OPERATING SYSTEM

By – Aditi Mam

June 2014 Paper II

Match the following:

List-I	List-II
a. Multilevel feedback queue	i. Time-slicing
b. FCFS	ii. Criteria to move processes between queues
c. Shortest process next	iii. Batch processing
d. Round Robin Scheduling	iv. Exponential Smoothening

Codes:

	a	b	c	d
(A)	i	iii	ii	iv
(B)	iv	iii	ii	i
(C)	iii	i	iv	i
(D)	ii	iii	iv	i

June 2014 Paper II

Match the following:

List-I	List-II
a. Multilevel feedback queue	i. Time-slicing
b. FCFS	ii. Criteria to move processes between queues
c. Shortest process next	iii. Batch processing
d. Round Robin Scheduling	iv. Exponential Smoothening

Codes:

	a	b	c	d
(A)	i	iii	ii	iv
(B)	iv	iii	ii	i
(C)	iii	i	iv	i
(D)	ii	iii	iv	i

Paper III June 2015

Match the following for operating system techniques with the most appropriate advantage:

List-I	List-II
(a) Spooling	(i) Allows several jobs in memory to improve CPU utilization
(b) Multiprogramming	(ii) Access to shared resources among geographically dispersed computers in a transparent way
(c) Time sharing	(iii) overlapping I/O and computations
(d) Distributed computing	(iv) Allows many users to share a computer simultaneously by switching processor frequently

Codes:

	(a)	(b)	(c)	(d)
(A)	(iii)	(i)	(ii)	(iv)
(B)	(iii)	(i)	(iv)	(ii)
(C)	(iv)	(iii)	(ii)	(i)
(D)	(ii)	(iii)	(iv)	(i)

Paper III June 2015

Match the following for operating system techniques with the most appropriate advantage:

List-I	List-II
(a) Spooling	(i) Allows several jobs in memory to improve CPU utilization
(b) Multiprogramming	(ii) Access to shared resources among geographically dispersed computers in a transparent way
(c) Time sharing	(iii) overlapping I/O and computations
(d) Distributed computing	(iv) Allows many users to share a computer simultaneously by switching processor frequently

Codes:

	(a)	(b)	(c)	(d)
(A)	(iii)	(i)	(ii)	(iv)
(B)	(iii)	(i)	(iv)	(ii)
(C)	(iv)	(iii)	(ii)	(i)
(D)	(ii)	(iii)	(iv)	(i)

Paper III June 2015

Which of the following statements is not true for Multi Level Feedback Queue processor scheduling algorithm?

(A) Queues have different priorities.

(B) Each queue may have different scheduling algorithm

(C) Processes are permanently assigned to a queue

(D) This algorithm can be configured to match a specific system under design

(C) Processes are permanently assigned to a queue

Processes are not permanently assigned to a queue they can very between queues as their priorities change

Paper II July 2016

A scheduling Algorithm assigns priority proportional to the waiting time of a process. Every process starts with priority zero (lowest priority). The scheduler re-evaluates the process priority for every 'T' time units and decides next process to be scheduled. If the process have no I/O operations and all arrive at time zero, then the scheduler implements ..... criteria.

(A) Priority scheduling

(B) Round Robin Scheduling

(C) Shortest Job First

(D) FCFS

(B) Round Robin Scheduling

Because here the quanta for round robin is T units, after a process is scheduled it gets executed for T time units and waiting time becomes least and it again gets chance when every other process has completed T time units.

Paper III July 2016

In UNIX, processes that have finished execution but have not yet had their status collected are known as .....

(A) Sleeping processes **(B) Stopped Processes**

(C) Zombie Processes **(D) Orphan Processes**

(C) Zombie Processes

A process that has finished the execution but still has an entry in the process table to report to its parent process is known as a zombie process.

