Operating Systems

Dr. Sapumal Ahangama
Department of Computer Science and Engineering

OUTLINE

- ▶ Introduction
- ▶ Services by Operating Systems
- ▶ Bootstrap
- ▶ Processes and Threads

OPERATING SYSTEM (OS)

▸ **Software component** that provides the basic functionality of the system by offering programs that operate, control, and support the **fundamental resources** of the computer

▸ Fundamental Resources?

SERVICES GIVEN BY OS

▶ Provide user interfaces

▶ File system access

▶ I/O support services

▶ Bootstrapping

▶ Handles interrupt processing

▶ Networking services

▶ Allocates resources

▶ Security

▶ Tools to tune the system

OPERATING SYSTEM (OS)
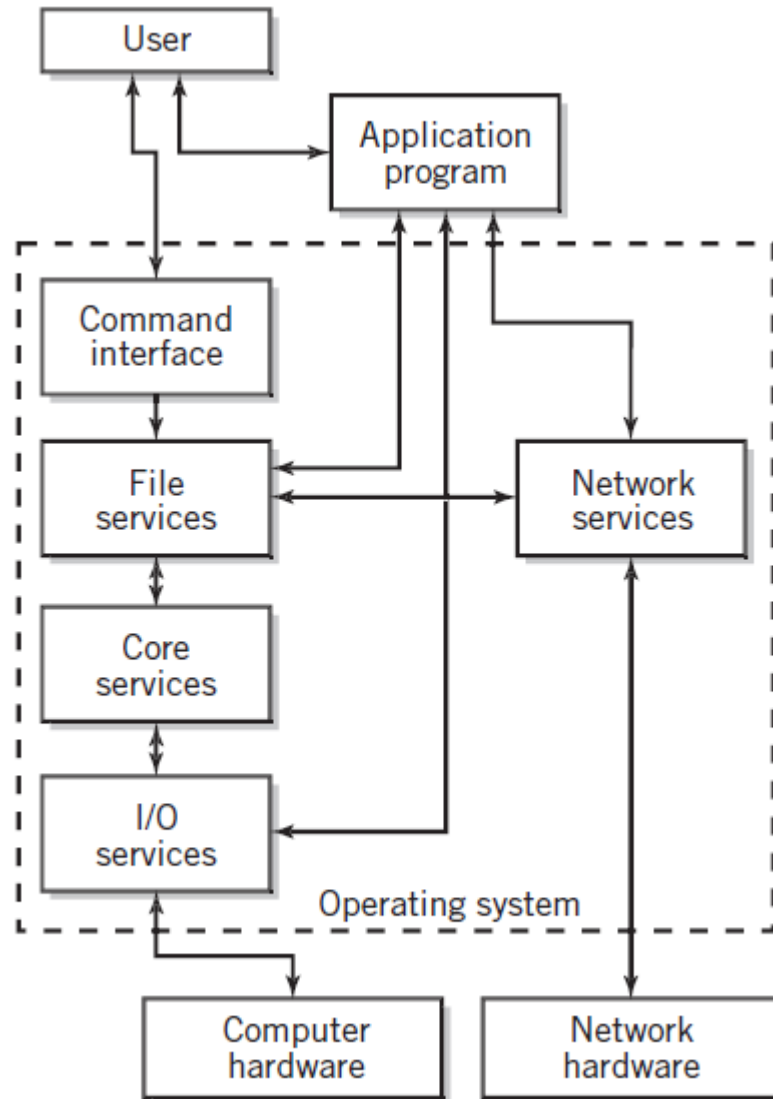
USER INTERFACE (UI)

▸ Provides the user with the capability of executing commands

▸ For some OSs, user interface is considered as separate **shell** that interacts with the **kernel**

▸ Types of UI

  ▸ Graphical User Interface (GUI)

  ▸ Command Line Interface (CLI)


```
mark@linux-desktop: /tmp/tutorial

File  Edit  View  Search  Terminal  Help
mark@linux-desktop:~$ mkdir /tmp/tutorial
mark@linux-desktop:~$ cd /tmp/tutorial
mark@linux-desktop:/tmp/tutorial$ mkdir dir1 dir2 dir3
mark@linux-desktop:/tmp/tutorial$ mkdir
mkdir: missing operand
Try 'mkdir --help' for more information.
mark@linux-desktop:/tmp/tutorial$ cd /etc ~/Desktop
bash: cd: too many arguments
mark@linux-desktop:/tmp/tutorial$ ls
dir1  dir2  dir3
mark@linux-desktop:/tmp/tutorial$ █
```

