OPERATING SYSTEMS

Process Management 1

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

Course Syllabus - Unit 1

UNIT 1: Introduction and Process Management

Operating-System Structure & Operations, Kernel Data Structures,Computing Environments, Operating-System Services, OperatingSystem Design and Implementation. Process concept: Process in memory, Process State, Process Control Block, Process Creation and Termination, CPU Scheduling and Scheduling Algorithms, IPC - Shared Memory & Message Passing, Pipes - Named and Ordinary. Case Study: Linux/Windows Scheduling Policies.

OPERATING SYSTEMS

Course Outline

Class No.	Chapter Title / Reference Literature	Topics to be covered	% of Portions covered	
			Reference chapter	Cumulative
1	1.1-1.2	What Operating Systems Do, Computer-System Organization?	1	21.4
2	1.3,1.4,1.5	Computer-System Architecture, Operating-System Structure & Operations	1	
3	1.10,1.11	Kernel Data Structures, Computing Environments	1	
4	2.1,2.6	Operating-System Services, Operating System Design and Implementation	2	
5	3.1-3.3	Process concept: Process in memory, Process State, Process Control Block, Process Creation and Termination	3	
6	5.1-5.2	CPU Scheduling: Basic Concepts, Scheduling Criteria	5	
7	5.3	Scheduling Algorithms: First-Come, First-Served Scheduling, Shortest-Job-First Scheduling	5	
8	5.3	Scheduling Algorithms: Shortest-Job-First Scheduling (Pre-emptive), Priority Scheduling	5	
9	5.3	Round-Robin Scheduling, Multi-level Queue, Multi-Level Feedback Queue Scheduling	5	
10	5.5,5.6	Multiple-Processor Scheduling, Real-Time CPU Scheduling	5	
11	5.7	Case Study: Linux/Windows Scheduling Policies	5	
12	3.4,3.6.3	IPC – Shared Memory & Message Passing, Pipes – Named and Ordinary	3,6	

Topics Outline

- Process Concept
  - The Process
  - Process State
  - Process Control Block
- Process Scheduling
  - Scheduling Queues
  - Schedulers
  - Context Switch
- Operations on Processes
  - Process Creation
  - Process Termination
- Typical Q and As

Process Concept

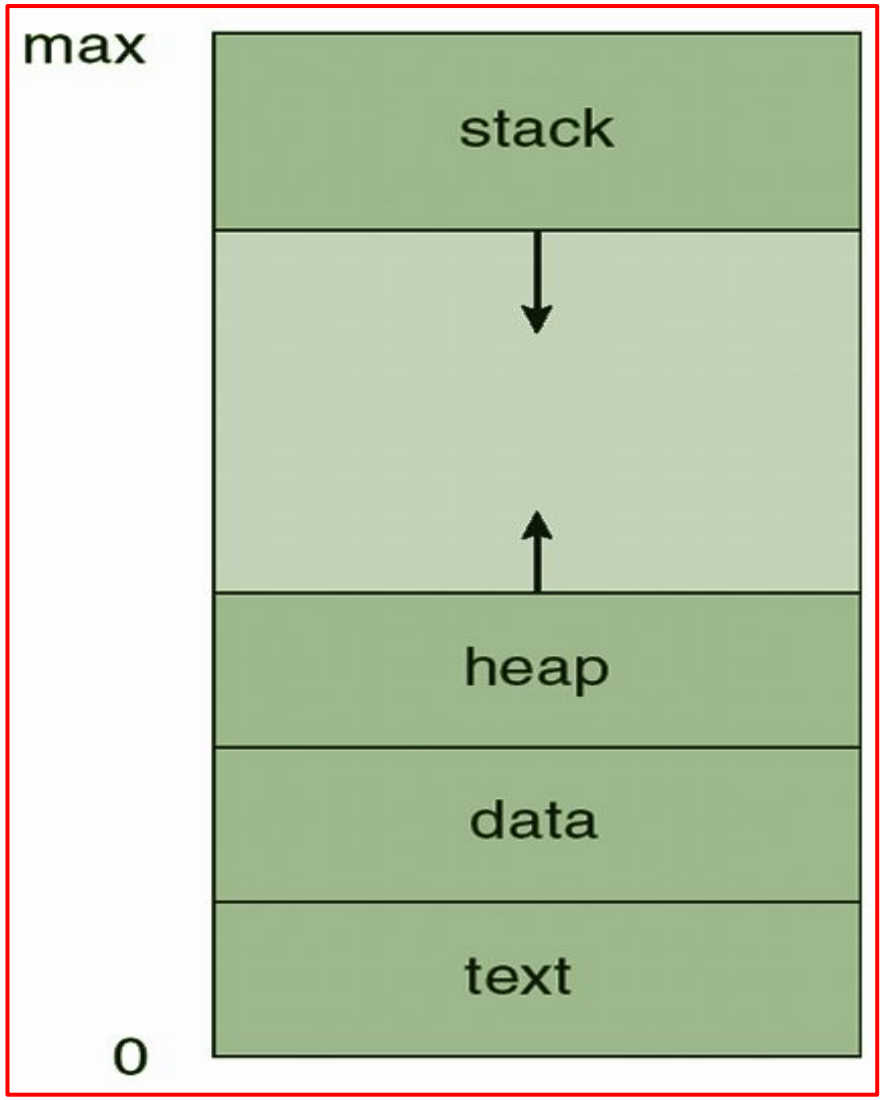
- An operating system executes a variety of programs:
  - Batch system – **jobs**
  - Time-shared systems – **user programs or tasks**
- Some Literature uses the terms **job** and **process** almost interchangeably
- **Process** – a program in execution; process execution must progress in sequential fashion

