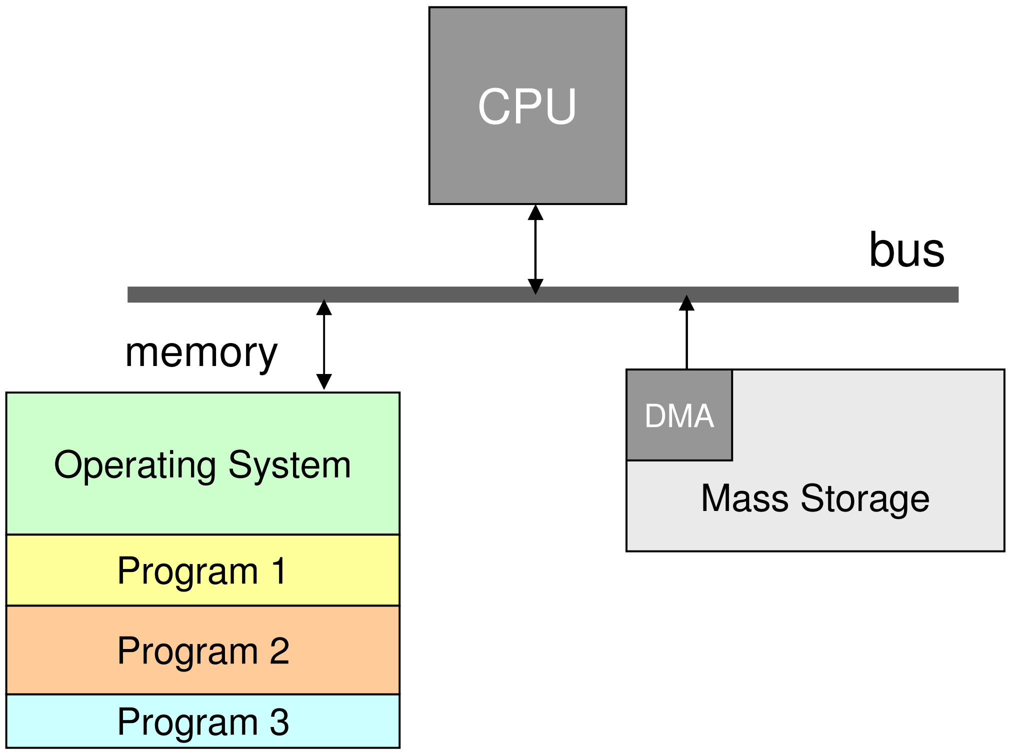
Batch vs. Interactive Processing

- There are two ways a computer processing tasks: batch processing and interactive processing
  - For batch processing, a user asks the ***shell*** to run a special type of program called "script" to execute a sequence of programs
  - For interactive processing, a user uses a keyboard, a mouse, and tablets, etc. to issue commands to the ***shell*** and input data into a computer



Computer Memory Map

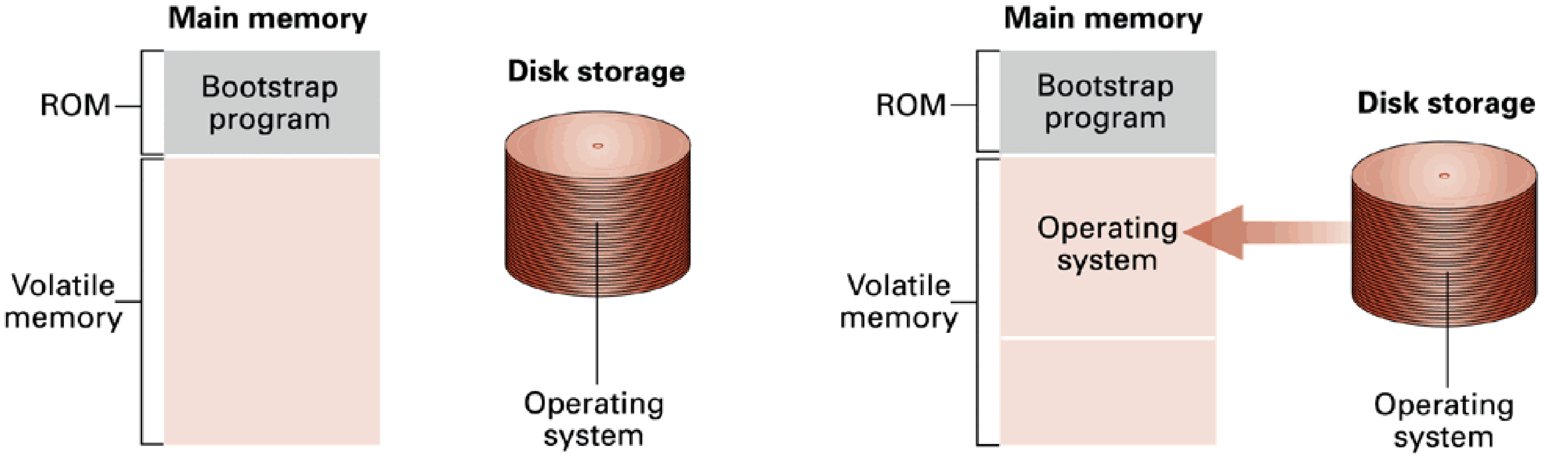
- Today, most computers have multiple programs running in memory simultaneously, under the control of the OS:



