Operating System

ACSE0403A

Unit: 2

Process Management

B. Tech 4th Semester

Surbhi Jha

Assistant Professor

Department of CSE

Evaluation Scheme

Sl. No.	Subject Codes	Subject Name	Periods			Evaluation Scheme				End Semester		Total	Credit
			L	T	P	CT	TA	TOTAL	PS	TE	PE		
1	AAS0402	Engineering Mathematics-IV	3	1	0	30	20	50		100		150	4
2	AASL0401	Technical Communication	2	1	0	30	20	50		100		150	3
3	AIT0401	Software Engineering	3	0	0	30	20	50		100		150	3
4	ACSE0403A	Operating Systems	3	0	0	30	20	50		100		150	3
5	ACSE0404	Theory of Automata and Formal Languages	3	0	0	30	20	50		100		150	3
6	ACSAI0402	Database Management Systems	3	1	0	30	20	50		100		150	4
7	AIT0451	Software Engineering Lab	0	0	2				25		25	50	1
8	ACSE0453A	Operating Systems Lab	0	0	2				25		25	50	1
9	ACSAI0452	Database Management Systems Lab	0	0	2				25		25	50	1
10	ACSE0459	Mini Project using Open Technology	0	0	2				50			50	1
11	ANC0402 / ANC0401	Environmental Science*/Cyber Security*(Non Credit)	2	0	0	30	20	50		50		100	0
12		MOOCs** (For B.Tech. Hons. Degree)											
		GRAND TOTAL										1100	24

Subject Syllabus

B. TECH. SECOND YEAR			
Course Code	ACSE0403A	**L T P**	**Credits**
Course Title	**Operating Systems**	**3 0 0**	**3**

Course objective:

The objective of the course is to provide an understanding of the basic modules and architecture of an operating system and the functions of the modules to manage, coordinate and control all the parts of the computer system. This course cover processor scheduling, deadlocks, memory management, process synchronization, system call and file system management.

Pre-requisites:

1. Basic knowledge of computer fundamentals, Data structure and Computer organization.

Course Contents / Syllabus

UNIT-I	Fundamental Concepts of Operating System	8 Hours

Introduction, Functions of Operating System, Characteristics of Operating System, Computer System Structure, Evolution of Operating Systems-Bare Machine, Single Processing, Batch Processing,Multiprogramming,Multitasking,Multithreaded,Interactive, Time sharing, Real Time System, Distributed System, Multiprocessor Systems, Multithreaded Systems, System Calls, System Programs and System Boot, Interrupt Handling, Operating System Structure- Simple structure, Layered Structure, Monolithic, Microkernel and Hybrid, System Components, Operating System Services, Case Studies: Windows, Unix and Linux.

UNIT-II	Process Management	8 Hours

Scheduling Concepts, Performance Criteria, Process States, Process Transition Diagram, Schedulers, Process Control Block (PCB), Process Address Space, Process Identification Information, Threads and their management, Types of Scheduling: Long Term Scheduling, Mid Term Scheduling, Short Term Scheduling, Pre-emptive and Non Pre-emptive Scheduling, Dispatcher, Scheduling Algorithm: FCFS, Non Pre-emptive SJF, Pre-emptive SJF, Non Pre-emptive Priority, Pre-emptive Priority, Round Robin, Multilevel Queue Scheduling and Multilevel Feedback Queue Scheduling.

UNIT-III	Deadlock and Concurrent Processing	8 Hours

Deadlock: System model, Deadlock characterization, Prevention, Avoidance and detection, Recovery from Deadlock,Principle of Concurrency, Process Synchronization, Producer / Consumer Problem, Mutual Exclusion, Critical Section Problem, Peterson's Solution, Lamport Bakery Solution, Semaphores, Test and Set Operation; Critical Section Problems and their solutions - Bound Buffer Problem, Reader-Writer Problem, Dining Philosopher Problem, Sleeping Barber Problem; Inter Process Communication Models and Schemes, Process Generation.

Subject Syllabus

UNIT-IV	**Memory Management**	**8 Hours**
Memory Management function, Address Binding Loading : Compile Time, Load Time and Execution Time, MMU, Types of Linking, Types of Loading, Swapping, Multiprogramming with Fixed Partitions, Multiprogramming with variable partitions, Memory Allocation: Allocation Strategies First Fit, Best Fit, and Worst Fit, Paging, Segmentation, Paged Segmentation, Virtual Memory Concepts, Demand Paging, Performance of Demand Paging, Page Replacement Algorithms: FIFO,LRU, Optimal and LFU, Belady's Anomaly, Thrashing, Cache Memory Organization, Locality of Reference.		
UNIT-V	**I/O Management and Disk Scheduling**	**8 Hours**
I/O Devices, and I/O Subsystems, I/O Buffering, I/O Ports, Disk Storage: Seek Time, Rotational Latency, Data Transfer Time, Average Access Time and Controller Time,Disk Storage Strategies, Disk Scheduling:FCFS, SSTF, SCAN, C-SCAN, LOOK and C-LOOK. Directory and Directory Structure, File		

System: File concept, File Access Mechanism: - Sequential Access, Direct Access and Index Access methods, File Allocation Method: Contiguous, Linked and Indexed, Free Space Management: -Bit Vector, Linked List, Grouping and Counting File System Implementation Issues, File System Protection and Security, RAID.

Syllabus For Unit-2

UNIT-2 Process Management

Scheduling Concepts, Performance Criteria, Process States, Process Transition Diagram, Schedulers, Process Control Block (PCB), Process Address Space, Process Identification Information, Threads and their management, Types of Scheduling: Long Term Scheduling, Mid Term Scheduling, Short Term Scheduling, Preemptive and Non Pre-emptive Scheduling, Dispatcher, Scheduling Algorithm: FCFS, Non Pre-emptive SJF, Preemptive SJF, Non Pre-emptive Priority, Pre-emptive Priority, Round Robin, Multilevel Queue Scheduling and Multilevel Feedback Queue Scheduling.

Branch wise Applications

- Airlines reservation system.
- Air traffic control system.
- Systems that provide immediate updating.
- Used in any system that provides up to date and minute information on stock prices.
- Defense application systems like RADAR.
- Networked Multimedia Systems.
- Command Control Systems.

Course Objectives

- Provide an understanding of the basic modules and architecture of an operating system and the functions of the modules to manage, coordinate and control all the parts of the computer system.
- Processor scheduling, deadlocks, memory management, process synchronization, system call and file system management.

Course Outcomes

Course outcome: After completion of this course students will be able to:

CO 1	Understand the fundamentals of an operating systems, functions and their structure and functions.	K1, K2
CO2	**Implement concept of process management policies, CPU Scheduling and thread management.**	**K5**
CO3	Understand and implement the requirement of process synchronization and apply deadlock handling algorithms.	K2,K5
CO4	Evaluate the memory management and its allocation policies.	K5
CO5	Understand and analyze the I/O management and File systems	K2, K4

Program Outcomes

1. Engineering knowledge
2. Problem analysis
3. Design/development of solutions
4. Conduct investigations of complex problems
5. Modern tool usage
6. The engineer and society
7. Environment and sustainability
8. Ethics:
9. Individual and team work
10. Communication
11. Project management and finance
12. Life-long learning

COs and POs Mapping

OPERATING SYSTEM (ACSE0403A)												
Sub CODE	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ACSE0403A .1	3	3	2	2	1	2	-	2	3	2	2	3
ACSE0403A .2	3	3	3	2	2	3	-	2	3	1	1	3
ACSE0403A .3	**3**	**3**	**2**	**2**	**2**	**2**	**-**	**2**	**2**	**3**	**1**	**3**
ACSE0403A .4	3	2	2	3	1	2	-	1	2	1	2	3
ACSE0403A .5	3	1	2	2	2	2	-	1	2	2	2	3
Average	3	2.4	2.2	2.2	1.6	2.2	-	1.8	2.2	1.8	1.6	3

Program Specific Outcomes(PSOs)

On successful completion of B. Tech. (I.T.) Program, the Information Technology graduates will be able to:

- **PSO1:-** Work as a software developer, database administrator, tester or networking engineer for providing solutions to the real world and industrial problems.
- **PSO2:**- Apply core subjects of information technology related to data structure and algorithm, software engineering, web technology, operating system, database and networking to solve complex IT problems.
- **PSO3:**-Practice multi-disciplinary and modern computing techniques by lifelong learning to establish innovative career.
- **PSO4:-**Work in a team or individual to manage projects with ethical concern to be a successful employee or employer in IT industry.

NIET Greater Noida
GET FUTURE READY
Estd 2001
AN AUTONOMOUS INSTITUTE

COs and PSOs Mapping

Course Outcomes	Program Specific Outcomes			
	PSO1	PSO2	PSO3	PSO4
ACSE0403A .1	**2**	**1**	**2**	**2**
ACSE0403A .2	2	2	1	2
ACSE0403A .3	2	3	3	2
ACSE0403A .4	2	2	1	2
ACSE0403A .5	2	2	2	2
Average	2	2	1.8	2

Program Educational Objectives (PEOs)

- **PEO1:**Apply sound knowledge in the field of information technology to fulfill the needs of IT industry.
- **PEO2:**Design innovative and interdisciplinary systems through latest digital technologies.
- **PEO3:**Inculcate professional – social ethics, team work and leadership for serving the society.
- **PEO4:**Inculcate lifelong learning in the field of computing for successful career in organizations and R&D sectors.

Faculty wise Result Analysis

Semester & Section	Subject Code	Result

End Semester Question Paper Templates (Offline Pattern/Online Pattern

Printed page: ....

Subject Code: ....

Roll No:

NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY ,GREATER NOIDA

(An Autonomous Institute Affiliated to AKTU, Lucknow)

B.Tech/B.Voc./MBA/MCA/M.Tech (Integrated)

(SEM: ..... THEORY EXAMINATION (2020-2021)

Subject ....

Time: 3 Hours

Max. Marks:100

General Instructions:

- All questions are compulsory. Answers should be brief and to the point.
- This Question paper consists of ....pages & ...8.........questions.
- It comprises of three Sections, A, B, and C. You are to attempt all the sections.
- Section A -Question No- 1 is objective type questions carrying 1 mark each, Question No- 2 is very short

End Semester Question Paper Templates (Offline Pattern/Online Pattern

- <u>Section B</u> - Question No-3 is Long answer type -I questions with external choice carrying 6 marks each. You need to attempt any five out of seven questions given.
- <u>Section C</u> - Question No. 4-8 are Long answer type -II (within unit choice) questions carrying 10 marks each. You need to attempt any one part <u>a or b.</u>
- Students are instructed to cross the blank sheets before handing over the answer sheet to the invigilator.
- No sheet should be left blank. Any written material after a blank sheet will not be evaluated/checked.