Paper II January 2017

Match the following w.r.t. Input/Output management:

List-I	List-II
a. Device controller	i. Extracts information from the controller register and store it in buffer
b. Device driver	ii. I/O scheduling
c. Interrupt handler	iii. Performs data transfer
d. Kernel I/O subsystem	iv. Processing of I/O request

Codes:

	a	b	c	d
(1)	iii	iv	i	ii
(2)	ii	i	iv	iii
(3)	iv	i	ii	iii
(4)	i	iii	iv	ii

Paper II January 2017

Match the following w.r.t. Input/Output management:

List-I	List-II
a. Device controller	i. Extracts information from the controller register and store it in data buffer
b. Device driver	ii. I/O scheduling
c. Interrupt handler	iii. Performs data transfer
d. Kernel I/O subsystem	iv. Processing of I/O request

Codes:

	a	b	c	d
(1)	iii	iv	i	ii
(2)	ii	i	iv	iii
(3)	iv	i	ii	iii
(4)	i	iii	iv	ii

Paper II January 2017

Distributed operating systems consist of:

(1) Loosely coupled O.S. software on a loosely coupled hardware

(2) Loosely coupled O.S. software on a tightly coupled hardware

(3) Tightly coupled O.S. software on a loosely coupled hardware

(4) Tightly coupled O.S. software on a tightly coupled hardware

(3) Tightly coupled O.S. software on a loosely coupled hardware

Loosely Coupled	Tightly Coupled
There is distributed memory in loosely coupled multiprocessor system.	There is shared memory, in tightly coupled multiprocessor system.
Loosely Coupled Multiprocessor System has low data rate.	Tightly coupled multiprocessor system has high data rate.
The cost of loosely coupled multiprocessor system is less.	Tightly coupled multiprocessor system is more costly.
In loosely coupled multiprocessor system, modules are connected through **Message transfer system** network.	While there is PMIN, IOPIN and ISIN networks.
In loosely coupled multiprocessor, Memory conflicts don't take place.	While tightly coupled multiprocessor system have memory conflicts.
Loosely Coupled Multiprocessor system has low degree of interaction between tasks.	Tightly Coupled multiprocessor system has high degree of interaction between tasks.
In loosely coupled multiprocessor, there is direct connection between processor and I/O devices.	While in tightly coupled multiprocessor, IOPIN helps connection between processor and I/O devices.
Applications of loosely coupled multiprocessor are in distributed computing systems.	Applications of tightly coupled multiprocessor are in parallel processing systems.

Paper II November 2017

One of the disadvantages of user level threads compared to Kernel level threads is

(1) If a user level thread of a process executes a system call, all threads in that process are blocked.

(2) Scheduling is application dependent.

(3) Thread switching doesn’t require kernel mode privileges.

(4) The library procedures invoked for thread management in user level threads are local procedures.

(1) If a user level thread of a process executes a system call, all threads in that process are blocked.

Parameters	User Level Thread	Kernel Level Thread
Implemented by	User threads are implemented by users.	Kernel threads are implemented by Operating System (OS).
Recognize	The operating System doesn't recognize user-level threads.	Kernel threads are recognized by Operating System.
Implementation	Implementation of User threads is easy.	Implementation of Kernel-Level thread is complicated.
Context switch time	Context switch time is less.	Context switch time is more.
Hardware support	Context switch requires no hardware support.	Hardware support is needed.
Blocking operation	If one user-level thread performs a blocking operation then the entire process will be blocked.	If one kernel thread performs a blocking operation then another thread can continue execution.

Multithreading	Multithread applications cannot take advantage of multiprocessing.	Kernels can be multithreaded.
Creation and Management	User-level threads can be created and managed more quickly.	Kernel-level level threads take more time to create and manage.
Operating System	Any operating system can support user-level threads.	Kernel-level threads are operating system-specific.
Thread Management	The thread library contains the code for thread creation, message passing, thread scheduling, data transfer, and thread destroying	The application code does not contain thread management code. It is merely an API to the kernel mode. The Windows operating system makes use of this feature.

Paper II November 2017

Which module gives control of the CPU to the process selected by the short - term scheduler?

(1) Dispatcher

(2) Interrupt

(3) Scheduler

(4) Threading

(1) Dispatcher

Dispatcher is the model that controls the CPU to the process selected by the short-term scheduler.