Getting it Started (Bootstrapping)

- ❑ The bootstrap program is stored in non-volatile read-only memory (ROM); its functions are:
  - ■ Run by the CPU when power is turned on
  - ■ Transfers operating system from mass storage to main memory
  - ■ Executes jump to operating system

Boot Strapping (Booting)



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

Step 2: Bootstrap program directs the transfer of the operating system into main memory and then transfers control to it.

Processes

- A *program* is a static set of instructions
- A *process* (a.k.a. *task or job*) is the activity of executing a program
  - A process may occupy more memory space than a program
- The *process state* (a.k.a. *process context*) is the current status of the activity. It is a snapshot of relevant parts of the machine at a particular time
- Process state usually includes
  - program counter and some registers
  - associated main memory
  - the execution state

Classification of Tasks (1/2)

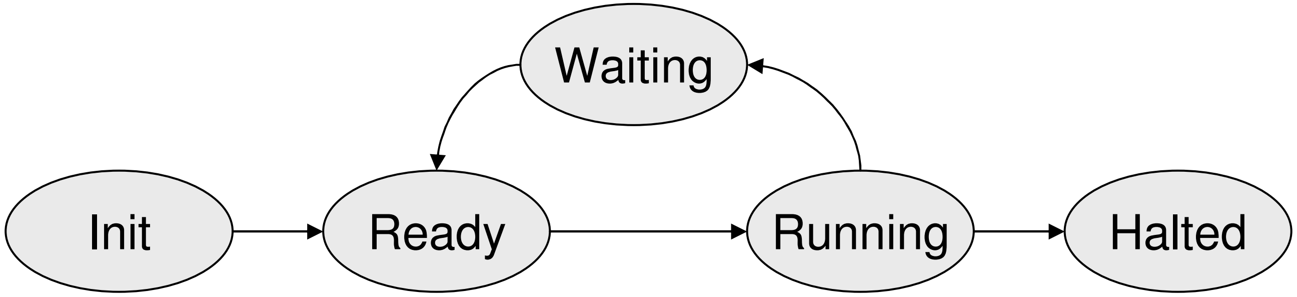
- There are different types of tasks a computer runs:
  - Background tasks:
    - Handling routine (synchronous) tasks
    - Usually called *task level*
  - Foreground tasks:
    - Handling asynchronous events
    - Usually called *interrupt level*
  - Interactive tasks:
    - Handling man-machine interfaces
    - Can be part of the foreground tasks

Classification of Tasks (2/2)

- We can also classify tasks based on time constraint:
  - Batch (offline) tasks:
    - Usually computationally intensive
    - Can be executed without human intervention
  - Real-time tasks:
    - Must finish operations by deadlines
    - Hard real-time: missing deadline causes failure
    - Soft real-time: missing deadline results in degraded performance

Process Execution States

- ❑ When a user select a program to run in a computer, the program becomes a process and it will go through the following execution states
  - Initialization
  - Ready
  - Running
  - Waiting/Sleeping/Blocked/Interrupted
  - Halted



Scheduler

❑ The scheduler maintains a *process table* within the OS; When a user runs a program, it creates a new entry in the process table

❑ After the initialization, the scheduler will add it to the ready pool of processes

❑ If a process cannot continue execution for some reasons (e.g. waiting for data), it will be added to the waiting pool