FILE MANAGEMENT

- Maintains the mapping between a file's logical storage needs and the physical location

- Underlying physical storage issues abstracted from the user

- Maintains directory structure of the files

- Tools to manage the files

- Provides information of the files

- Security mechanisms

I/O SERVICES

- The OS includes I/O device drivers for each device installed on the system

- Provides services,
  - Interrupt handling
  - Services to file management system
  - API for other services
  - Handle I/O requests and transfers

- ROM based drivers for critical I/O devices

PROCESS CONTROL MANAGEMENT

- Every executing program is a process
  - Application programs
  - OS services

- Processes need execution time, memory, I/O services etc.
  - OS manages these common resources among processes
- Management of processes
- Interprocess communication

MEMORY MANAGEMENT

‣ Each program being executed needs to reside in the memory

‣ Memory management will load programs and data to the memory and provide required memory for execution

‣ Main tasks
 ‣ Keeping track of memory
 ‣ Maintains program queues for loading
 ‣ Deallocates program's memory space

SCHEDULING AND DISPATCH

- **High-Level Scheduling**: Which jobs will be admitted to the system with which priority

- **Dispatching:** Actual selection of the process to be executed next
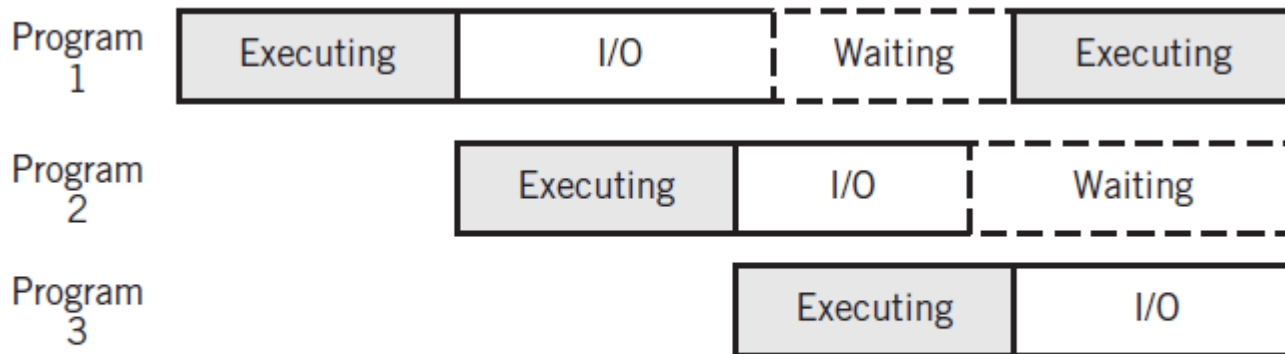
SCHEDULING AND DISPATCH