Process Concept: The Process

- Program is ***passive*** entity stored on disk (**executable file**), process is ***active*** currently in the **Physical memory** and/or **Virtual memory**
- Program becomes process when executable file **loaded** into **memory**
- Execution of program started via GUI mouse clicks, command line entry of its name, etc
- **One program** can be **several processes**
  - Consider multiple users executing the same program

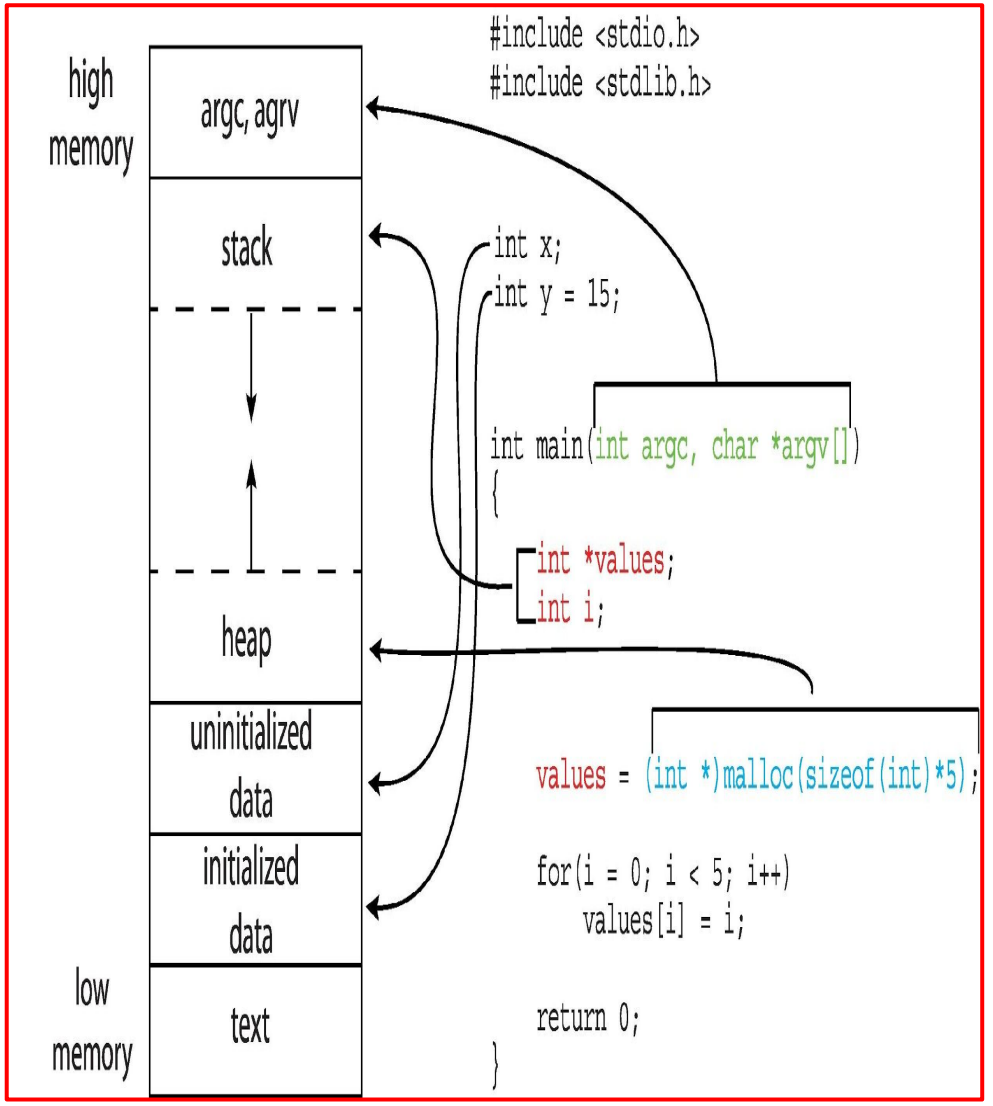
Process Concept: The Process

- Multiple parts
  - The program code, also called **text section**
  - Current activity including **program counter**, processor registers
  - **Stack** containing temporary data
    - Function parameters, return addresses, local variables
  - **Data section** containing global variables
  - **Heap** containing memory dynamically allocated during run time



Process Concept: The Process

- Multiple parts
  - The program code, also called **text section**
  - Current activity including **program counter**, processor registers
  - **Stack** containing temporary data
    - Function parameters, return addresses, local variables
  - **Data section** containing global variables
  - **Heap** containing memory dynamically allocated during run time