Paper III November 2017

User level threads are threads that are visible to the programmer and are unknown to the kernel. The operating system kernel supports and manages kernel level threads. Three different types of models relate user and kernel level threads.

Which of the following statements is/are true?

(i) The Many - to - one model maps many user threads to one kernel thread

(ii) The one - to - one model maps one user thread to one kernel thread

(iii) The many - to - many model maps many user threads to smaller or equal kernel threads

(i) Many - to - one model maps many kernel threads to one user thread

(ii) One - to - one model maps one kernel thread to one user thread

(iii) Many - to - many model maps many kernel threads to smaller or equal user threads

Code :

(1) (a) is true; (b) is false

(2) (a) is false; (b) is true

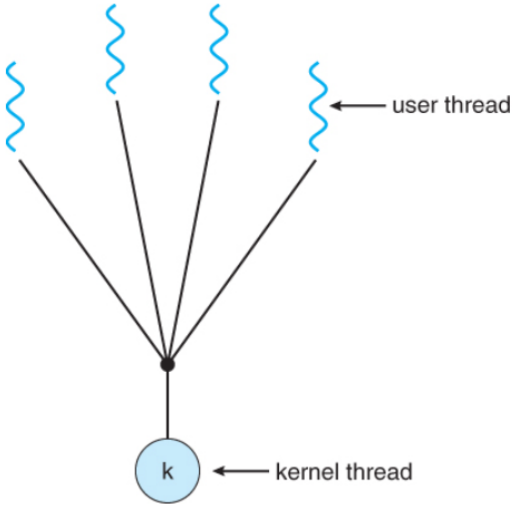
(3) Both (a) and (b) are true

(4) Both (a) and (b) are false

(1) (a) is true; (b) is false

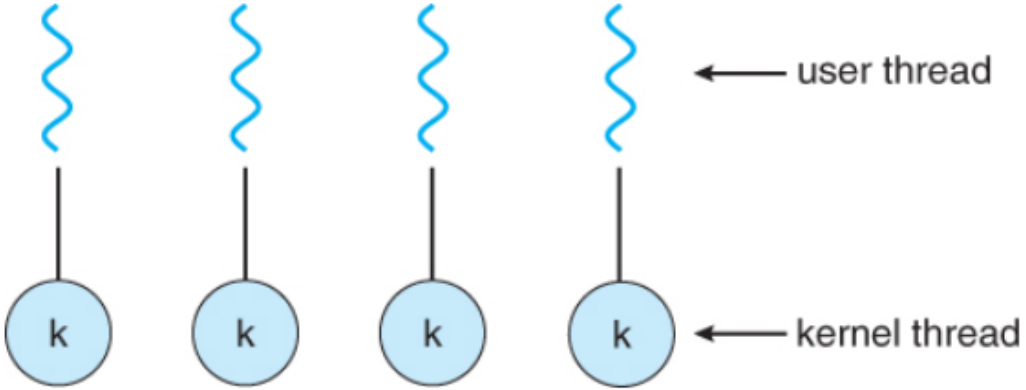
1 Many-To-One Model

- In the many-to-one model, many user-level threads are all mapped onto a single kernel thread.
- Thread management is handled by the thread library in user space, which is very efficient.
- However, if a blocking system call is made, then the entire process blocks, even if the other user threads would otherwise be able to continue.
- Because a single kernel thread can operate only on a single CPU, the many-to-one model does not allow individual processes to be split across multiple CPUs.
- Green threads for Solaris and GNU Portable Threads implement the many-to-one model in the past, but few systems continue to do so today.