SCHEDULING AND DISPATCH

- Objectives of dispatching
  - Ensure fairness
  - Maximize throughput
  - Minimize turnaround time
  - Maximize CPU utilization
  - Maximize resource allocation
  - Promote graceful degradation
  - Minimize response time
  - Provide consistent response time
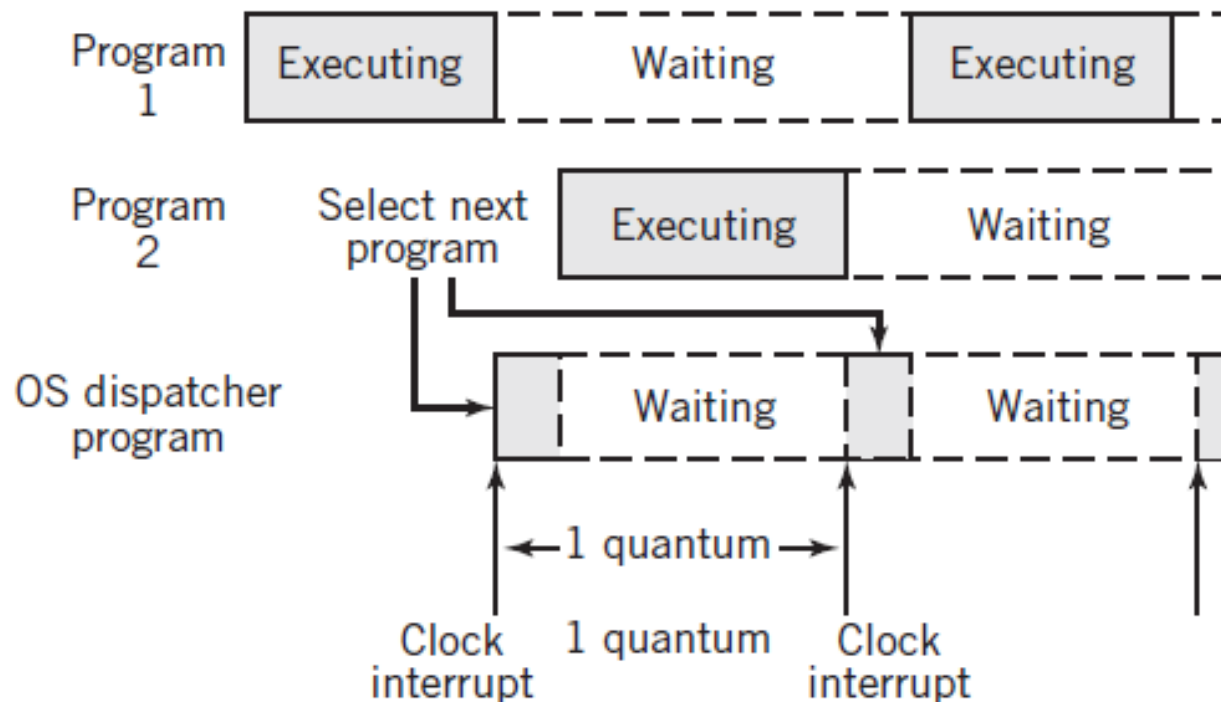 - Prevent starvation

SCHEDULING AND DISPATCH

▸ Non-Preemptive multitasking example

Program 1	Executing	I/O	Waiting	Executing

Program 2		Executing	I/O	Waiting

Program 3			Executing	I/O

SCHEDULING AND DISPATCH

▸ Preemptive multitasking

SECONDARY STORAGE MANAGEMENT

▶ Multiple programs will request I/O access to secondary storage services

▶ Secondary storage management will optimize and schedule these requests

NETWORK SERVICES

▸ Operating system provide the communication software necessary to implement the features and facilities of TCP/IP

  ▸ Locate and connection

  ▸ Pass packet data

  ▸ Error checking and correction

  ▸ Support for distributed processing

SECURITY

- Prevent unauthorized use of the system
  - Login services
  - Firewalls
- Control of access to memory by processes
- Other process controls