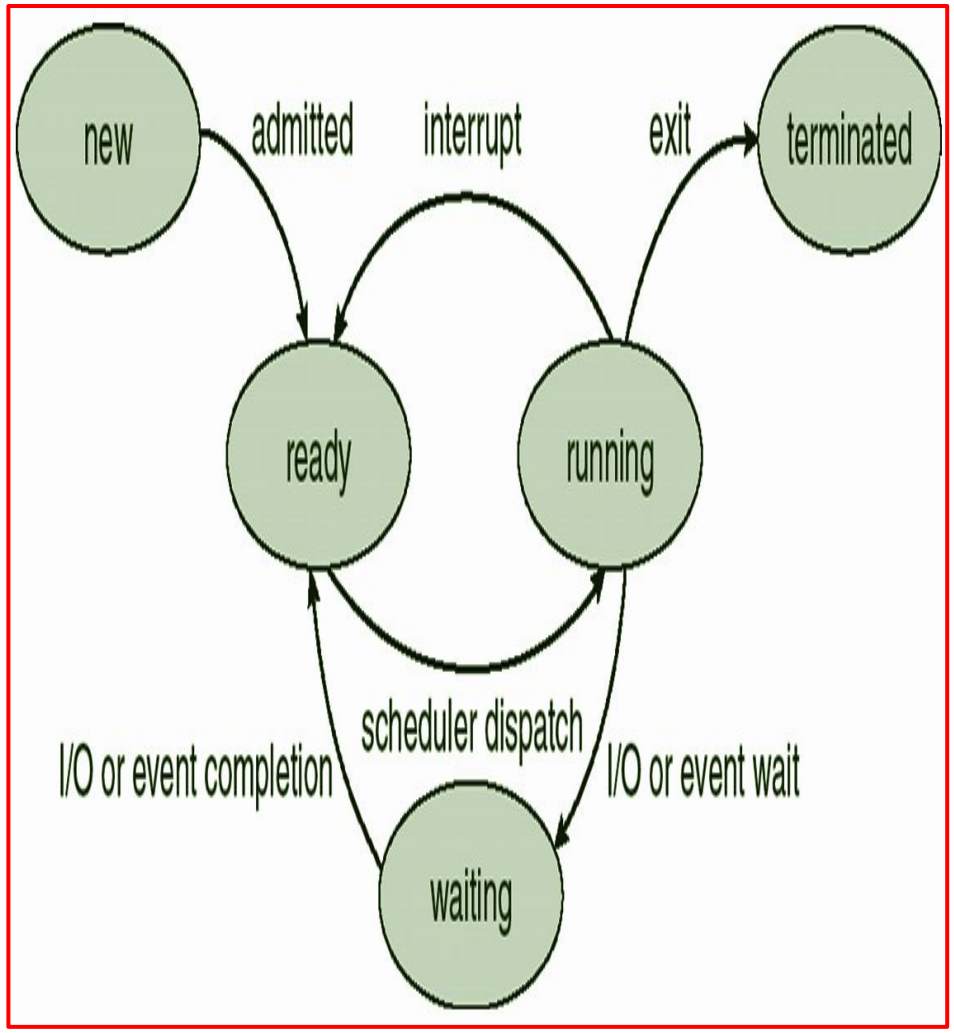


Process Concept: The Process State

- As a process executes, it changes state
  - **New**: The process is being created
  - **Running**: Instructions are being executed
  - **Waiting**: The process is waiting for some event to occur
  - **Ready**: The process is waiting to be assigned to a processor
  - **Terminated**: The process has finished execution