End Semester Question Paper Templates (Offline Pattern/Online Pattern

		SECTION – A		**CO**
1.	**Attempt all parts-**		**[10×1=10]**	
	1-a.	**Question-**	**(1)**	
	1-b.	**Question-**	**(1)**	
	1-c.	**Question-**	**(1)**	
	1-d.	**Question-**	**(1)**	
	1-e.	**Question-**	**(1)**	
	1-f.	**Question-**	**(1)**	
	1-g.	**Question-**	**(1)**	
	1-h.	**Question-**	**(1)**	
	1-i.	**Question-**	**(1)**	
	1-j.	**Question-**	**(1)**	

2.	Attempt all parts-		[5×2=10]	CO
	2-a.	Question-	(2)	
	2-b.	Question-	(2)	
	2-c.	Question-	(2)	
	2-d.	Question-	(2)	
	2-e.	Question-	(2)	

End Semester Question Paper Templates (Offline Pattern/Online Pattern

SECTION – B				CO
3.	Answer any five of the following-		[5×6=30]	
	3-a.	Question-	(6)	
	3-b.	Question-	(6)	
	3-c.	Question-	(6)	
	3-d.	Question-	(6)	
	3-e.	Question-	(6)	
	3-f.	Question-	(6)	
	3-g.	Question-	(6)	

End Semester Question Paper Templates (Offline Pattern/Online Pattern

SECTION – C				CO
4	Answer any one of the following-		[5×10=50]	
	4-a.	Question-	(10)	
	4-b.	Question-	(10)	
5.	Answer any one of the following-			
	5-a.	Question-	(10)	
	5-b.	Question-	(10)	

End Semester Question Paper Templates (Offline Pattern/Online Pattern

6.	Answer any one of the following-			
	6-a.	Question-	(10)	
	6-b.	Question-	(10)	
7.	Answer any one of the following-			
	7-a.	Question-	(10)	
	7-b.	Question-	(10)	
8.	Answer any one of the following-			
	8-a.	Question-	(10)	
	8-b.	Question-	(10)	

Prerequisite and Recap

Prerequisite

First get your computer hardware basics cleared.

- Digital logic and it's design (Basics will make understand storing memory and page faults, Difference between RAM and ROM. etc.,)
- Computer Organization and Architecture(Design of Computer architecture will help you understand computer peripherals and its storages, accessing them in operating system)
- Strong programming skills (Knowledge of C).

Recap

- To Understood Types of operating system
- To analyzed different structure of Operating system
- To Understood the System calls and Function etc.

Objective of Unit 2

After going through this unit, you should be able to:

- To examine issues related to multithreaded programming.
- To introduce CPU scheduling, which is the basis for multiprogrammed operating systems.
- To describe various CPU-scheduling algorithms.
- To discuss evaluation criteria for selecting a CPU-scheduling algorithm for a particular system.
- To examine the scheduling algorithms of several operating systems.
- understand the concept of Process Concept, Process State and Process Control Block (PCB)
- understand the concept of Threads.
- compare and contrast different CPU scheduling algorithms.

Unit -Recap

- To understand the basic concepts and components of operating systems
- To understand computer system organization and structure.
- To understand operating system services and types of system call.
- Demonstrate the structure and functions of an operating system.
- Compare between different operating systems.

Brief Introduction about the subject with video

An operating system acts as an intermediary between the user of a computer and computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs conveniently and efficiently

YouTube/other Video Links

- https://www.youtube.com/playlist?list=PLmXKhU9FNesSFvj6gASuWmQd23Ul5omtD

Unit-2 Content

- Scheduling Concept
- Performance Criteria
- Process Concept
- Process State
- Process Transition Diagram
- Process Control Block (PCB)
- Thread
  - User thread
  - Kernel thread
- Single and Multithreaded Processes
- Multithreading Models
  - Many-to-One
  - One-to-One
  - Many-to-Many Model

Unit-2 Content

- Types of Schedulers
- CPU Scheduling Algorithms
  - First Come First Serve(FCFS) Scheduling.
  - Shortest-Job-First(SJF) Scheduling.
  - Priority Scheduling.
  - Round Robin(RR) Scheduling.
  - Multilevel Queue Scheduling.
  - Multilevel Feedback Queue Scheduling.
  - Multiple-Processor Scheduling

Unit-2 Objective

After going through this unit, you should be able to:

- understand the concept of Process Concept, Process State
- and Process Control Block (PCB)
- understand the concept of Threads.
- compare and contrast different CPU scheduling algorithms.

Course Outcome of Unit 2

At the end of semester, students will be able to:

CO2: **Implement concept of process management policies, CPU Scheduling and thread management**

CO-PO Mapping

OPERATING SYSTEM(ACSE0403A)												
CODE	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ACSE0403A. 2	3	3	3	2	2	3	2	-2	3	-	1	3

Topic mapping with CO

Topic	CO
Scheduling Concepts, Performance Criteria	CO2
Process States, Process Transition Diagram, Schedulers	CO2
Process Control Block (PCB),	CO2
Process Address Space, Process Identification Information	CO2
Threads and their management	CO2

Topic mapping with CO

Topic	CO
Types of Scheduling: Long Term Scheduling, Mid Term Scheduling, Short Term Scheduling	CO2
Preemptive and Non Pre-emptive Scheduling, Dispatcher	CO2
Dispatcher, Scheduling Algorithm	CO2
FCFS, Non Pre-emptive SJF	CO2
Preemptive SJF, Non Pre-emptive Priority	CO2

Topic mapping with CO

Topic	CO
Pre-emptive Priority	CO2
Round Robin	CO2
Multilevel Queue Scheduling	CO2
Multilevel Feedback Queue Scheduling.	CO2

Topic Objectives

Topic	Objective
	Students will be able to
Process Management	Understand the Process
Types of Scheduling: Long Term Scheduling, Mid Term Scheduling, Short Term Scheduling	Understand the different types of scheduling
Scheduling Algorithm	Understand the Scheduling Concept
Non Pre-emptive Priority, Pre-emptive Priority, Round Robin	Understand the different types of Pre-emptive and non pre-emptive algorithm
Multilevel Queue Scheduling	Know the different types of multilevel queue scheduling

Introduction (CO2)

Process management involves various tasks like creation, scheduling, termination of processes, and a dead lock. Process is a program that is under execution, which is an important part of modern-day operating systems. The OS must allocate resources that enable processes to share and exchange information. It also protects the resources of each process from other methods and allows synchronization among processes.

The scheduling is the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process on the basis of a particular strategy.

For video lecture: https://www.youtube.com/watch?v=zWtFoPL8BYg

Introduction (CO2)

Program

- It is a set of instructions that has been designed to complete a certain task.
- It is a passive entity.
- It resides in the secondary memory of the system.
- It exists in a single place and continues to exist until it has been explicitly deleted.
- It is considered as a static entity.
- It doesn’t have a resource requirement.
- It requires memory space to store instructions.
- It doesn’t have a control block.

Process

- It is an instance of a program that is being currently executed.
- It is an active entity.
- It is created when program is in execution and is being loaded into the main memory.
- It exists for a limited amount of time.
- It gets terminated once the task has been completed.
- It is considered as a dynamic entity.
- It has a high resource requirement.
- It requires resources such as CPU, memory address, I/O during its working.
- It has its own control block, which is known as Process Control Block.

Scheduling Concepts(CO2)

Various criteria or characteristics that help in designing a good scheduling are:

CPU Utilization − A scheduling algorithm should be designed so that CPU remains busy as possible. It should make efficient use of CPU.

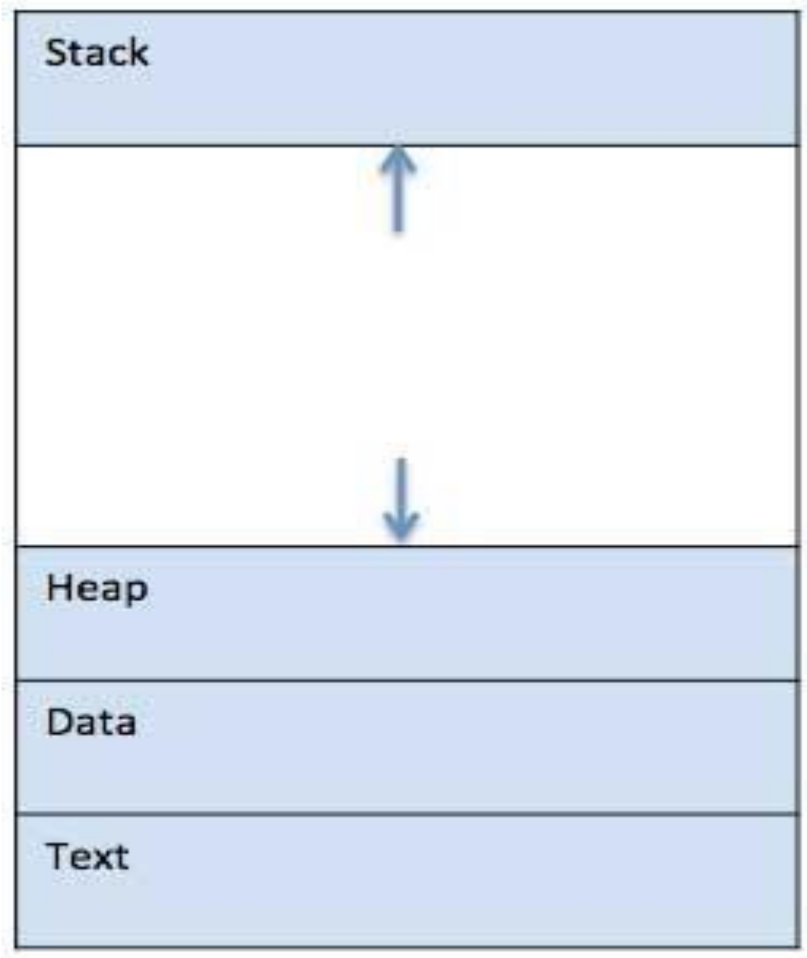
Throughput − Throughput is the amount of work completed in a unit of time.

- **Response time** − Response time is the time taken to start responding to the request. A scheduler must aim to minimize response time for interactive users.
- **Turnaround time** − Turnaround time refers to the time between the moment of submission of a job/ process and the time of its completion. Thus how long it takes to execute a process is also an important factor.
- **Waiting time** − It is the time a job waits for resource allocation when several jobs are competing in multiprogramming system. The aim is to minimize the waiting time.
- **Fairness** − A good scheduler should make sure that each process gets its fair share of the CPU.

Process (CO2)

- A process is basically a program in execution. The execution of a process must progress in a sequential fashion.
- A process is defined as an entity which represents the basic unit of work to be implemented in the system.
- To put it in simple terms, we write our computer programs in a text file and when we execute this program, it becomes a process which performs all the tasks mentioned in the program.
- When a program is loaded into the memory and it becomes a process, it can be divided into four sections – stack, heap, text and data. The following image shows a simplified layout of a process inside main memory .