SYSTEM ADMINISTRATION

▸ System admin services are provided through the OS

▸ System configuration

▸ User management

▸ Managing file systems, networks

▸ Backup services, system recovery
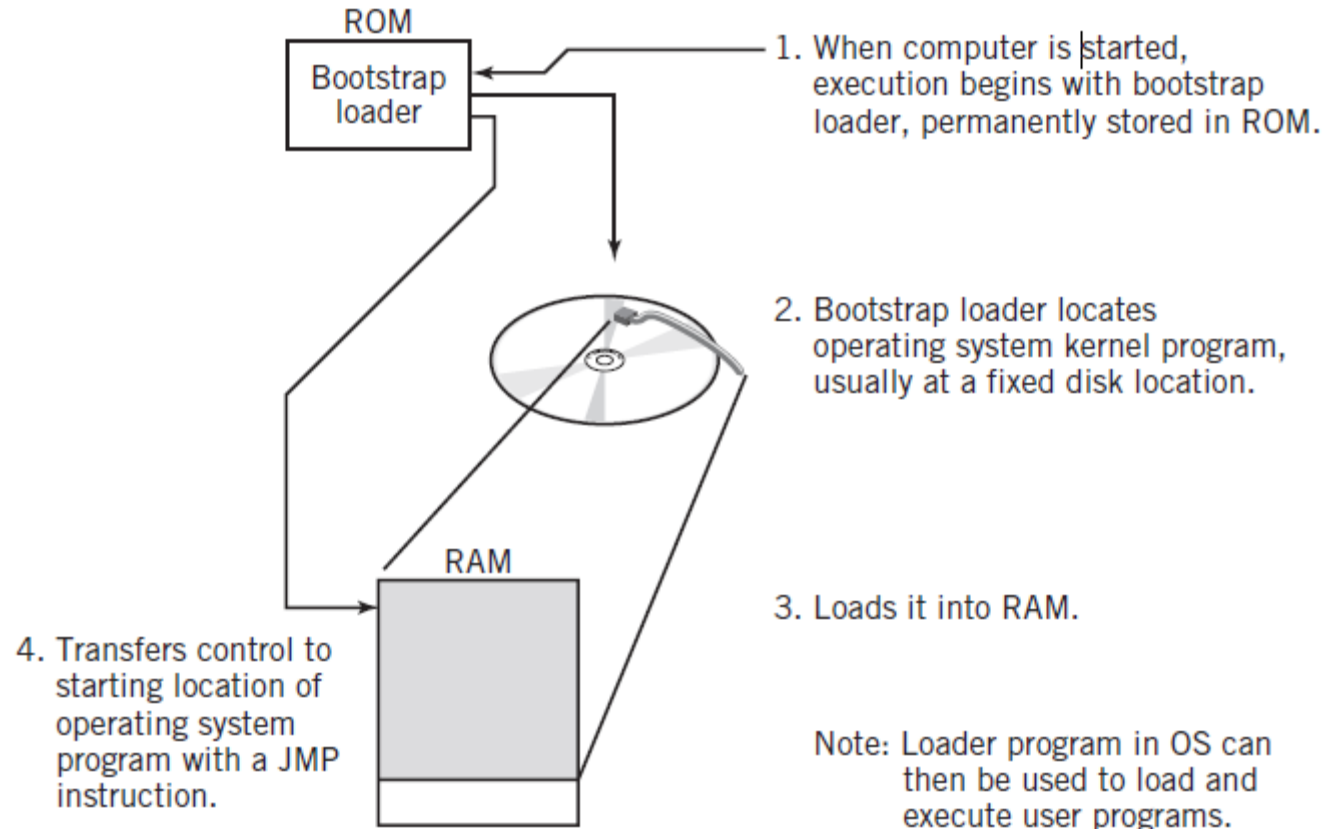
▸ Managing software upgrades

▸ Monitoring

BOOTSTRAP

▸ Memory at start-up?

▸ Bootstrap will conduct the initial program load at startup

   ▸ Located at the ROM

   ▸ Program loader will load from secondary storage to memory

   ▸ Transfers control

BOOTSTRAP

ROM

Bootstrap loader

1. When computer is started, execution begins with bootstrap loader, permanently stored in ROM.

2. Bootstrap loader locates operating system kernel program, usually at a fixed disk location.

RAM

3. Loads it into RAM.

4. Transfers control to starting location of operating system program with a JMP instruction.

Note: Loader program in OS can then be used to load and execute user programs.

PROCESSES AND THREADS

‣ System processes vs. User processes

‣ During process spawning, the OS

  ‣ Assign unique identifier

  ‣ Creates a **process control block**

  ‣ Allocates the memory and other resources

  ‣ Monitoring while it runs

‣ At process exit, resources are returned to the system pool

PROCESSES AND THREADS

▶ Process control block is a data block with all the relevant information about the process