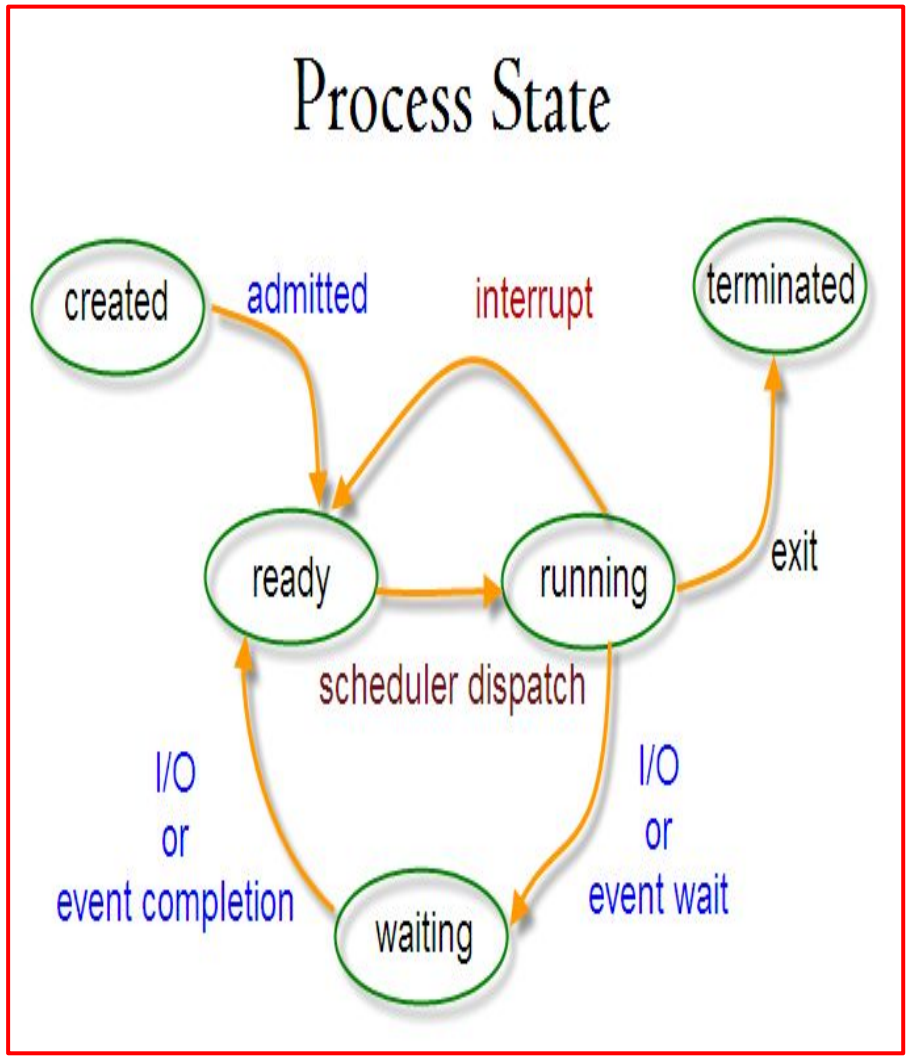
Process Concept: The Process State

- These names are arbitrary, and they vary across operating systems.
- The states that they represent are found on all systems
- Certain operating systems also more finely delineate process states.
- It is important to realize that only one process can be running on any processor at any instant.
- Many processes may be ready and waiting



Process Concept: The Process State

- These names are arbitrary, and they vary across operating systems.
- The states that they represent are found on all systems
- Certain operating systems also more finely delineate process states.
- It is important to realize that only one process can be running on any processor at any instant.
- Many processes may be ready and waiting



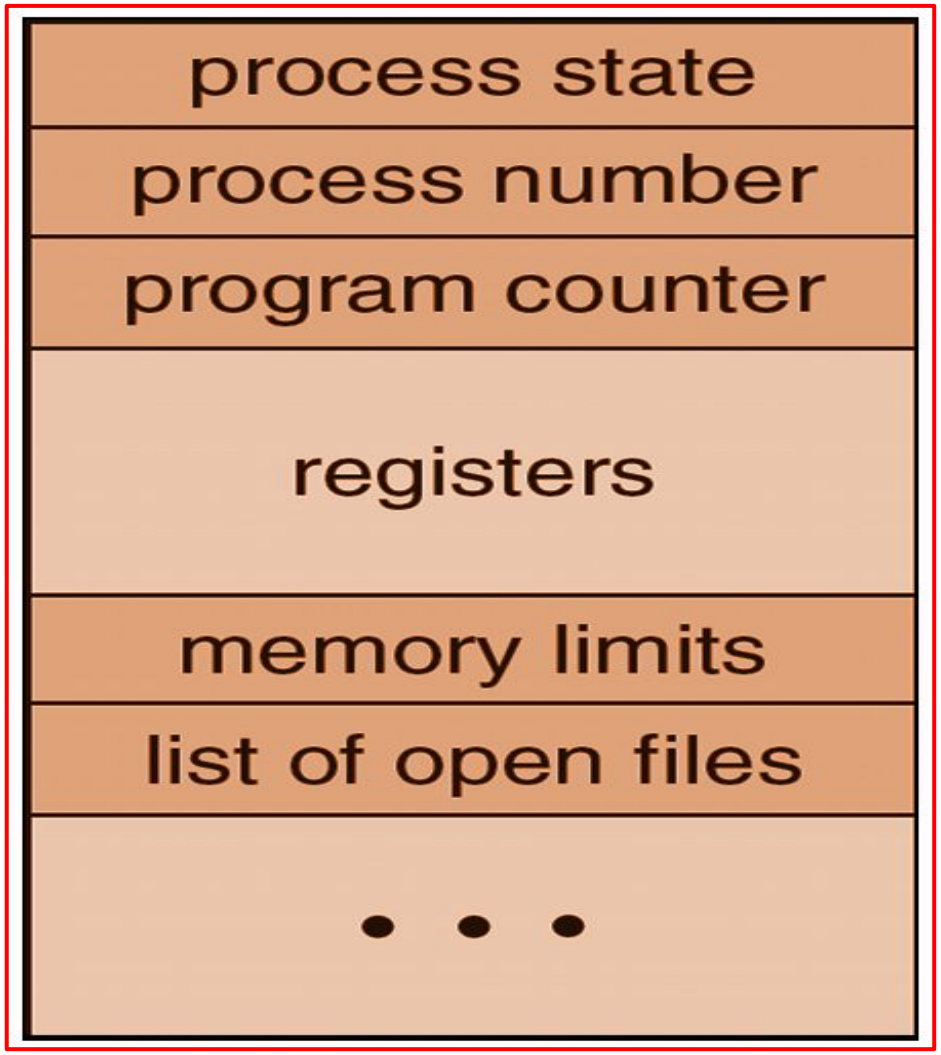
Process Concept: The Process Control Block (PCB)

Information associated with each process (also called **task control block**)