Process (CO2)

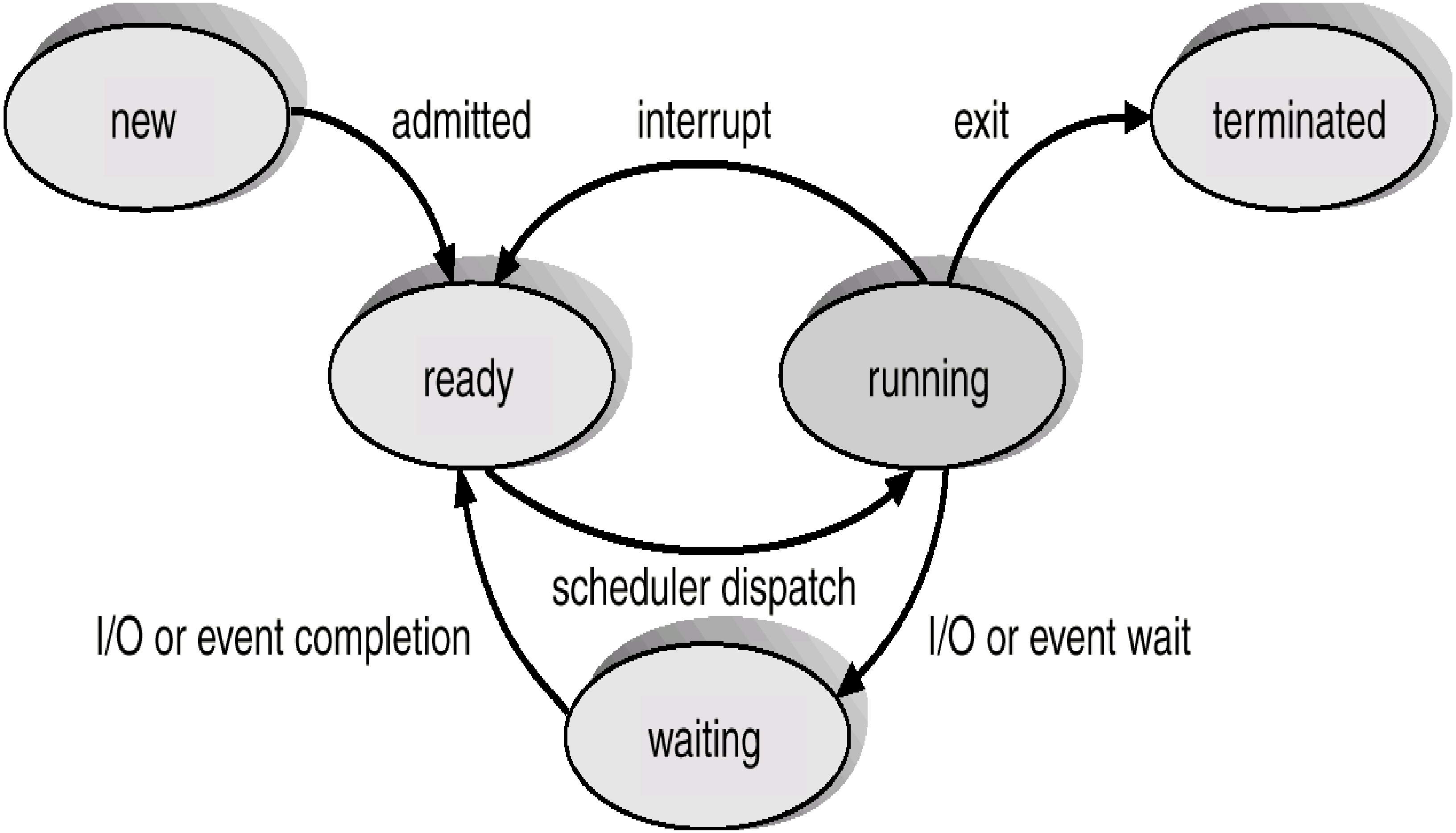


Process State(CO2)

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 Transition Diagram(CO2)

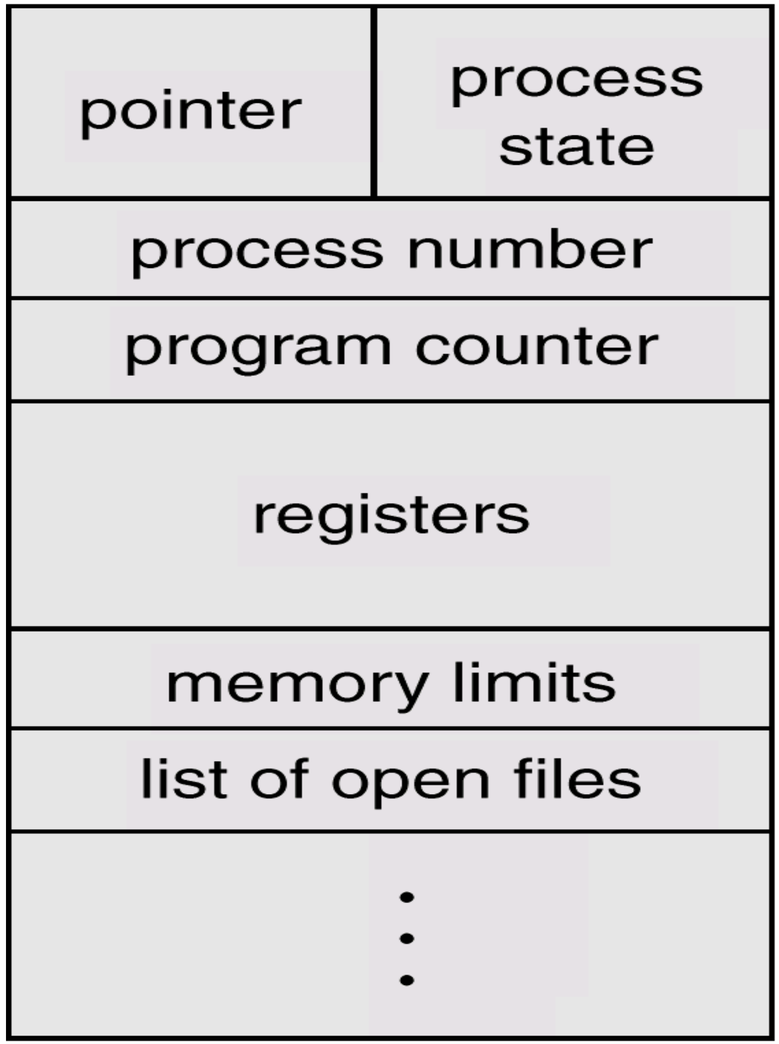


Process Control Block (PCB)(CO2)

Information associated with each process.

- Process ID
- Process state
- Program counter
- CPU registers
- CPU scheduling information
- Memory-management information
- Accounting information
- I/O status information

Process Control Block (PCB)(CO2)



Process Control Block (PCB)(CO2)

Process State

This specifies the process state i.e. new, ready, running, waiting or terminated.

Process Number

This shows the number of the particular process.

Program Counter

This contains the address of the next instruction that needs to be executed in the process.

Registers

This specifies the registers that are used by the process. They may include accumulators, index registers, stack pointers, general purpose registers etc.

List of Open Files

These are the different files that are associated with the process

Process Control Block (PCB)(CO2)

CPU Scheduling Information

The process priority, pointers to scheduling queues etc. is the CPU scheduling information that is contained in the PCB. This may also include any other scheduling parameters.

Memory Management Information

The memory management information includes the page tables or the segment tables depending on the memory system used. It also contains the value of the base registers, limit registers etc.

I/O Status Information

This information includes the list of I/O devices used by the process, the list of files etc.

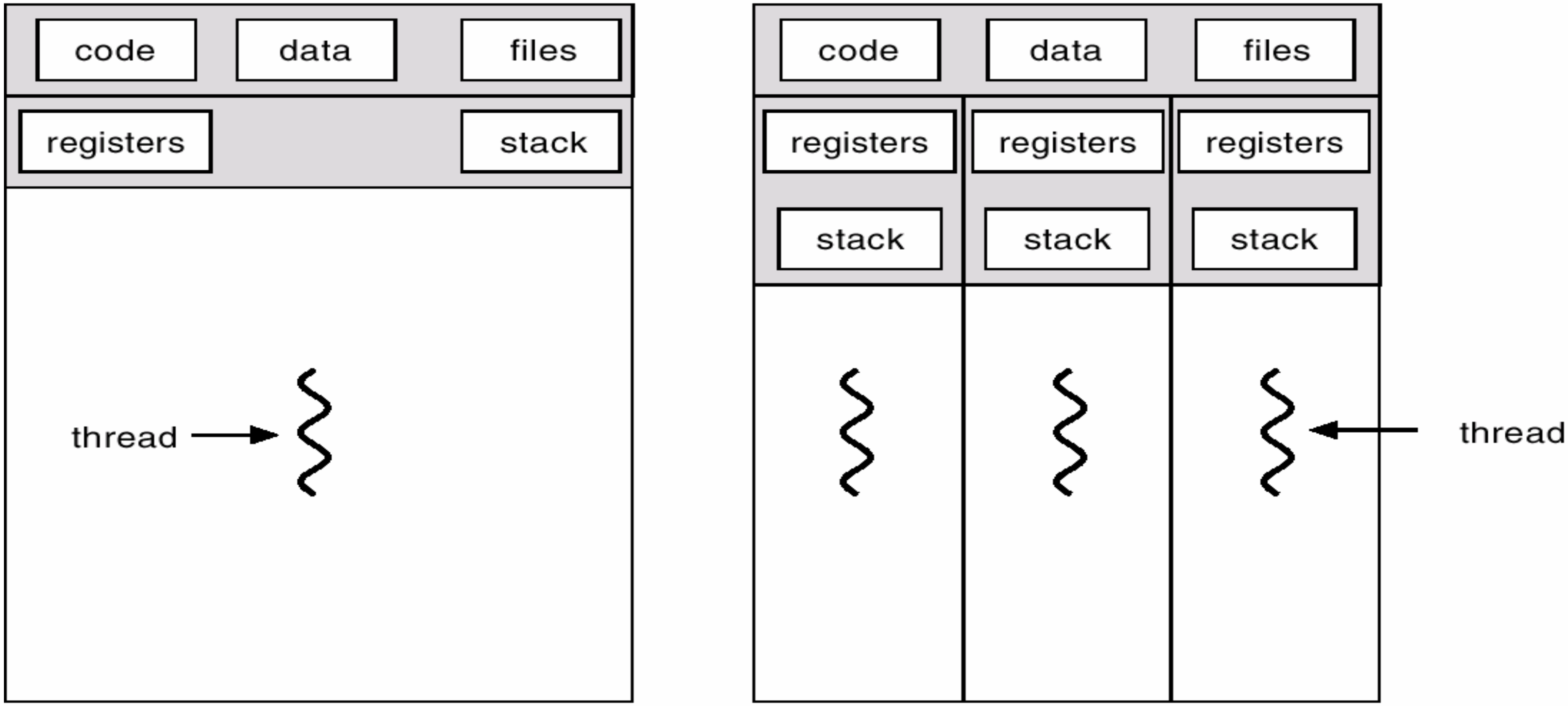
Accounting information

The time limits, account numbers, amount of CPU used, process numbers etc. are all a part of the PCB accounting information.

Thread(CO2)

- A thread is a basic unit of CPU utilization
- Thread is an execution unit which consists of its own program counter, a stack, and a set of registers.
- Threads are also known as Lightweight processes
- Threads are popular way to improve application through parallelism.
- As each thread has its own independent resource for process execution, multpile processes can be executed parallely by increasing number of threads.