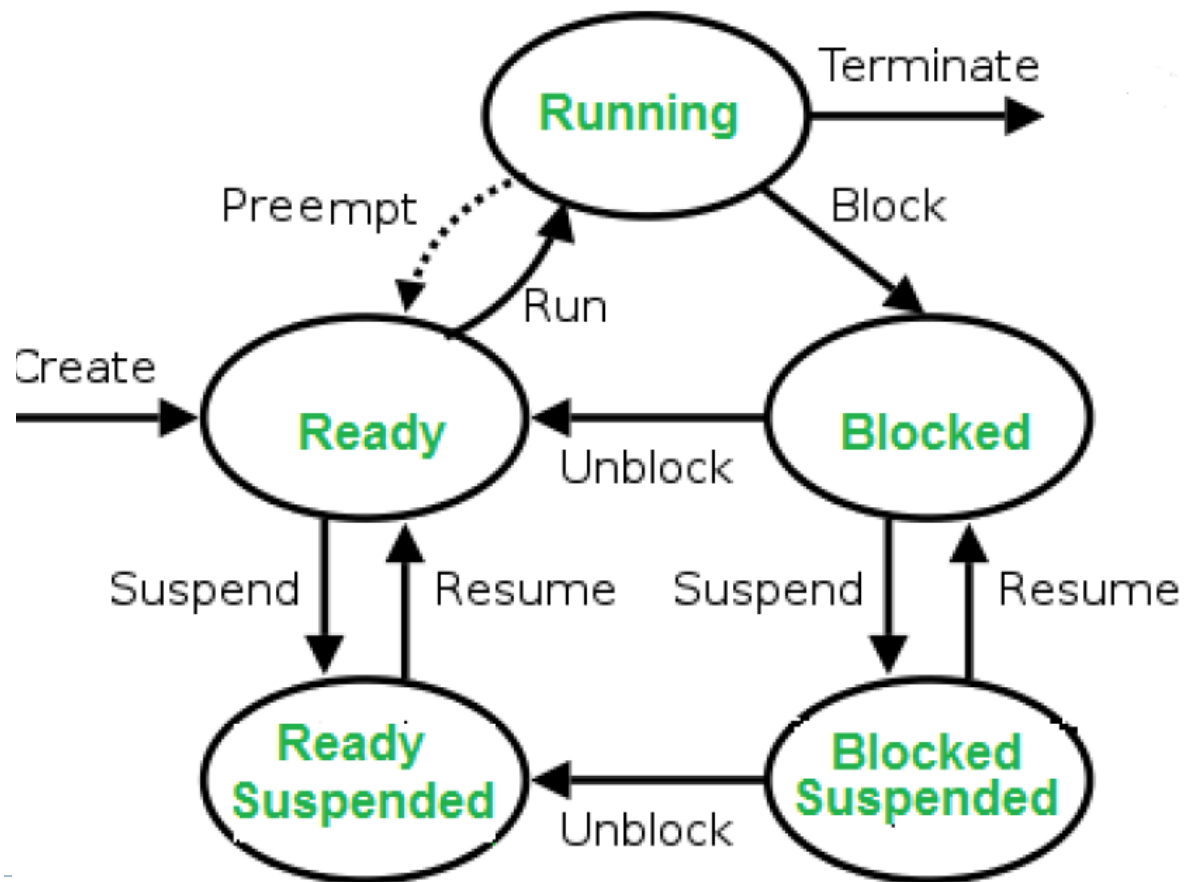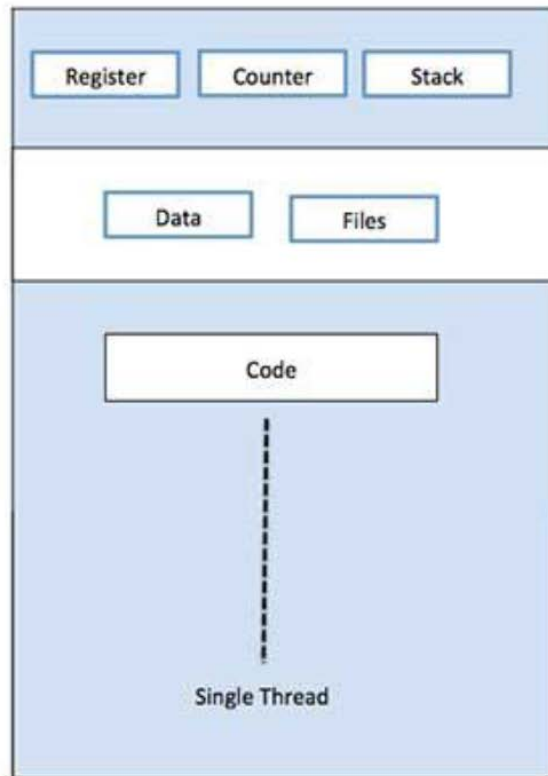| Process ID |
|---|
| Pointer to parent process |
| Pointer area to child processes ... |
| Process state |
| Program counter |
| Register save area ... |
| Memory pointers |
| Priority information |
| Accounting information |
| Pointers to shared memory areas, shared processes and libraries, files, and other I/O resources |