- **Process state** => running, waiting, etc
- **Program counter** => location of instruction to next execute
- **CPU registers** => contents of all process-centric registers
- **CPU scheduling information** => priorities, scheduling queue pointers
- **Memory-management information** => memory allocated to the process
- **Accounting information** => CPU used, clock time elapsed since start, time limits
- **I/O status information** => I/O devices allocated to process, list of open files

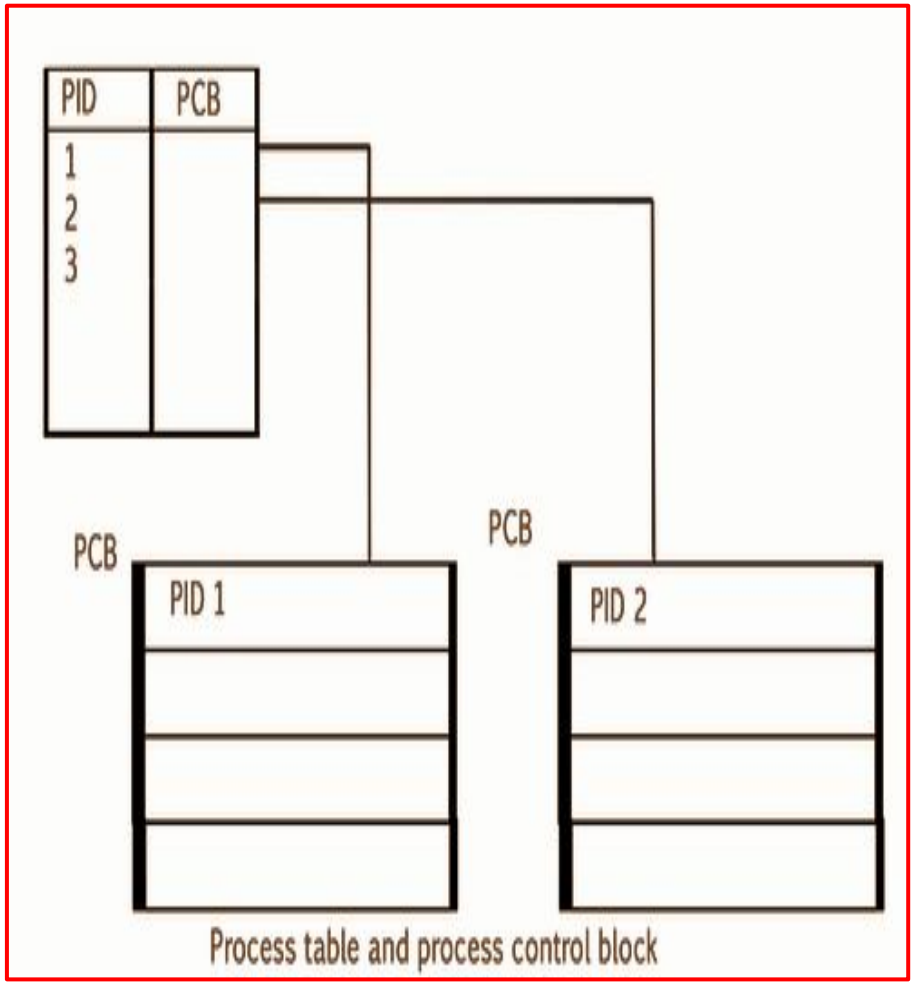
Process Concept: The PCB Representation

- **Pointer –** It is a stack pointer which is required to be saved when the process is switched from one state to another to retain the current position of the process.
- **Process state –** It stores the respective state of the process.
- **Process number –** Every process is assigned with a unique id known as process ID or PID which stores the process identifier.
- **Program counter –** It stores the counter which contains the address of the next instruction that is to be executed for the process.
- **Register –** These are the CPU registers which includes: accumulator, base, registers and general purpose registers.
- **Memory limits –** This field contains the information about memory management system used by operating system. This may include the page tables, segment tables etc.
- **Open files list –** This information includes the list of files opened for a process.



Process Concept: The PCB Representation

- **Pointer** – It is a stack pointer which is required to be saved when the process is switched from one state to another to retain the current position of the process.
- **Process state** – It stores the respective state of the process.
- **Process number** – Every process is assigned with a unique id known as process ID or PID which stores the process identifier.
- **Program counter** – It stores the counter which contains the address of the next instruction that is to be executed for the process.
- **Register** – These are the CPU registers which includes: accumulator, base, registers and general purpose registers.
- **Memory limits** – This field contains the information about memory management system used by operating system. This may include the page tables, segment tables etc.
- **Open files list** – This information includes the list of files opened for a process.