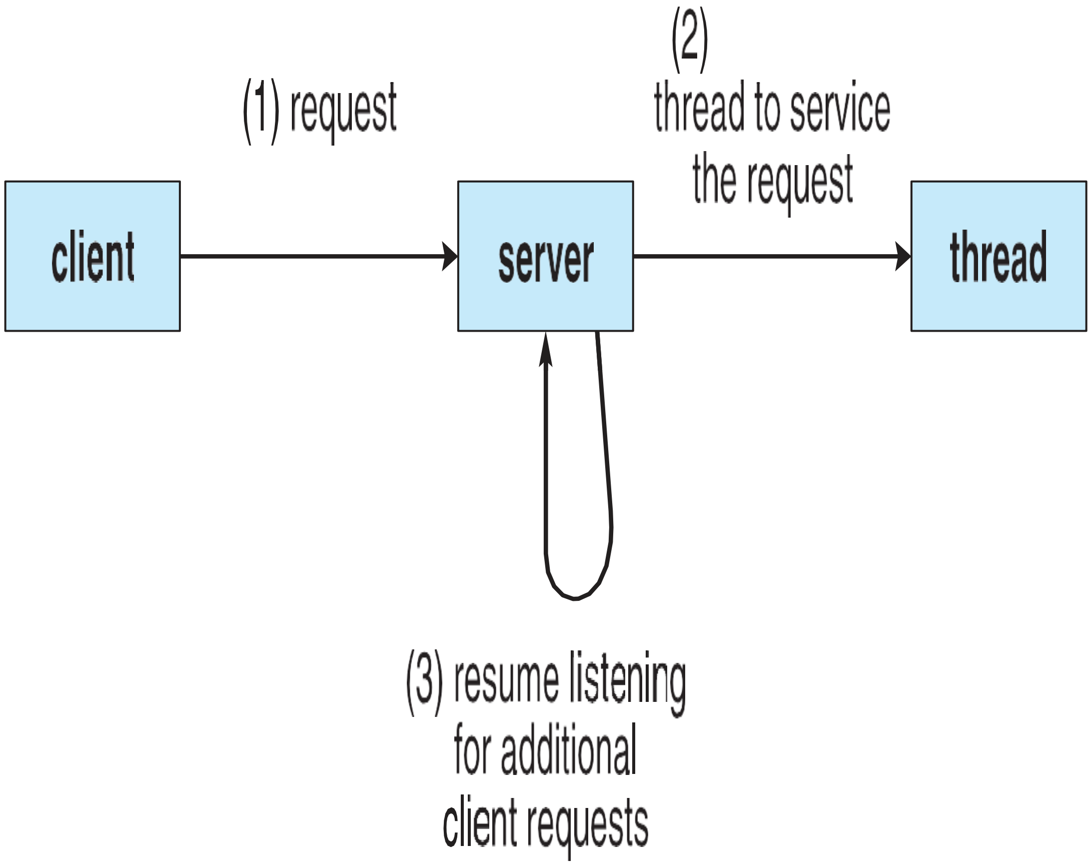
Single and Multithreaded Processes(CO2)



single-threaded

multithreaded

Multithreaded Server Architecture (CO2)



Benefits of Multithreading(CO2)

- **Responsiveness –** may allow continued execution if part of process is blocked, especially important for user interfaces.
- **Resource Sharing –** threads share resources of process, easier than shared memory or message passing.
- **Economy –** cheaper than process creation, thread switching has lower overhead than context switching.
- **Scalability –** process can take advantage of multiprocessor architectures.

User Threads(CO2)

User Threads

- Thread management done by user-level threads library
- No need for kernel intervention
- Drawback : all may run in single process. If one blocks, all block.
- Examples

  •POSIX Pthreads

  •Mach C-threads

  •Solaris threads

Kernel Threads

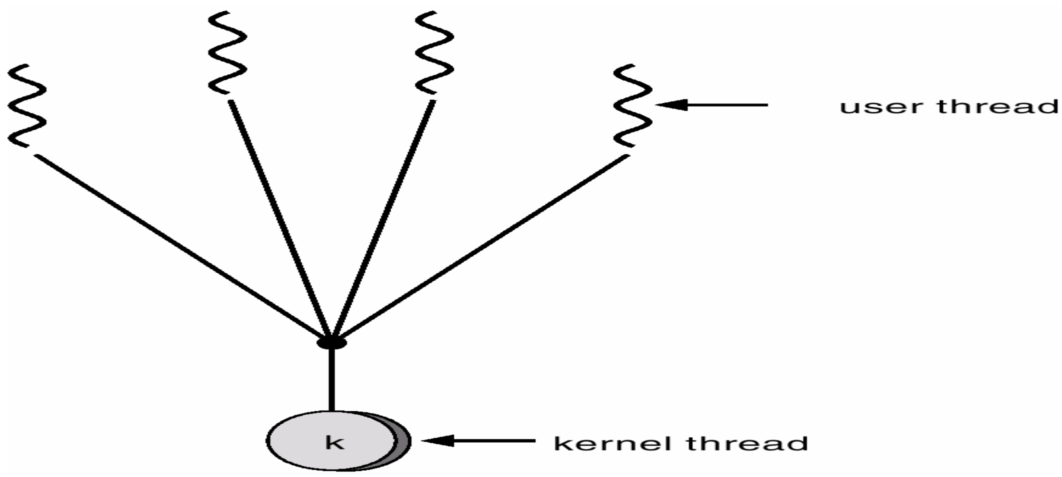
- Supported by the Kernel
- Generally slower to create than user threads
- If one blocks another in the application can be run
- Can be scheduled on different CPUs in multiprocessor
- Examples
  - Windows 95/98/NT/2000
  - Solaris
  - Tru64 UNIX
  - BeOS
  - Linux

Multithreading Models(CO2)

- Many-to-One
- One-to-One
- Many-to-Many

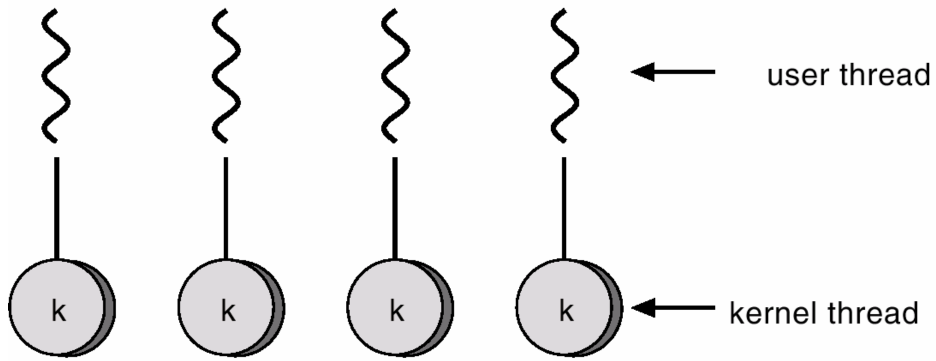
Many-to-One(CO2)

- Many user-level threads mapped to single kernel thread.
- Used on systems that do not support kernel threads.



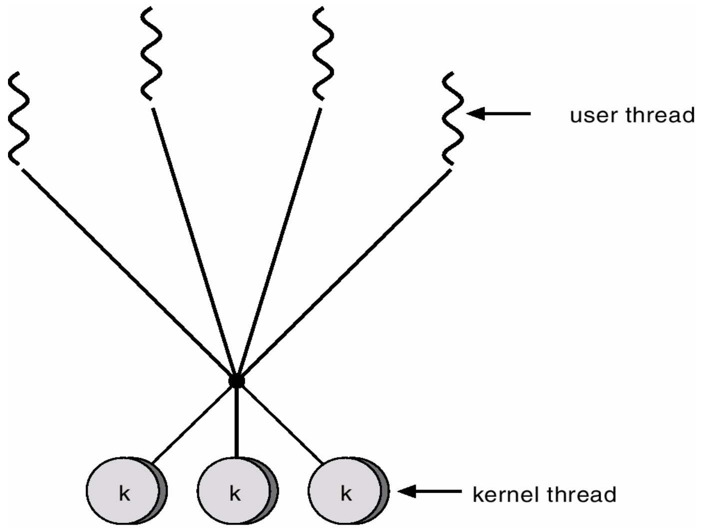
One-to-One(CO2)

- Each user-level thread maps to kernel thread.
- Examples
  - Windows 95/98/NT/2000
  - OS/2



Many-to-Many Model(CO2)

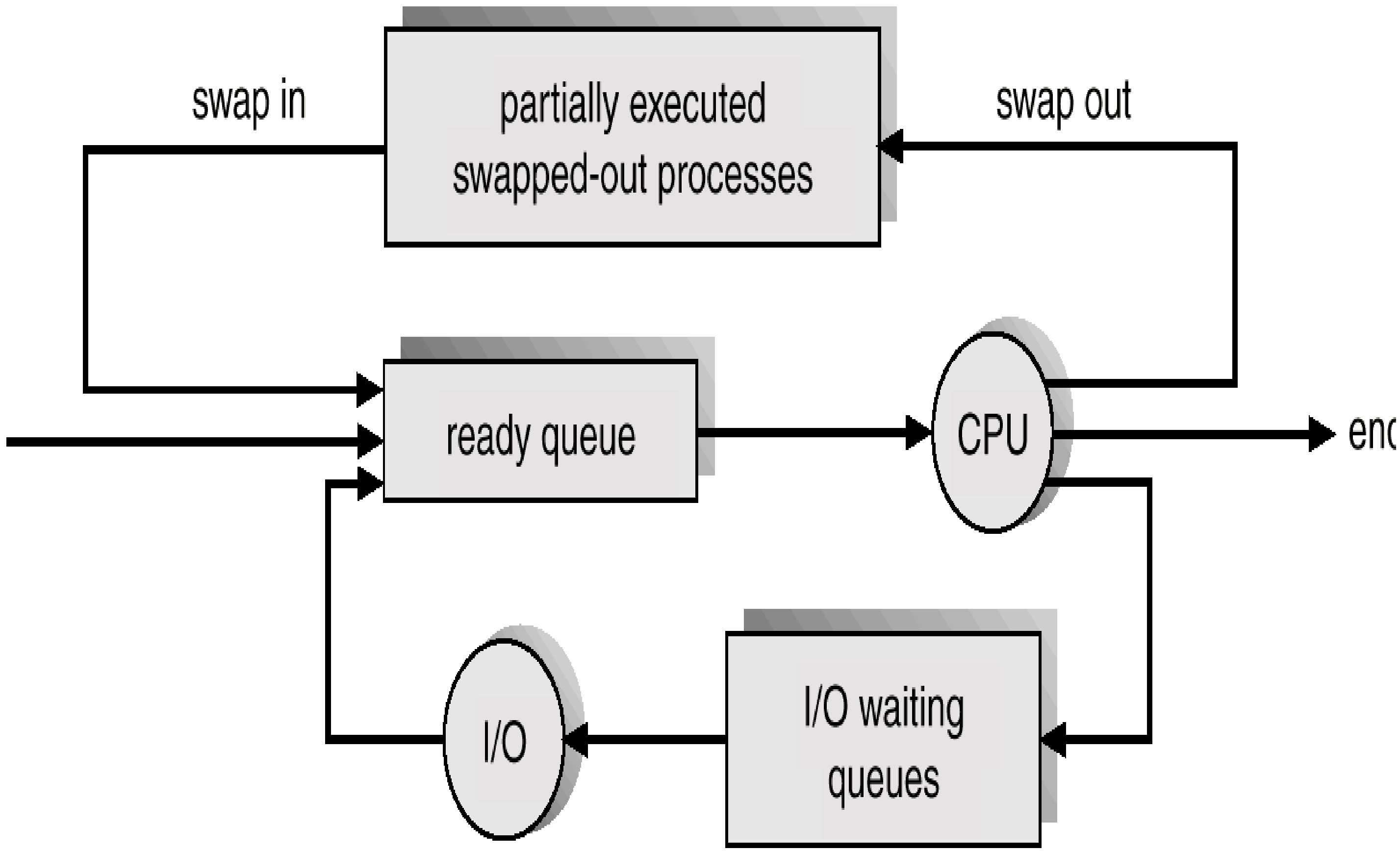
- Many user-level threads mapped to same or lesser number of kernel level threads.
- If a user thread is blocked, other user threads can be scheduled to other kernel threads. Thus system doesn’t block if a particular thread is blocked.
- Multiple threads may not run in parallel on muti-core system because only one may be in kernel at a time
- Few systems currently use this model
- Examples:
    - **Solaris Green Threads**
    - **GNU Portable Threads**