PROCESSES AND THREADS

▸ States of a process
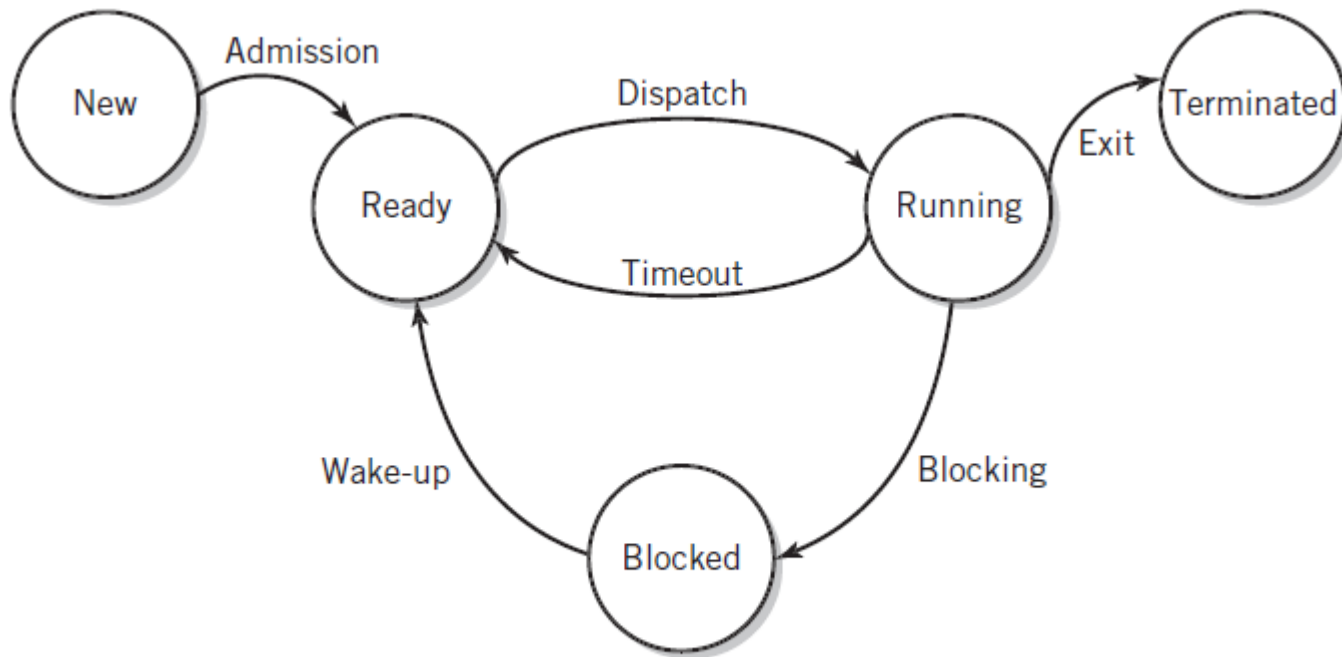
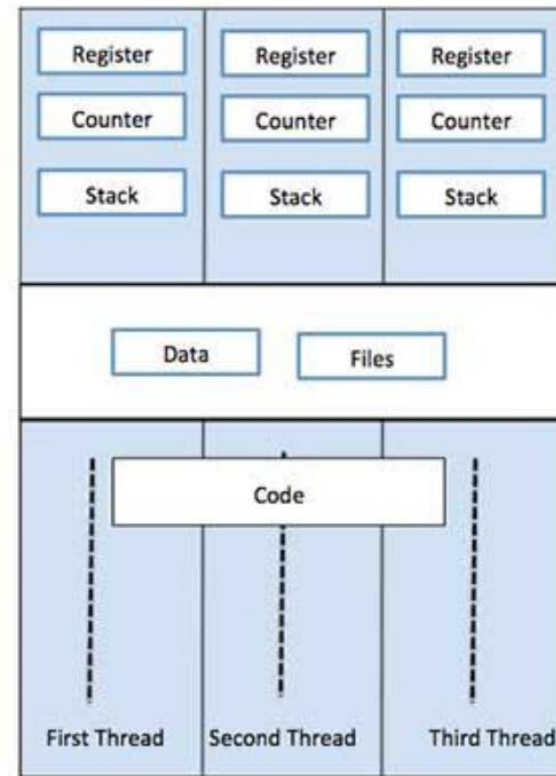# PROCESSES AND THREADS