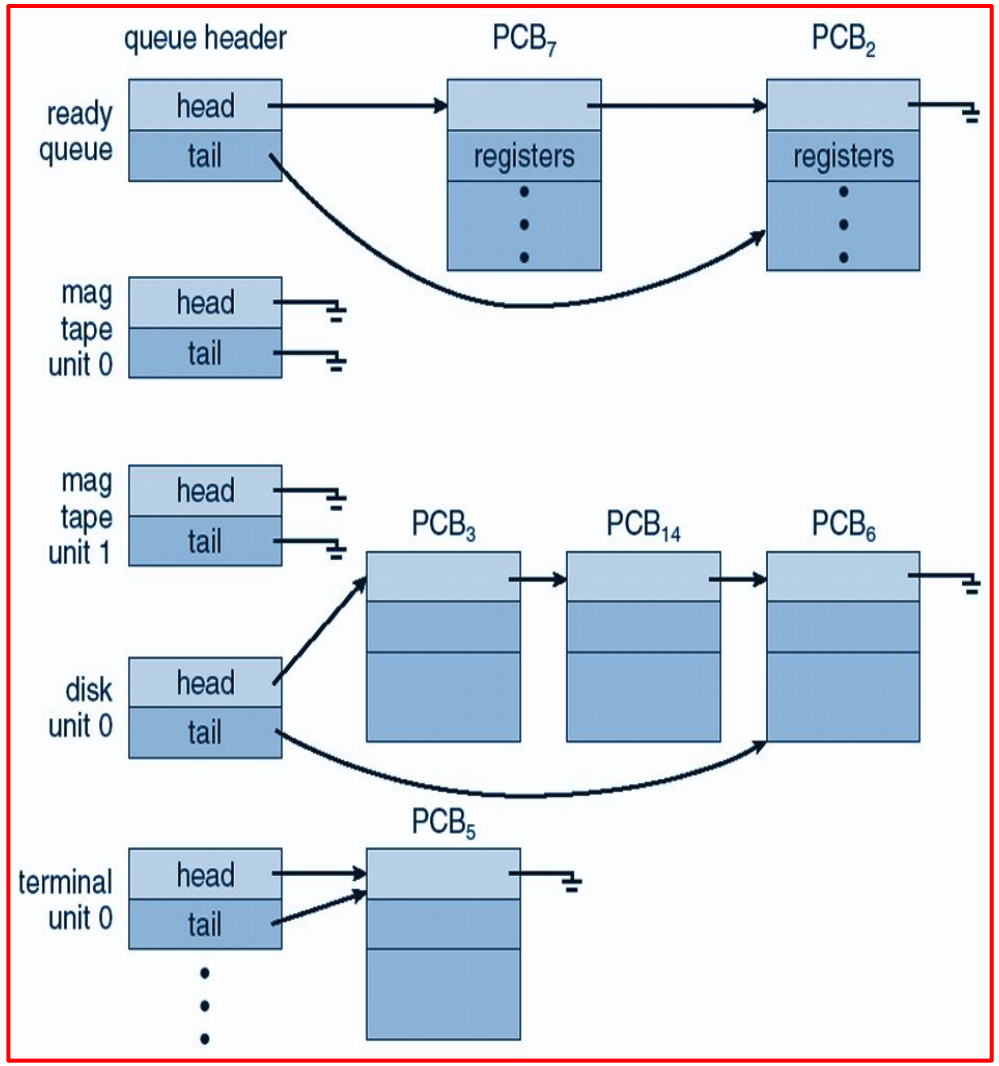
Process table and process control block

Process Scheduling

- The **objective** of **multiprogramming** is to have some process running at all times, to **maximize CPU utilization**.
- The **objective** of **time sharing** is to switch the CPU among processes so frequently that users can interact with each program while it is running.
- To meet these objectives, the process scheduler selects an available process, possibly from a set of several available processes for program execution on the CPU .
- For a single-processor system, there will **never** be **more than one running process**.
- If there are more processes, the rest will have to **wait** until the CPU is free and can be **rescheduled**.