Types of Schedulers(CO2)

- Long-term scheduler (or job scheduler) – selects which processes should be brought into the ready queue. However, the main goal of this type of scheduler is to offer a balanced mix of jobs, like Processor, I/O jobs., that allows managing multiprogramming.
- Short-term scheduler (or CPU scheduler) – selects which process should be executed next and allocates CPU. The dispatcher gives control of the CPU to the process selected by the short term scheduler.
- Medium term scheduler – Medium-term scheduling is an important part of **swapping**. It enables you to handle the swapped out-processes.
  In this scheduler, a running process can become suspended, which makes an I/O request.
  A running process can become suspended if it makes an I/O request.
  A suspended process can't make any progress towards completion.
  In order to remove the process from memory and make space for other processes, the suspended process should be moved to secondary storage.

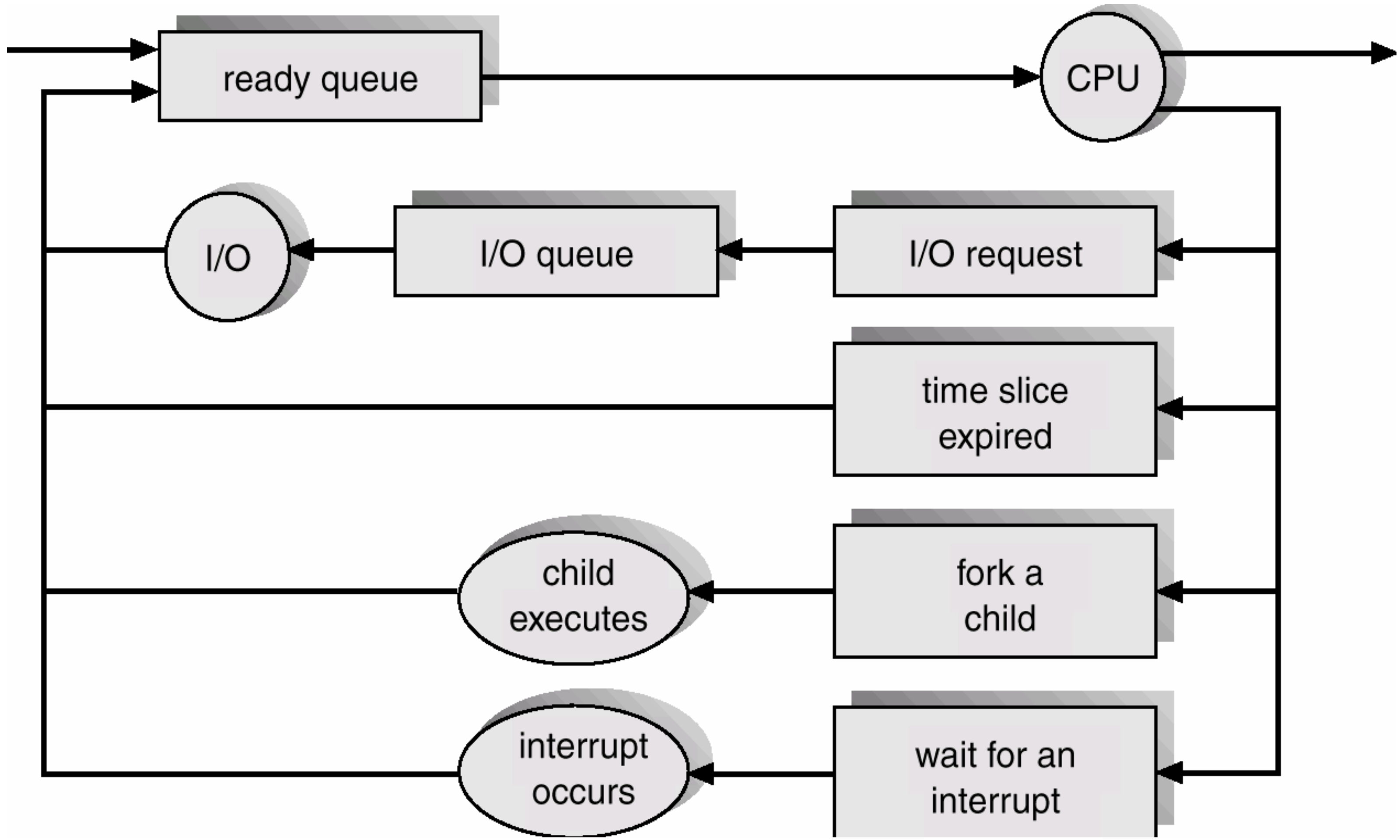
Addition of Medium Term Scheduling(CO2)



Process Scheduling Queues(CO2)

- 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 queue – set of processes waiting for an I/O device.
- Processes migrate between the various queues

Representation of Process Scheduling(CO2)



CPU Scheduler(CO2)

Selects from among the processes in memory (i.e. Ready Queue) that are ready to execute, and allocates the CPU to one of them.

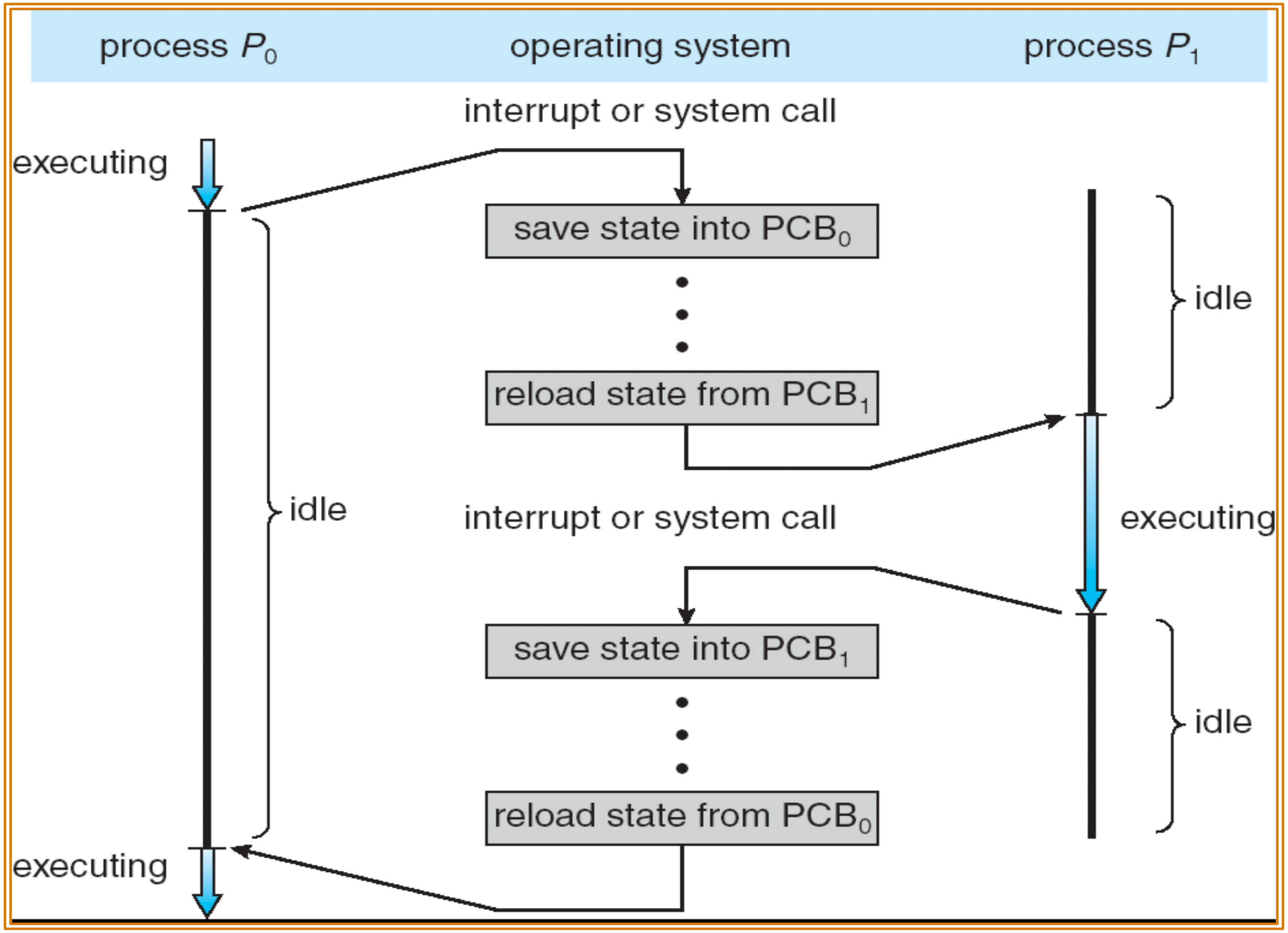
1. Preemptive: allows a process to be interrupted in the midst of its CPU execution, taking the CPU away to another process

2. Non- Preemptive: ensures that a process relinquishes control of CPU when it finishes with its current CPU burst

CPU Scheduler(CO2)

- CPU scheduling decisions may take place when a process:
  1. Switches from running to waiting state.
  2. Switches from running to ready state.
  3. Switches from waiting to ready.
  4. Terminates.
- Preemptive: allows a process to be interrupted
- Non- Preemptive: allows a process finishes with its current CPU burst
- Scheduling under 1 and 4 is nonpreemptive.
- All other scheduling is preemptive.

Context Switching(CO2)



Dispatcher(CO2)

- Dispatcher module gives control of the CPU to the process selected by the short-term scheduler; this involves:
  - Context switching
  - switching to user mode
  - jumping to the proper location in the user program to restart that program
- *Dispatch latency* – time it takes for the dispatcher to stop one process and start to run another process.

Daily Quiz

1. What is Process? Explain PCB block.
2. What is difference between job and process.
3. What is process scheduler? Explain Characteristics of good process scheduler.
4. What do you mean by threads? Explain its types.