❑ If a process is finished, the scheduler will remove it from the process table

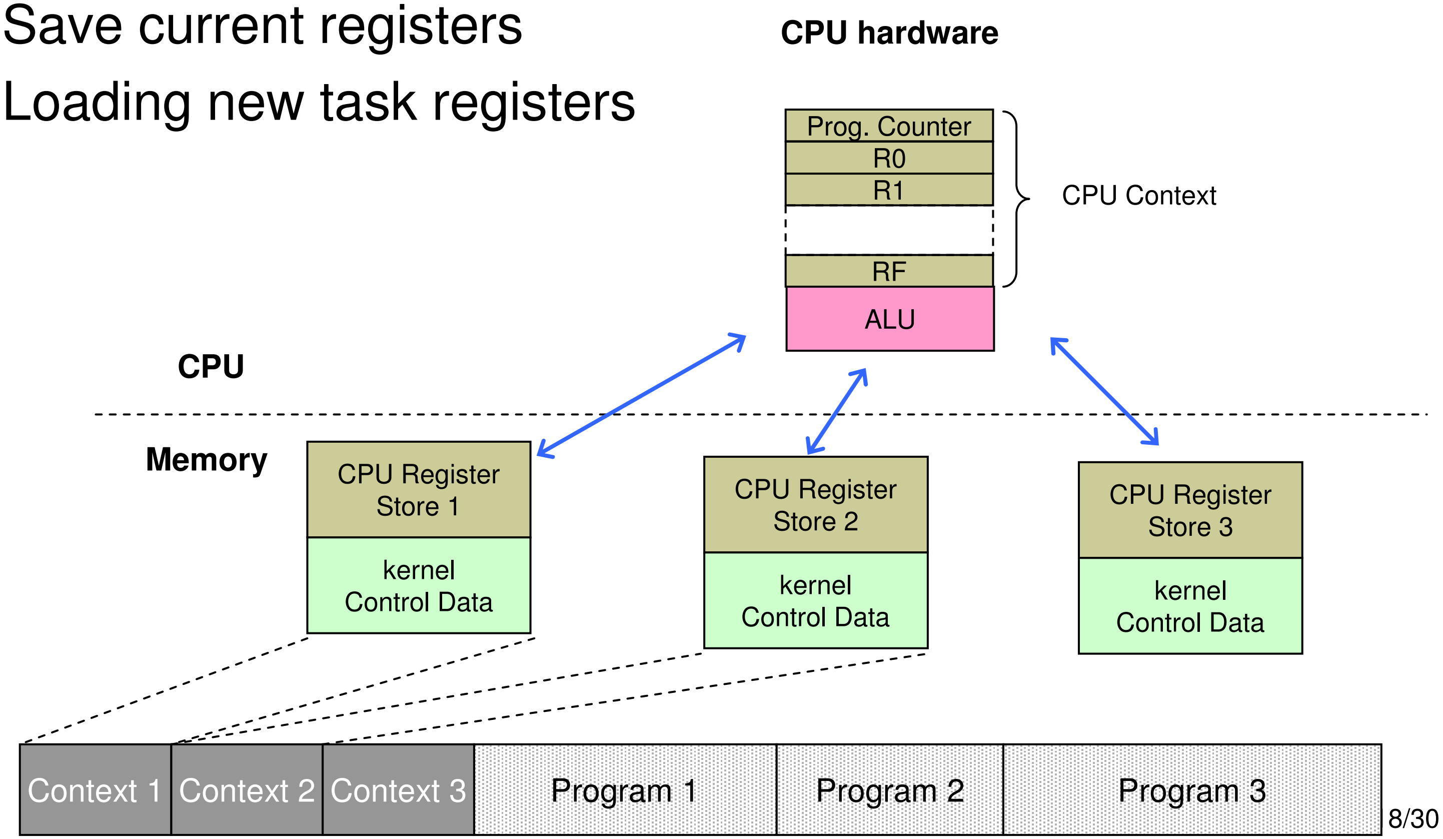
Dispatcher

- Dispatcher is the kernel component that assigns the CPU to execute a ready process
- Dispatcher gain CPU via one of the following ways:
    - Tasks lost CPU to interrupts (preemptive multitasking†)
    - Tasks give up CPU voluntarily (cooperative multitasking)
- Dispatchers assign CPU resources base on:
    - Priority
    - Best effort