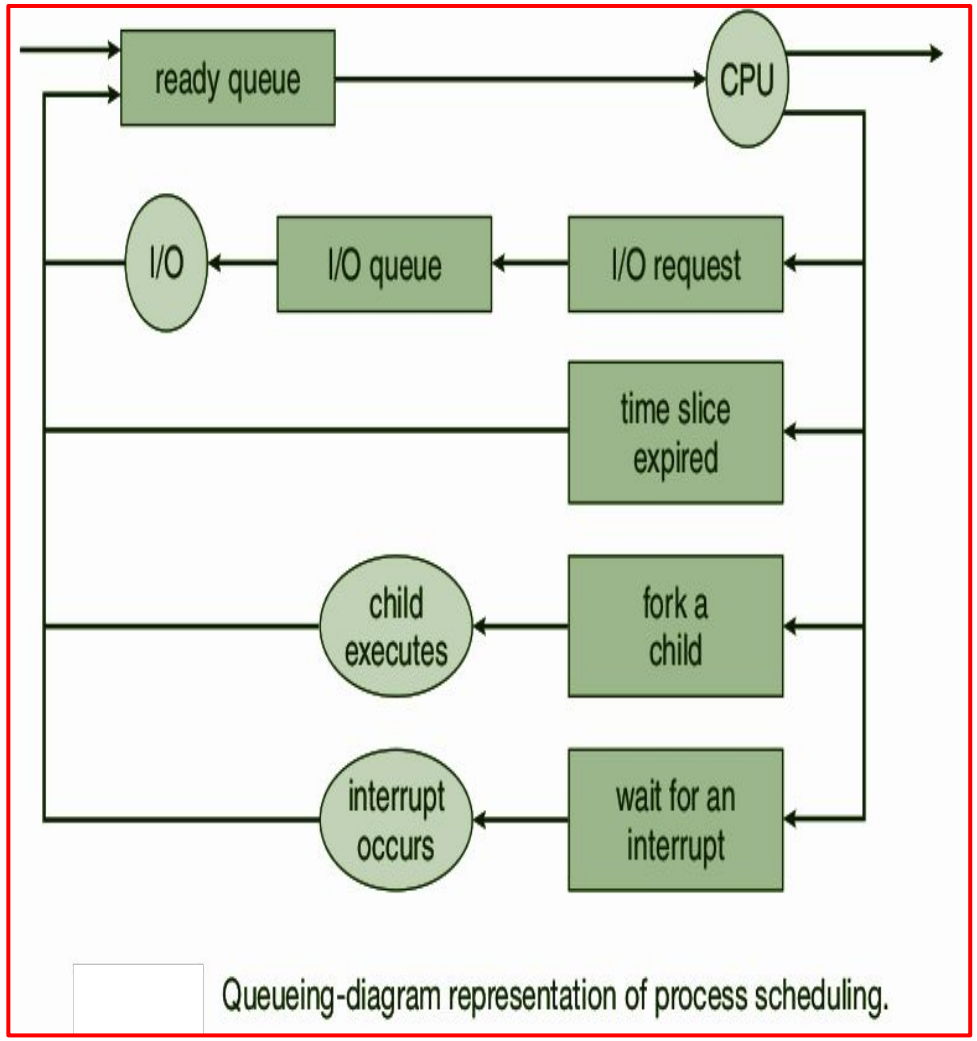
Process Scheduling: Scheduling Queues

- Maximize CPU use, quickly switch processes onto CPU for time sharing
- Process scheduler selects among available processes for next execution on CPU
- Maintains scheduling queues of processes
  - **Job queue** – set of all processes in the system
  - **Ready queue** – set of all processes residing in main memory, ready and waiting to execute
  - **Device queues** – set of processes waiting for an I/O device
  - Processes **migrate** among the various queues



Process Scheduling: Scheduling Queues

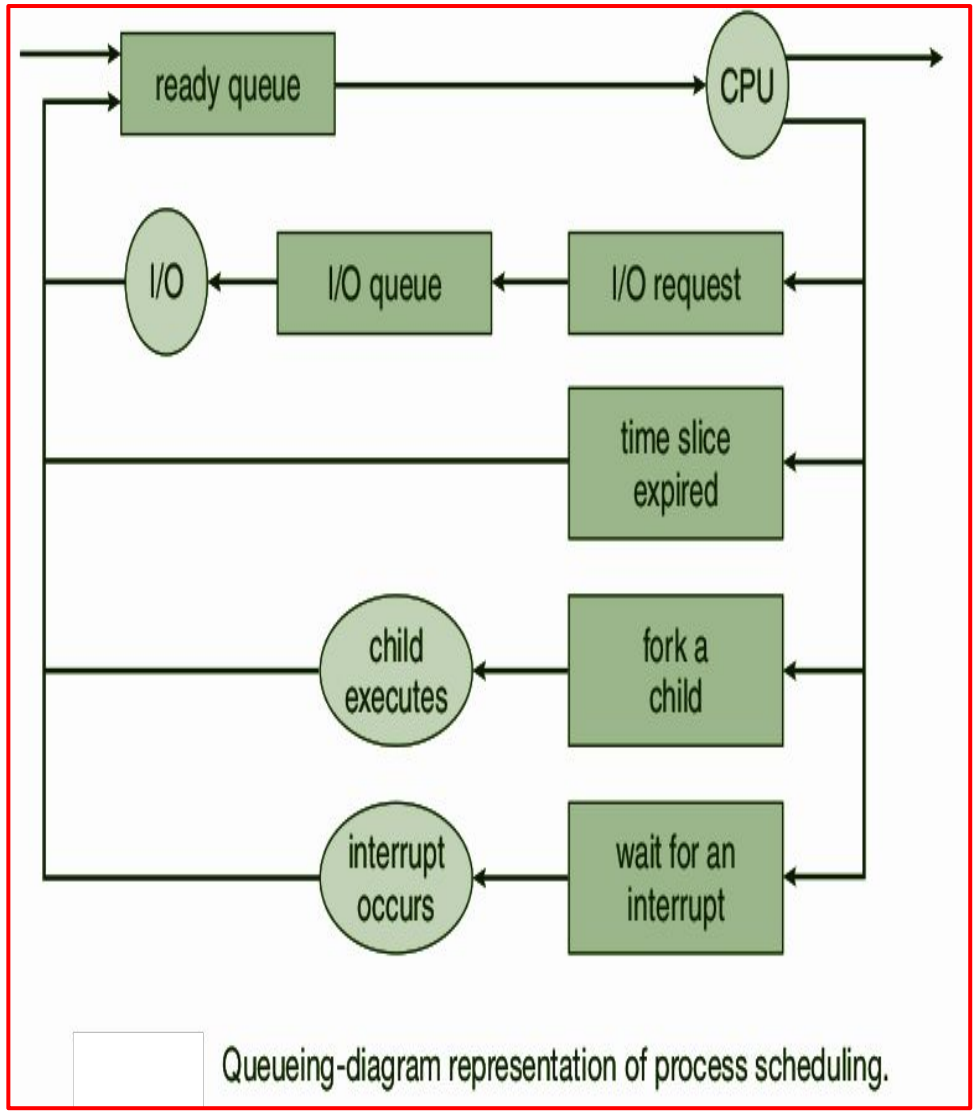
- A common representation of process scheduling is a queueing diagram, as shown in the figure on the side
- Two types of queues are present: the **ready queue** and a set of **device queues**.
- The **circles** represent the **resources** that serve the queues, and the arrows indicate the flow of processes in the system.
- A new process is initially put in the ready queue.
- It waits there until it is selected for execution, or dispatched.