Daily Quiz

5. A Process Control Block(PCB) does not contain which of the following?

a) Code

b) Stack

c) Bootstrap program

d) Data

6. The number of processes completed per unit time is known as

a) Output

b) Throughput

c) Efficiency

d) capacity

Topic mapping with CO

Topic	CO
Scheduling Criteria	**CO2**
CPU Scheduling algorithms	**CO2**

Topic Objectives

Topic	Objective
	Students will be able to
Scheduling Criteria	Understand different types of scheduling.
CPU Scheduling algorithms	Know about the how CPU Scheduling algorithms works.

Scheduling Criteria(CO2)

- CPU utilization – keep the CPU as busy as possible
- Throughput – Number of processes that complete their execution per time unit
- Turnaround time – amount of time to execute a particular process (finishing time – arrival time)
- Waiting time – amount of time a process has been waiting in the ready queue
- Response time – amount of time it takes from when a request was submitted until the first response is produced, **not** output(for time-sharing environment)

CPU Scheduling algorithms(CO2)

CPU Scheduling Algorithms

1. First Come First Serve(FCFS) Scheduling.
2. Shortest-Job-First(SJF) Scheduling.
3. Priority Scheduling.
4. Round Robin(RR) Scheduling.
5. Multilevel Queue Scheduling.
6. Multilevel Feedback Queue Scheduling.

CPU Scheduling algorithms(CO2)

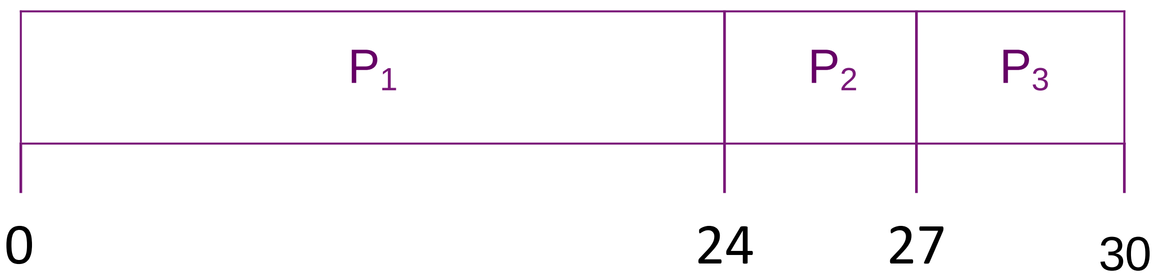
First Come First Serve(FCFS) Scheduling

- In FCFS, the process which arrives first in front of CPU will be executed first by the CPU.
- It is a non-preemptive type of scheduling algorithm, i.e. in this scheduling algorithm priority of processes does not matter, or whatever the priority of the process is, the process will be executed in the manner they arrived in front of the CPU.

First-Come, First-Served (FCFS) Scheduling(CO2)

Process	Burst Time
P_1 P_2 P_3	24
	3
	3

Suppose that the processes arrive in the order: P_1 , P_2 , P_3.
The Gantt Chart for the schedule is:

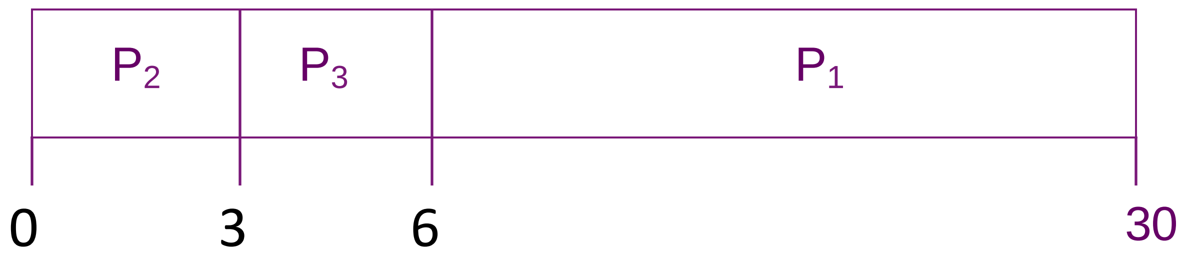


Waiting time for $P_1 = 0$; $P_2 = 24$; $P_3 = 27$

Average waiting time: $(0 + 24 + 27)/3 = 17$

FCFS Scheduling (CO2)

Suppose that the processes arrive in the order
P_2 , P_3 , P_1 .
The Gantt chart for the schedule is:



Waiting time for $P_1 = 6$; $P_2 = 0$, $P_3 = 3$

Average waiting time: $(6 + 0 + 3)/3 = 3$

Much better than previous case.

Convoy effect short process behind long process

FCFS Scheduling (CO2)

Characteristics of FCFS method:

- It offers non-preemptive scheduling algorithm.
- Jobs are always executed on a first-come, first-serve basis
- It is easy to implement and use.
- However, this method is poor in performance, and the general wait time is quite high.

Problems with FCFS Scheduling (CO2)

Below we have a few shortcomings or problems with the FCFS scheduling algorithm:

- It is **Non Pre-emptive** algorithm, which means the **process priority** doesn't matter. If a process with very least priority is being executed, more like **daily routine backup** process, which takes more time, and all of a sudden some other high priority process arrives, like **interrupt to avoid system crash**, the high priority process will have to wait, and hence in this case, the system will crash, just because of improper process scheduling.
- Not optimal Average Waiting Time.
- Resource utilization in parallel is not possible, which leads to **Convoy Effect**, and hence poor resource(CPU, I/O, etc.) utilization.

Shortest-Job-First (SJF) Scheduling(CO2)

This algorithm associates with each process the length of the process's next CPU burst. When the CPU is available, it is assigned to the process that has the smallest next CPU burst. If the next CPU bursts of two processes are the same, FCFS scheduling is used to break the tie.

Shortest-Job-First (SJR) Scheduling(CO2)

- Two schemes:
  - Non-preemptive – once CPU given to the process it cannot be preempted until process completes its CPU burst.
  - Preemptive – if a new process arrives with CPU burst length less than remaining time of current executing process, preempt. This scheme is know as the Shortest-Remaining-Time-First (SRTF).
- SJF is optimal – gives minimum average waiting time for a given set of processes.

Example of Non-Preemptive SJF(CO2)

PROCESS	BURST TIME
P1	21
P2	3
P3	6
P4	2

In Shortest Job First Scheduling, the shortest Process is executed first.

Hence the GANTT chart will be following :

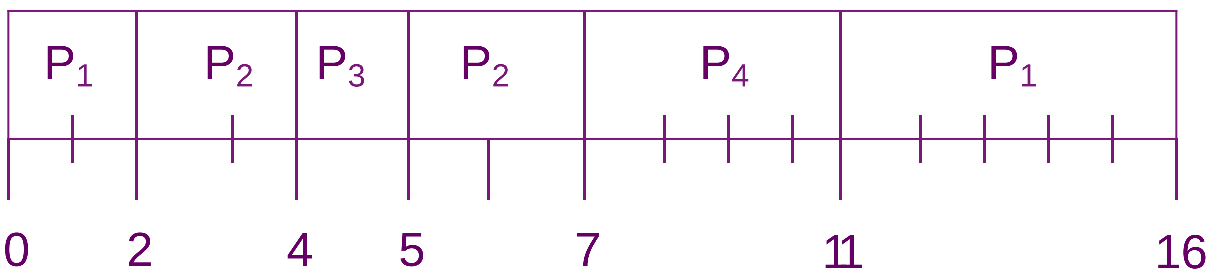
P4	P2	P3	P1	
0	2	5	11	32

Now, the average waiting time will be = (0 + 2 + 5 + 11)/4 = 4.5 ms

Example of Preemptive SJF(CO2)

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

SJF (preemptive)



Average waiting time = (9 + 1 + 0 +2)/4 = 3

Priority Scheduling(CO2)

- Priority scheduling is a method of scheduling processes based on priority. In this method, the scheduler selects the tasks to work as per the priority.
- Priority scheduling also helps OS to involve priority assignments. The processes with higher priority should be carried out first, whereas jobs with equal priorities are carried out on a round-robin or FCFS basis.
- Priority can be decided based on memory requirements, time requirements, etc.

Priority Scheduling(CO2)

Types of Priority Scheduling Algorithm:-

1.Preemptive Priority Scheduling: If the new process arrived at the ready queue has a higher priority than the currently running process, the CPU is preempted, which means the processing of the current process is stopped and the incoming new process with higher priority gets the CPU for its execution.

2.Non-Preemptive Priority Scheduling: In case of non-preemptive priority scheduling algorithm if a new process arrives with a higher priority than the current running process, the incoming process is put at the head of the ready queue, which means after the execution of the current process it will be processed.

Priority Scheduling (Non-preemptive) Example (CO2)

Process	Arrival time	Burst time	Priority
P1	0 ms	5 ms	1
P2	1 ms	3 ms	2
P3	2 ms	8 ms	1
P4	3 ms	6 ms	3

NOTE: In this example, we are taking higer priority number as higher priority.

Gantt Chart

P1	P4	P2	P3
0ms 5ms	5ms 11ms	11ms 14ms	14ms 22ms

Priority Scheduling (Non-preemptive) Example (CO2)

```
---------------------------------------------
| Process | Waiting Time | Turnaround Time  |
---------------------------------------------
|    P1    |      0ms      |       5ms        |
|    P2    |      10ms     |       13ms       |
|    P3    |      12ms     |       20ms       |
|    P4    |      2ms      |       8ms        |
---------------------------------------------
Total waiting time: (0 + 10 + 12 + 2) = 24ms
Average waiting time: (24/4) = 6ms

Total turnaround time: (5 + 13 + 20 + 8) = 46ms
Average turnaround time: (46/4) = 11.5ms
```

Priority Scheduling(CO2)

Advantages of priority scheduling (non-preemptive):

- Higher priority processes like system processes are executed first.

Disadvantages of priority scheduling (non-preemptive):

- It can lead to starvation if only higher priority process comes into the ready state.
- If the priorities of more two processes are the same, then we have to use some other scheduling algorithm.

Priority Scheduling(CO2)

Problem with Priority Scheduling Algorithm

- In priority scheduling algorithm, the chances of **indefinite blocking** or **starvation**.
- A process is considered blocked when it is ready to run but has to wait for the CPU as some other process is running currently.
- But in case of priority scheduling if new higher priority processes keeps coming in the ready queue then the processes waiting in the ready queue with lower priority may have to wait for long durations before getting the CPU for execution.

Priority Scheduling(CO2)

Using Aging Technique with Priority Scheduling

- To prevent starvation of any process, we can use the concept of **aging** where we keep on increasing the priority of low-priority process based on the its waiting time.
- For example, if we decide the aging factor to be **0.5** for each day of waiting, then if a process with priority **20**(which is comparatively low priority) comes in the ready queue. After one day of waiting, its priority is increased to **19.5** and so on. Doing so, we can ensure that no process will have to wait for indefinite time for getting CPU time for processing.

Round Robin (CO2)

Round Robin(RR) scheduling algorithm is mainly designed for time-sharing systems. This algorithm is similar to FCFS scheduling, but in Round Robin(RR) scheduling, preemption is added which enables the system to switch between processes.