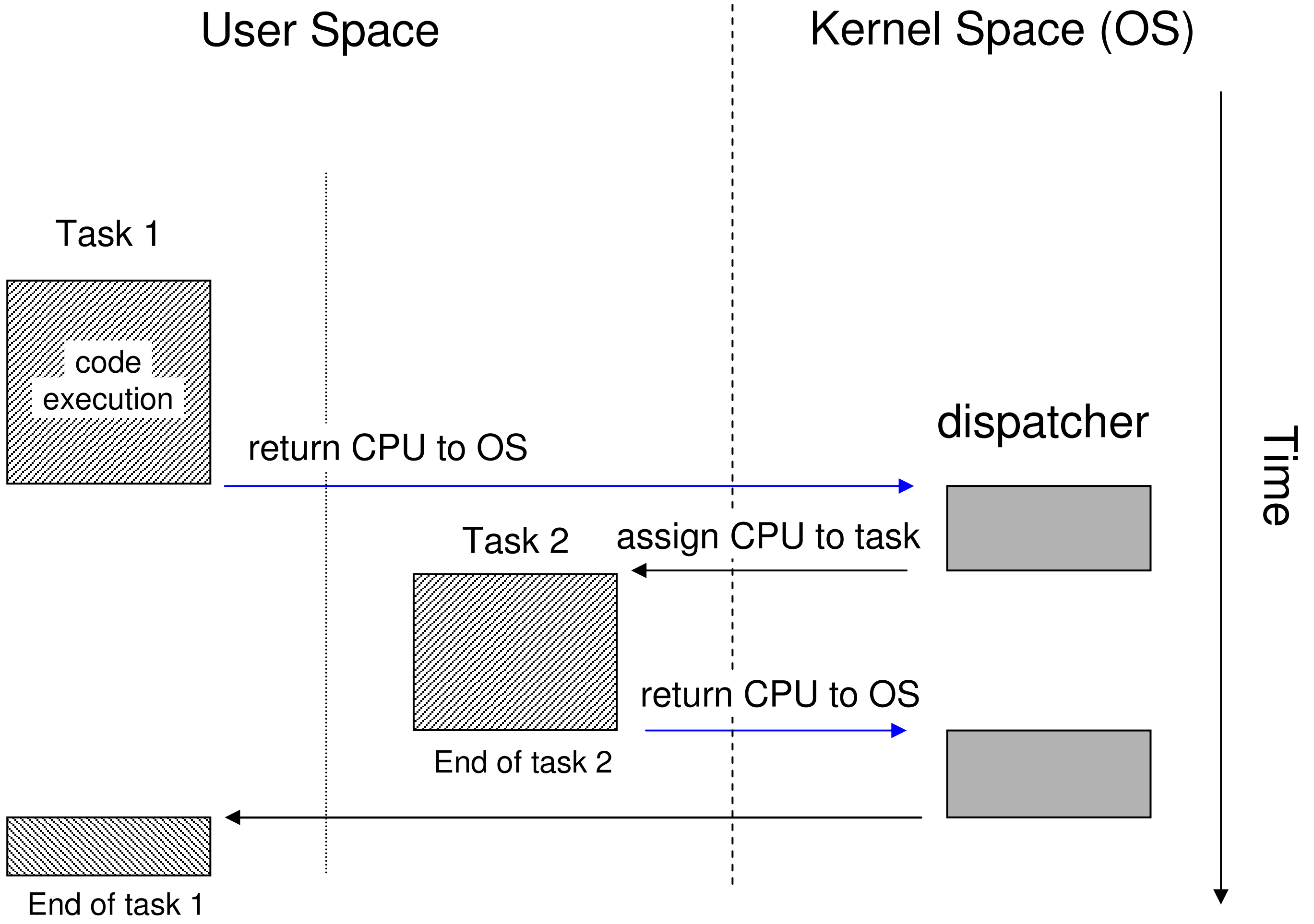
† The terminology "multitasking" is called "multiprogramming" in the textbook.

Context Switch (Process Switch)