Queueing-diagram representation of process scheduling.

Process Scheduling: Scheduling Queues

- Once the process is allocated the CPU and is executing, one of several events could occur
  - **Case 1:** The process could issue an I/O request and then be placed in an I/O queue
  - **Case 2:** The process could create a new child process and wait for the child's termination.
  - **Case 3:** The process could be removed forcibly from the CPU , as a result of an interrupt, and be put back in the ready queue.

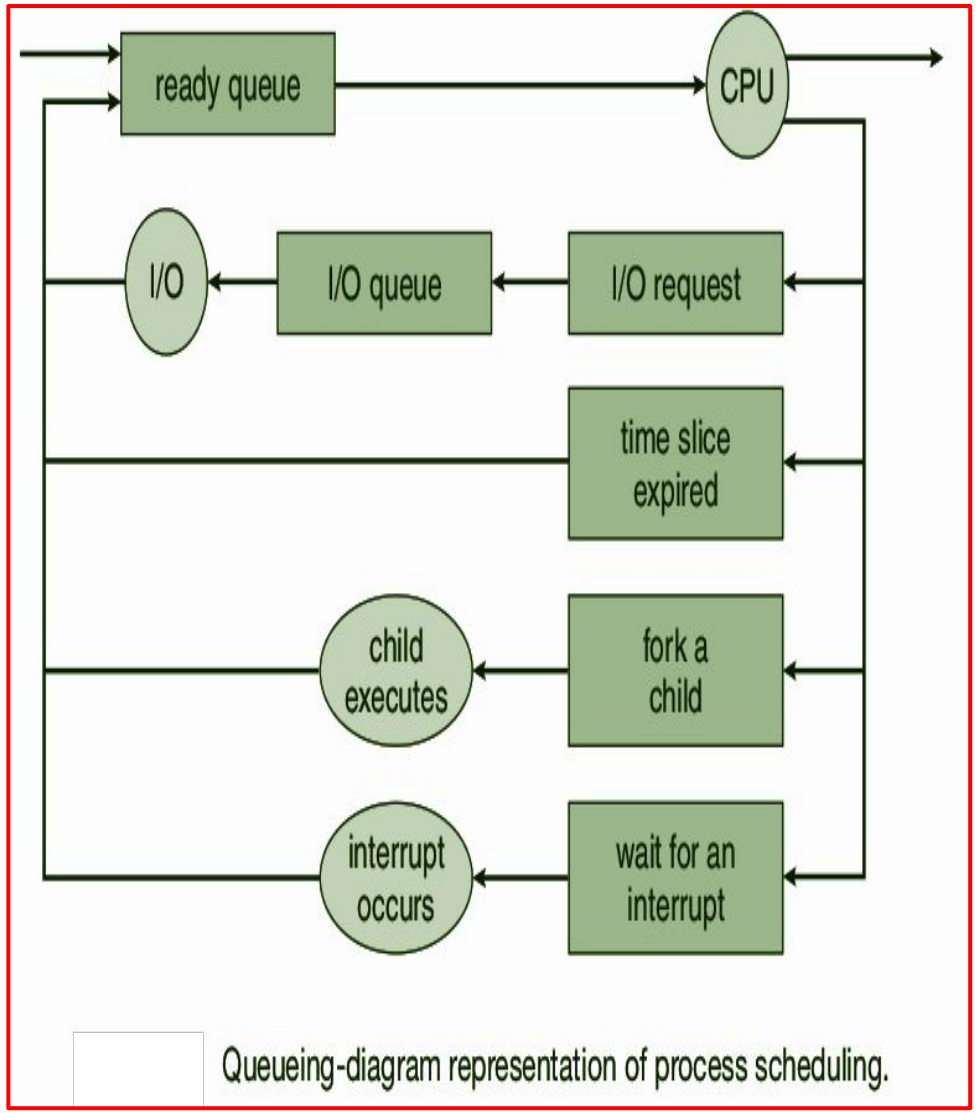


Queueing-diagram representation of process scheduling.

Process Scheduling: Scheduling Queues

- In the case 1 and case 2, the process eventually **switches** from the **waiting state** to the **ready state** and is then put back in the **ready queue.**

- A process continues this cycle until it **terminates**, at which time it is removed from all queues and has its PCB and resources **deallocated**



Queueing-diagram representation of process scheduling.

Topics uncovered in this session

- Process Concept
  - The Process
  - Process State
  - Process Control Block

- Process Scheduling
  - Scheduling Queues
  - Schedulers
  - Context Switch

- Operations on Processes
  - Process Creation
  - Process Termination

- Typical Q and As

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com