- A fixed time is allotted to each process, called a **quantum**, for execution.
- Once a process is executed for the given time period that process is preempted and another process executes for the given time period.
- Context switching is used to save states of preempted processes.
- This algorithm is simple and easy to implement
- It is important to note here that the length of time quantum is generally from 10 to 100 milliseconds in length.

Example of RR (CO2)

Process	Arrival time	Burst time
P1	0 ms	10 ms
P2	0 ms	5 ms
P3	0 ms	8 ms

Gantt Chart

P1	P2	P3	P1	P2	P3
1 2	2 4	4 6	6 8	8 10	10 12

P1	P2	P3	P1	P3	P1
12 14	14 15	15 17	17 19	19 21	21 23

Example of RR (CO2)

```
--------------------------------------------------
| Process | Waiting Time | Turnaround Time  |
--------------------------------------------------
|    P1      |       13ms        |        23ms            |
|    P2      |       10ms        |        15ms            |
|    P3      |       13ms        |        21ms            |
--------------------------------------------------
Total waiting time: (13 + 10 + 13) = 36ms
Average waiting time: (36/3) = 12ms

Total turnaround time: (23 + 15 + 21) = 59ms
Average turnaround time: (59/3) = 19.66ms
```

Advantages and Disadvantages of RR (CO2)

Advantages of round-robin:

- No starvation will be there in round-robin because every process will get chance for its execution.
- Used in time-sharing systems.

Disadvantages of round-robin:

- We have to perform a lot of context switching here, which will keep the CPU idle.

Multilevel Queue(CO2)

- Ready queue is partitioned into separate queues: foreground (interactive)

background (batch)

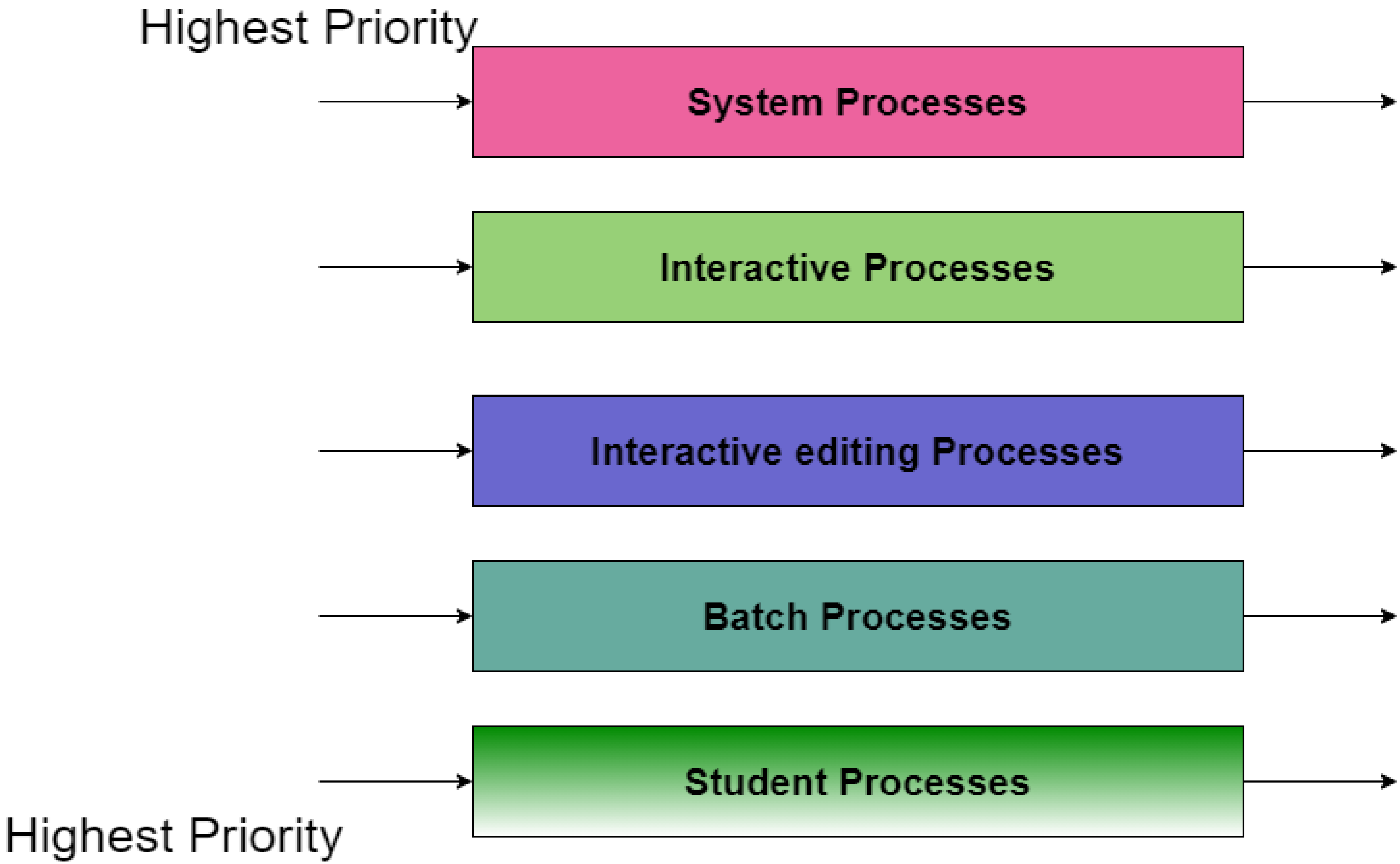
- Each queue has its own scheduling algorithm,

  foreground – RR

  background – FCFS

- Scheduling must be done between the queues.
  - Fixed priority scheduling; (i.e., serve all from foreground then from background). Possibility of starvation.
  - Time slice – each queue gets a certain amount of CPU time which it can schedule amongst its processes; i.e., 80% to foreground in RR
  - 20% to background in FCFS

Multilevel Queue Scheduling(CO2)



Multilevel Queue Scheduling(CO2)

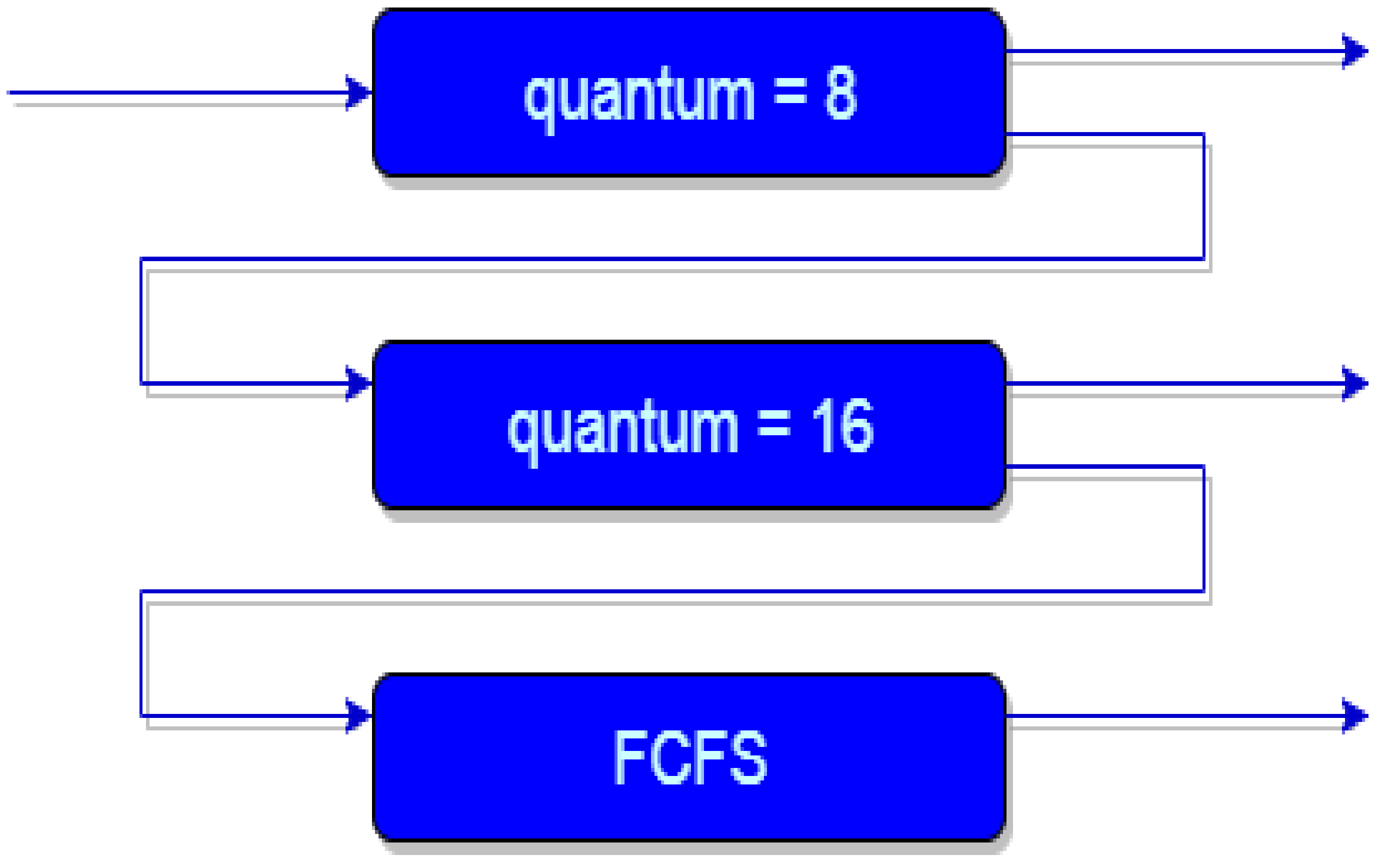
- **System Process** The Operating system itself has its own process to run and is termed as System Process.
- **Interactive Process** The Interactive Process is a process in which there should be the same kind of interaction (basically an online game).
- **Batch Processes** Batch processing is basically a technique in the Operating system that collects the programs and data together in the form of the **batch** before the **processing** starts.
- **Student Process** The system process always gets the highest priority while the student processes always get the lowest priority.

Multilevel Feedback Queue(C03)

In a multilevel queue-scheduling algorithm, processes are permanently assigned to a queue on entry to the system. Processes do not move between queues. In general, a multilevel feedback queue scheduler is defined by the following parameters:

- The number of queues.
- The scheduling algorithm for each queue.
- The method used to determine when to upgrade a process to a higher-priority queue.
- The method used to determine when to demote a process to a lower-priority queue.
- The method used to determine which queue a process will enter when that process needs service.

Multilevel Feedback Queues(CO2)



Example of Multilevel Feedback Queues(CO2)

- Three queues:
  - Q_0 – time quantum 8 milliseconds
  - Q_1 – time quantum 16 milliseconds
  - Q_2 – FCFS
- Scheduling
  - A new job enters queue Q_0 which is served FCFS. When it gains CPU, job receives 8 milliseconds. If it does not finish in 8 milliseconds, job is moved to queue Q_1.
  - At Q_1 job is again served FCFS and receives 16 additional milliseconds. If it still does not complete, it is preempted and moved to queue Q_2.