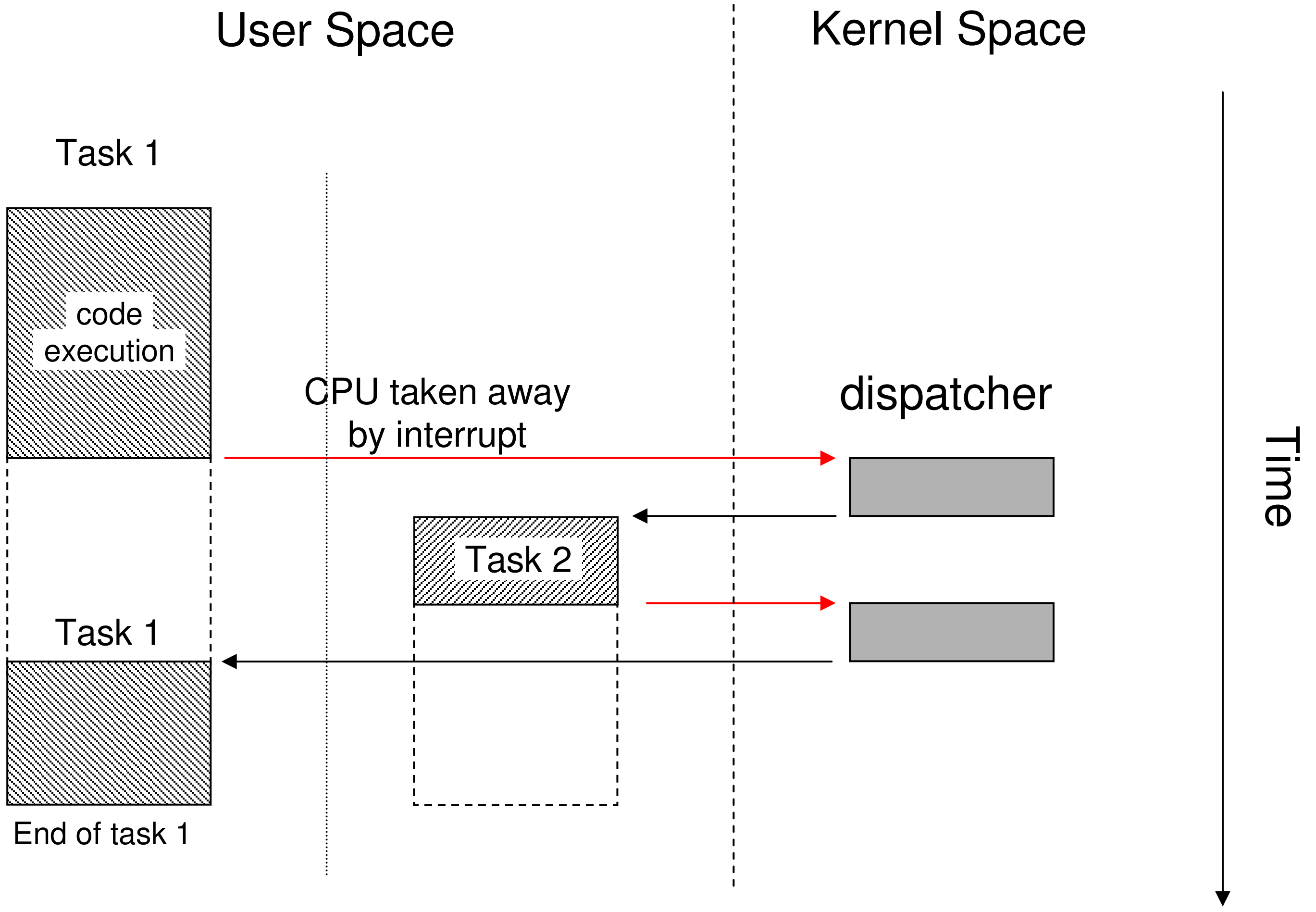
- The operation of assigning CPU to run another task is called context switching. Context switching steps:
  - Save current registers
  - Loading new task registers