▸ Extended states of a process

PROCESSES AND THREADS

▸ Threads: a piece of a process that can be executed independently of other parts of the process
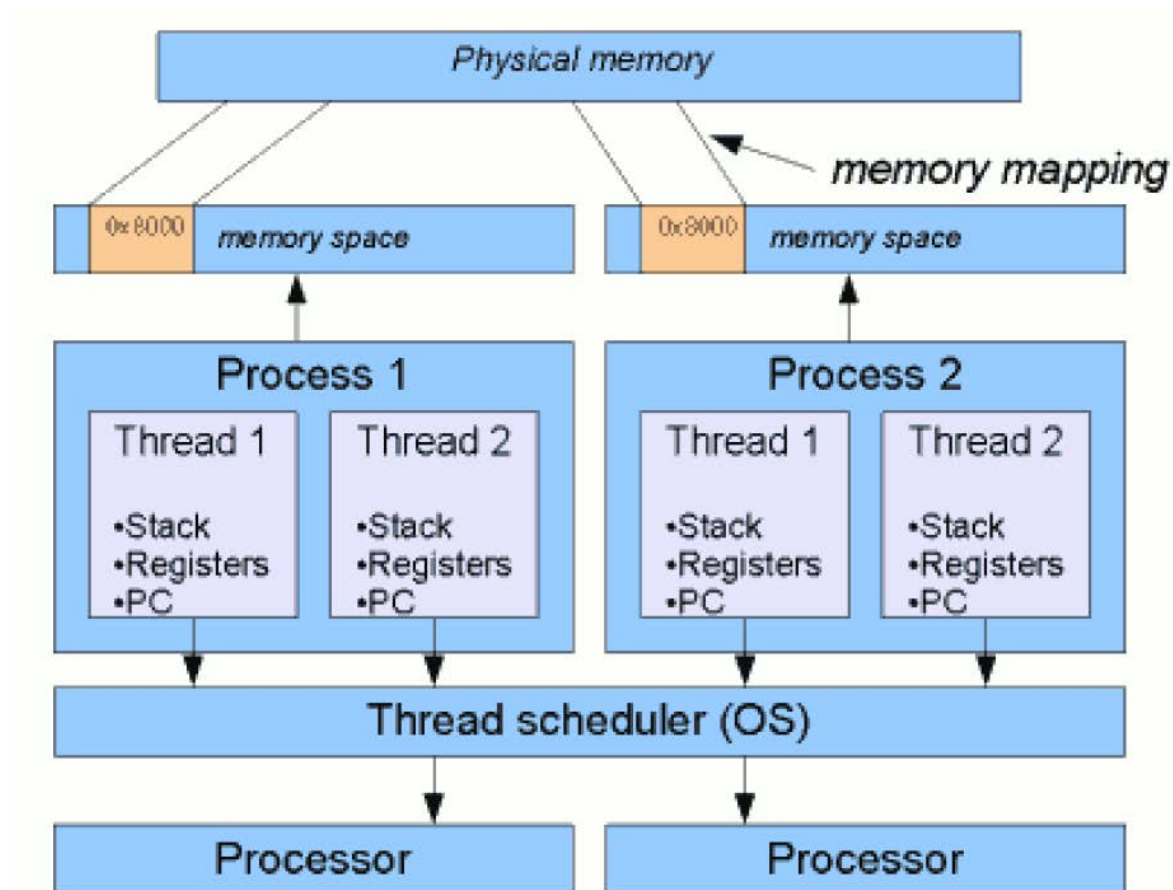

Single Process P with single thread

Single Process P with three threads

PROCESSES AND THREADS

THANK YOU

REFERNCES

- Chapter 15 and 18: The Architecture of Computer Hardware, Systems Software & Networking: An Information Technology Approach -4th Edition, Irv Englander -John Wiley and Sons