Need of Multilevel Feedback Queues(CO2)

- This scheduling is more flexible than Multilevel queue scheduling.
- This algorithm helps in reducing the response time.
- In order to optimize the turnaround time, the SJF algorithm is needed which basically requires the running time of processes in order to schedule them. As we know that the running time of processes is not known in advance. Also, this scheduling mainly runs a process for a time quantum and after that, it can change the priority of the process if the process is long. Thus this scheduling algorithm mainly learns from the past behavior of the processes and then it can predict the future behavior of the processes.

Advantages of Multilevel Feedback Queues(CO2)

Advantages of MFQS

- This is a flexible Scheduling Algorithm
- This scheduling algorithm allows different processes to move between different queues.
- In this algorithm, A process that waits too long in a lower priority queue may be moved to a higher priority queue which helps in preventing starvation.

Disadvantages of Multilevel Feedback Queues(CO2)

Disadvantages of MFQS

- This algorithm is too complex.
- As processes are moving around different queues which leads to the production of more CPU overheads.
- In order to select the best scheduler this algorithm requires some other means to select the values.

Multiple-Processor Scheduling(CO2)

- CPU scheduling more complex when multiple CPUs are available.
- Homogeneous processors within a multiprocessor.
- Load sharing or load balancing

Real-Time Scheduling(CO2)

- Hard real-time systems – required to complete a critical task within a guaranteed amount of time.
- Soft real-time computing – requires that critical processes receive priority over less fortunate ones.

Daily Quiz for Module (CPU Scheduling)

1. Which module gives control of the CPU to the process selected by the short-term scheduler?
   a) dispatcher
   b) interrupt
   c) scheduler
   d) none of the mentioned.

2. Which scheduling algorithm allocates the CPU first to the process that requests the CPU first?
   a) first-come, first-served scheduling
   b) shortest job scheduling
   c) priority scheduling
   d) none of the mentioned

3. In priority scheduling algorithm ______________
   a) CPU is allocated to the process with highest priority
   b) CPU is allocated to the process with lowest priority
   c) Equal priority processes can not be scheduled
   d) None of the mentioned

Daily Quiz for Module (CPU Scheduling)

4. In priority scheduling algorithm, when a process arrives at the ready queue, its priority is compared with the priority of ______________
   a) all process
   b) currently running process
   c) parent process
   d) init process
5. Which algorithm is defined in Time quantum?
   a) shortest job scheduling algorithm
   b) round robin scheduling algorithm
   c) priority scheduling algorithm
   d) multilevel queue scheduling algorithm

Process are classified into different groups in _____________
   a) shortest job scheduling algorithm
   b) round robin scheduling algorithm
   c) priority scheduling algorithm
   d) multilevel queue scheduling algorithm

Daily Quiz

1. A solution to the problem of indefinite blockage of low – priority processes is _______________

A. Starvation
B. Wait queue
C. Ready queue
D. Aging

2. The switching of the CPU from one process or thread to another is called _______________

A. process switch
B. task switch
C. context switch
D. all of the mentioned

Daily Quiz

3. The processes that are residing in main memory and are ready and waiting to execute are kept on a list called __________

A. job queue
B. ready queue
C. execution queue
D. process queue

4. In priority scheduling algorithm _____________

A. CPU is allocated to the process with highest priority
B. CPU is allocated to the process with lowest priority
C. Equal priority processes can not be scheduled
D. None of the mentioned

Daily Quiz

5. The interval from the time of submission of a process to the time of completion is termed as ______________

A. waiting time

B. turnaround time

C. response time

D. throughput

6. In the following cases non – preemptive scheduling occurs?

A. When a process switches from the running state to the ready state

B. **When a process goes from the running state to the waiting state**

C. When a process switches from the waiting state to the ready state

D. All of the mentioned

Daily Quiz

9. Scheduling is done so as to ____________

A. increase CPU utilization
B. decrease CPU utilization
C. keep the CPU more idle
D. none of the mentioned

10. A solution to the problem of indefinite blockage of low – priority processes is ____________
A. Starvation
B. Wait queue
C. Ready queue
D. Aging

Weekly Assignment

- Differentiate between preemptive and nonpreemptive scheduling.
- List out the necessary conditions for deadlock to occur..
- Define PCB
- List out the various scheduling criteria for CPU Scheduling.

MCQ

1. The switching of the CPU from one process or thread to another is called ____________

a) process switch

b) task switch

c) context switch

d) all of the mentioned

2. The systems which allow only one process execution at a time, are called ____________

a) uniprogramming systems

b) uniprocessing systems

c) unitasking systems

d) none of the mentioned

MCQ

3. Which of the following is not the state of a process?

A. New
B. Old
C. Waiting
D. Running

4. A single thread of control allows the process to perform ____________

A. only one task at a time
B. multiple tasks at a time
C. only two tasks at a time
D. all of the mentioned

MCQ

5. process can be terminated due to ___________

a) normal exit
b) fatal error
c) killed by another process
d) all of the mentioned

6. What is the ready state of a process?

a) when process is scheduled to run after some execution
b)when process is unable to run until some task has been completed
c) when process is using the CPU
d) none of the mentioned

MCQ

9.This is dynamically allocated memory to a process during its run time.

a)Heap

b). Stack

c) Queue

d) Data

10. How many state are there in Process Life Cycle?

A. 4

B. 5

C. 6

D. 7

Glossary Questions

1. Which one of the following can not be scheduled by the kernel _____________

2. Module gives control of the CPU to the process selected by the short-term scheduler _____________

3. The processes that are residing in main memory and are ready and waiting to execute are kept on a list called: _____________

4. The interval from the time of submission of a process to the time of completion is termed as: _____________
5. Process are classified into different groups in : _____________

(multilevel queue scheduling algorithm, Dispatcher, Ready Queue, Turnaround Time, User level Threads.**)**

Glossary Questions

Choose the correct option:

1. CPU scheduling is the basis of_____________.
2. The two steps of a process execution are_____________.
3. An I/O bound program will typically have _____________.
4. Scheduling is done so as to :_____________.

(I/O Burst, CPU Burst , many very short CPU bursts ,multiprogramming operating system, increase CPU utilization)

Faculty Video Links, Youtube & NPTEL Video Links and Online Courses Details

Youtube/other Video Links

- https://www.youtube.com/watch?v=_zOTMOubT1M
- https://www.youtube.com/playlist?list=PLmXKhU9FNesSFvj6gASuWmQd23Ul5omtD
- https://www.youtube.com/watch?v=x_UpLHXF9dU
- https://www.youtube.com/watch?v=cviEfwtdcEE
- https://nptel.ac.in/courses/106108101

Old Question Papers

B. TECH.

THEORY EXAMINATION (SEM–IV) 2016-17

OPERATING SYSTEM

Time : 3 Hours | *Max. Marks : 100*

Note : Be precise in your answer. In case of numerical problem assume data wherever not provided.

SECTION – A

1. Attempt all of the following questions: **10 x 2 = 20**

(a) Difference between Process and Program.

(b) Explain Context Switching.

(c) What is Demand paging?

(d) Explain Concept of Virtual Memory.

(e) Difference between Directory and File.

(f) Define multiprogramming system.

(g) Difference between External and Internal Fragmentation.

(h) What is Critical Section?

(i) Explain threads.

(j) Define operating system explain in short.

Old Question Papers

SECTION – B

2. Attempt any five of the following questions: 5 x 10 = 50

(a) Write down the different types of operating system

(b) What is Kernel? Describe various operations performed by Kernel.

(c) What is the cause of Thrashing? What steps are taken by the system to eliminate this problem?

(d) What do you understand by Process? Explain various states of process with suitable diagram. Explain process control block.

(e) Give the principles, mutual exclusion in critical section problem. Also discus how well these principles are followed in Dekker's solution.

(f) State the Producer-consumer problem. Given a solution to the solution using semaphores.

(g) Explain File organization and Access mechanism.

(h) Explain the services provided by operating system.

Old Question Papers

SECTION – C

Attempt any two of the following questions: **2 x 15 = 30**

3 (i) What is a deadlock? Discuss the necessary conditions for deadlock with examples

(ii) Describe Banker's algorithm for safe allocation.

4 What do you mean by cashing, spooling and error handling, explain in detail. Explain FCFS, SCAN & CSCAN scheduling with eg.

Expected Questions for University Exam

1. Draw the labeled process state transition diagram with describing the various process states.
2. Write a note on Deadlock Prevention.
3. Explain Multi Level Queue Scheduling.
4. Explain Short term, Middle term and Long term Schedulers.
5. Write down the steps of Deadlock Detection
6. Define PCB.
7. List out the various scheduling criteria for CPU Scheduling.
8. List out the necessary conditions for deadlock to occur.

Expected Questions for University Exam

9. Consider the following set of four processes, with the length of CPU burst time given in milliseconds

Process	Arrival Time	Burst Time
P_1	0	7
P_2	2	4
P_3	4	1
P_4	5	4

Draw Gantt chart and find average waiting time and response time using

- FCFS
- Round Robin (quantum=2)
- SJF and SRTF

Expected Questions for University Exam

10. Let us consider the following snapshot

	Allocation	Max	Available
	A B C	*A B C*	*A B C*
P_0	0 1 0	7 5 3	3 3 2
P_1	2 0 0	3 2 2	
P_2	3 0 2	9 0 2	
P_3	2 1 1	2 2 2	
P_4	0 0 2	4 3 3	

- What is the content of matrix need?
- Is the system in a safe state or not?
- If a request from process P_1 arrives for (1,0,2) can request be granted immediately or not?

Summary

In this module, we have studied the following:

- Process Concept
- Process State
- Process Transition Diagram
- Process Control Block (PCB)
- Thread
  - User thread
  - Kernel thread
- Single and Multithreaded Processes
- Multithreading Models
  - Many-to-One
  - One-to-One
  - Many-to-Many Model
- Types of Schedulers
- Context Switching

Recap of Unit

In this module, we have studied the following:

- Scheduling Concept
- Performance Criteria
- Process Concept
- Process State
- Process Transition Diagram
- Process Control Block (PCB)
- Thread
    - User thread
    - Kernel thread
- Single and Multithreaded Processes
- Multithreading Models
    - Many-to-One
    - One-to-One
    - Many-to-Many Model

Recap of Unit

- Types of Schedulers
- CPU Scheduling Algorithms
  - First Come First Serve(FCFS) Scheduling.
  - Shortest-Job-First(SJF) Scheduling.
  - Priority Scheduling.
  - Round Robin(RR) Scheduling.
  - Multilevel Queue Scheduling.
  - Multilevel Feedback Queue Scheduling.
  - Multiple-Processor Scheduling

References

Books :

1. Silberschatz, Galvin and Gagne, “Operating Systems Concepts”, Wiley
2. SibsankarHalder and Alex A Aravind, “Operating Systems”, Pearson Education
3. Harvey M Dietel, “ An Introduction to Operating System”, Pearson Education
4. D M Dhamdhere, “Operating Systems : A Concept basedApproach”, McGraw Hill.
5. Charles Crowley, “Operating Systems: A Design-Oriented Approach”, Tata McGraw Hill Education”.
6. Stuart E. Madnick & John J. Donovan, “ Operating Systems”, Tata McGraw

Thank You