Cooperative Multitasking

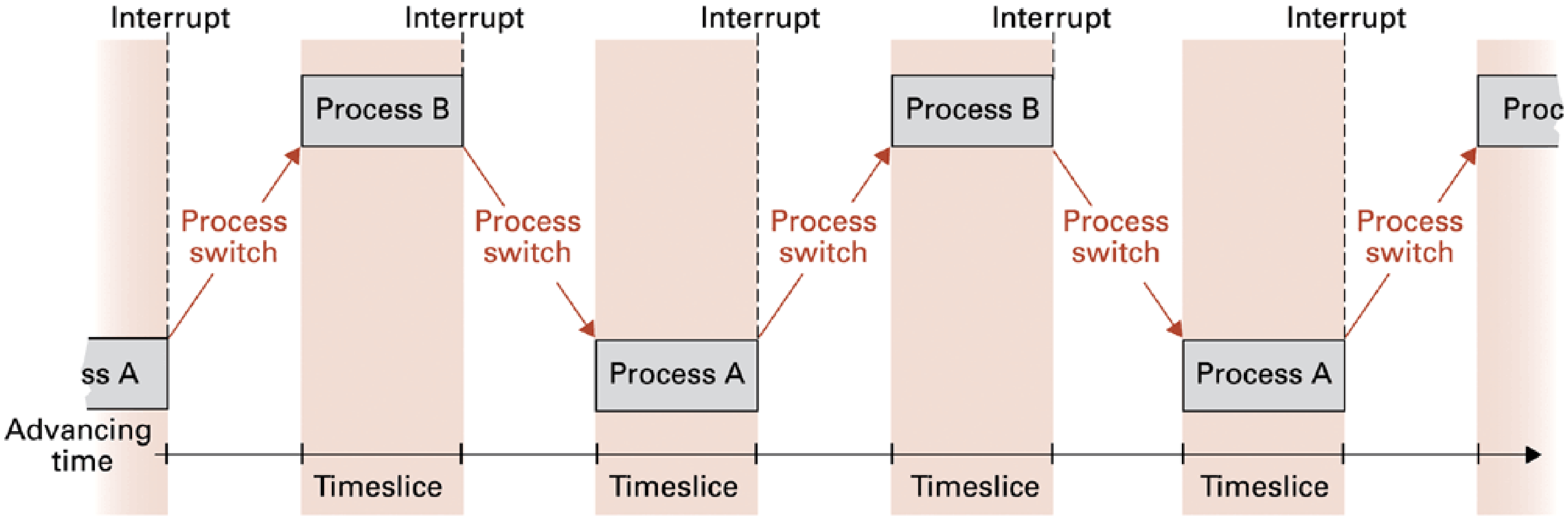


Preemptive Multitasking



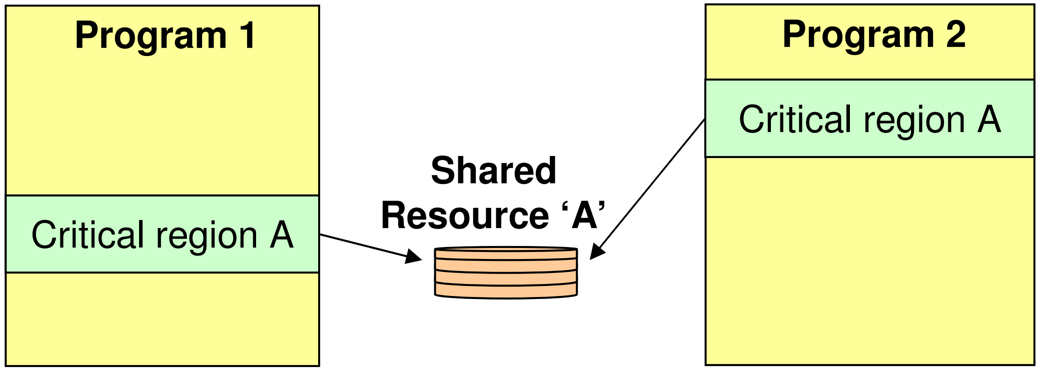
Time-Sharing

- Time-sharing is one type of preemptive multitasking
  - Each process gets the CPU for a short period of time (a few tens of milliseconds); the time period is called timeslice



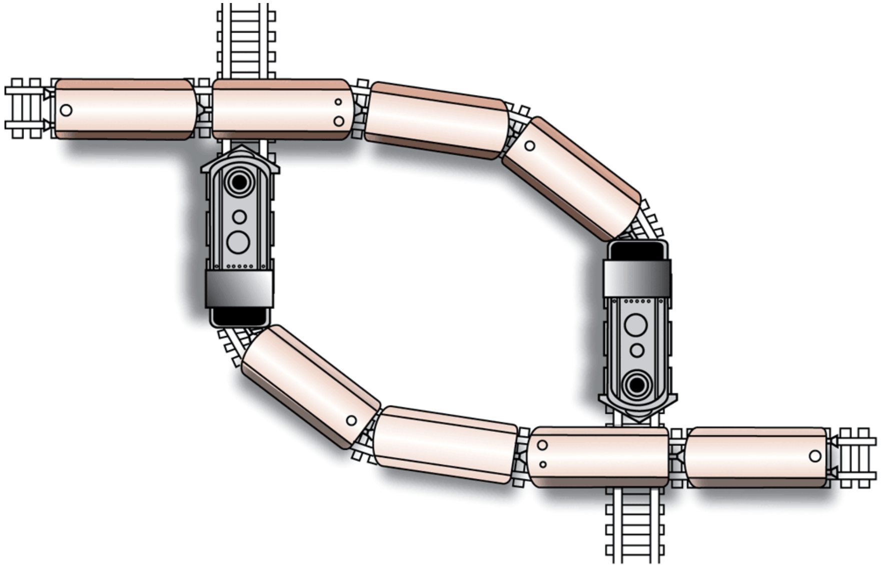
Handling Competition for Resources

- ❑ Semaphore
  - A “variable” (semaphore) is used to tell if a resource is in use
  - Test and set operations on the semaphore must be done atomically → CPUs need a ***test-and-set*** instruction
- ❑ Critical region
  - A critical region is a sequence of instructions that can be executed by only one process at a time (mutual exclusion)
  - Usually protected by a semaphore



Deadlock

- ❑ Two processes block each other from continuing
- ❑ Conditions that lead to deadlock
  1. Competition for non-sharable resources
  2. At least two resources are needed by both processes
  3. An allocated resource can not be forcibly retrieved

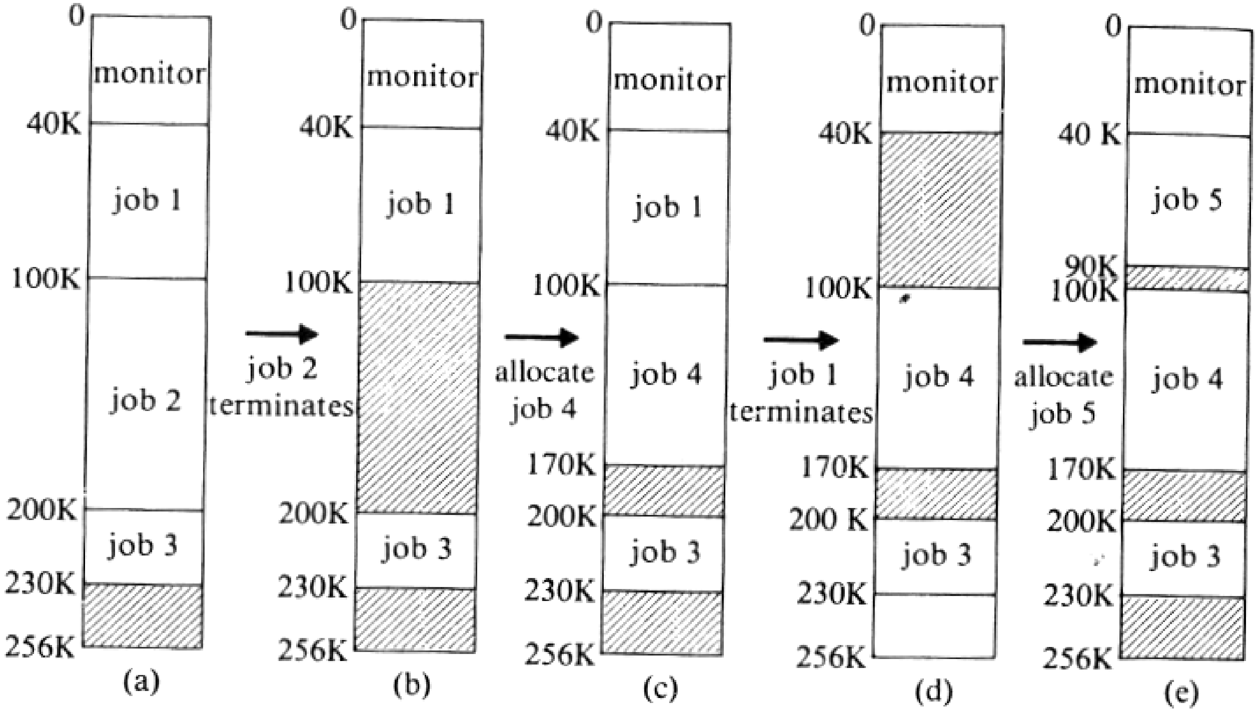


Spooling (Very Old Stuff)

- Spool stands for Simultaneous Peripheral Operation On-Line
- A spooling system stores the output requests from the main CPU to a mass storage device (typically controlled by a smaller processor) so that the main CPU does not have to wait until the request is done
  - For example, for output, the main CPU first sends data to the mass storage
  - Later, the I/O processor reads data from the mass storage and send them to the output devices

Memory Manager

- ❑ Computer main memory must be well organized when multiple processes are running simultaneously

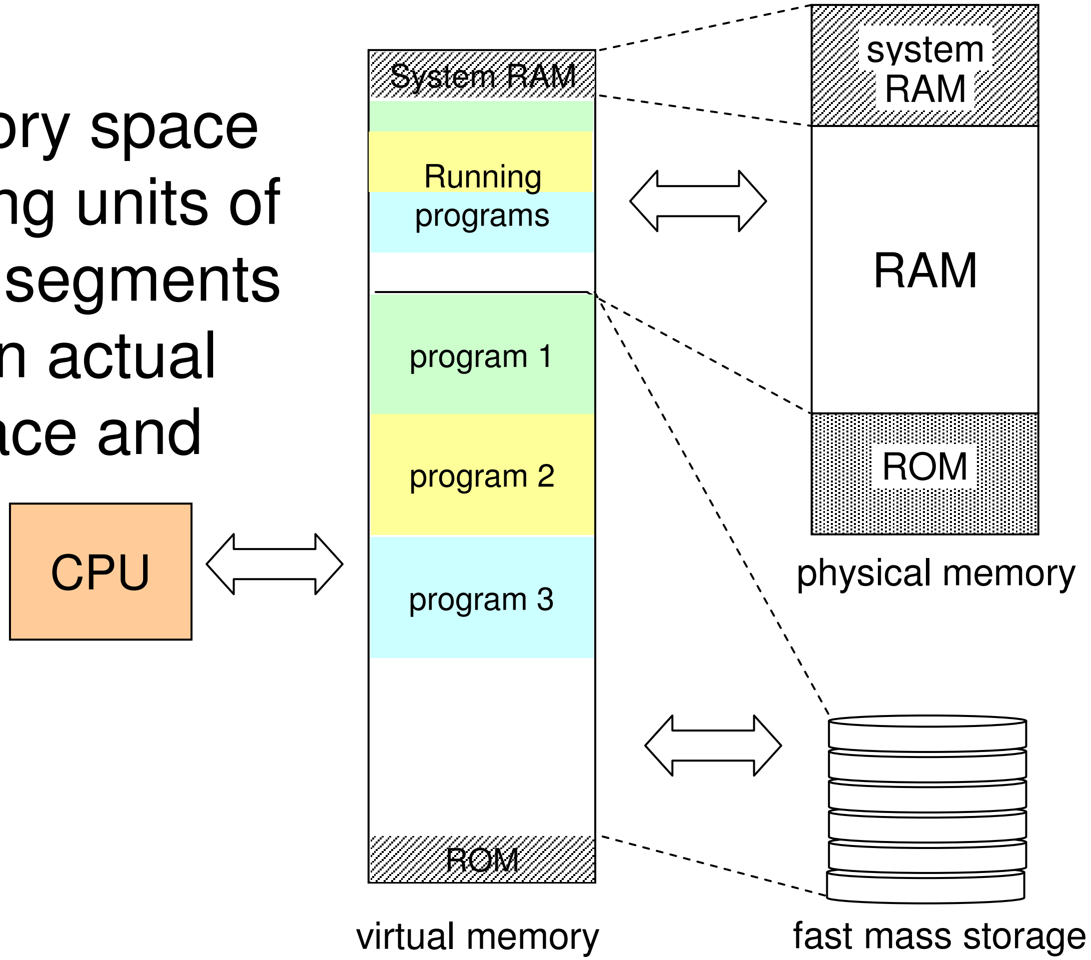


Memory Manager Tasks

- Allocate space in physical memory to a processes for their execution
- Manages virtual memory so that the total memory space of all running processes can be larger than the main memory
  - process images kept in secondary storage
  - images returned to main memory on demand during execution → this techniques is called *paging*

Virtual Memory Concept

- Virtual Memory: "imaginary" memory space created by shuffling units of data/code, called segments or pages, between actual main memory space and mass storage



File Manager

- A file manager controls a machine’s mass storage
  - A file is the logical unit in a computing systems that stores a coherent set of data
  - The way data are structured on a mass storage device is called a file system; a file manager may use different file system for different device
    - For example, MS Windows file manager deal with at least three different file systems: FAT, FAT32, and NTFS
- File manager usually groups files into a bundle called directory or folder
  - The chain of directories leads to the location of a file is called a directory path
  - A *file descriptor* contains the run time information to manipulate the file

Security Issue – External Attacks

- Most computer systems today are protected by using a login system
- Problems with a login system
  - Insecure passwords (password cracker, network sniffer, Trojan Horse login)
  - Sniffing software (a.k.a. spyware)
- Counter measures
  - Auditing software – a program that monitors the operation of a computer and report anything abnormal to the administrator

Security Issue – Internal Attacks

- To keep a user of a computer from manipulating data that does not belongs to him/her, a computer uses the following approaches:
  - CPU must support at least two privilege levels: one for regular programs, the other one for the OS kernel
  - In non-privilege mode, a program's capability is limited by some special purpose registers
  - The OS control process activities via privileged modes and privileged instructions
- Most OS's today have bugs that allow a hacker to get into privileged mode