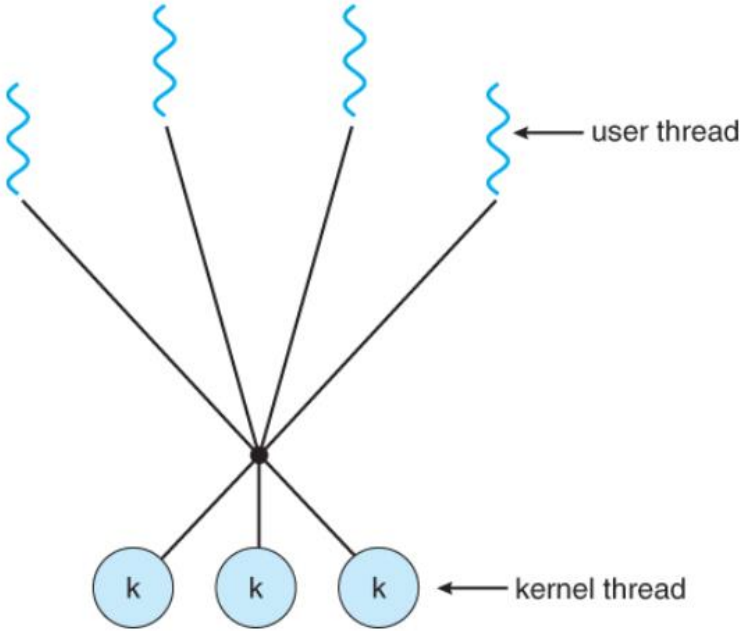
.2 One-To-One Model

- The one-to-one model creates a separate kernel thread to handle each user thread.
- One-to-one model overcomes the problems listed above involving blocking system calls and the splitting of processes across multiple CPUs.
- However the overhead of managing the one-to-one model is more significant, involving more overhead and slowing down the system.
- Most implementations of this model place a limit on how many threads can be created.
- Linux and Windows from 95 to XP implement the one-to-one model for threads.



.3 Many-To-Many Model

- The many-to-many model multiplexes any number of user threads onto an equal or smaller number of kernel threads, combining the best features of the one-to-one and many-to-one models.
- Users have no restrictions on the number of threads created.
- Blocking kernel system calls do not block the entire process.
- Processes can be split across multiple processors.
- Individual processes may be allocated variable numbers of kernel threads, depending on the number of CPUs present and other factors.



Paper II July 2018

In which of the following scheduling criteria, context switching will never take place?

(1) ROUND ROBIN

(2) Preemptive SJF

(3) Non-preemptive SJF

(4) Preemptive priority

Paper II July 2018

In which of the following scheduling criteria, context switching will never take place?

(1) ROUND ROBIN

(2) Preemptive SJF

(3) Non-preemptive SJF

(4) Preemptive priority

Paper II December 2018

A process residing in Main Memory and Ready and Waiting for execution, is kept on

(1) Execution Queue

(2) Job Queue

(3) Ready Queue

(4) Wait Queue

Paper II December 2018

A process residing in Main Memory and Ready and Waiting for execution, is kept on

(1) Execution Queue

(2) Job Queue

(3) Ready Queue

(4) Wait Queue

Paper II December 2018

..... system call creates new process in Unix.

(1) create

(2) create new

(3) fork

(4) fork new

(3) fork

In Unix like operating system, the fork system call is used for creating a new process where a process is a copy of itself, which is called the child process.

Paper II December 2019

Which of the following CPU scheduling algorithms is/are supported by LINUX operating system?

a) Non preemptive priority scheduling
b) Preemptive priority scheduling and time sharing CPU scheduling
c) Time sharing scheduling only
d) Priority scheduling only

b) Preemptive priority scheduling and time sharing CPU scheduling

Preemptive priority scheduling and time sharing CPU scheduling algorithms is/are supported by LINUX operating system.

Paper II December 2019

Identify the circumstances under which pre-emptive CPU scheduling is used:

(a) A process switches from Running state to Ready state

(b) A process switches from Waiting state to Ready state

(c) A process completes its execution

(d) A process switches from Ready to Waiting state

Choose the correct option

a) **(a) and (b) only**

b) **(a) and (d) only**

c) **(c) and (d) only**

d) **(a), (b), (c) only**

a) (a) and (b) only

Preemptive scheduling is used when a process switches from running state to ready state or from the waiting state to ready state.

Paper II December 2019

Which of the following methods are used to pass any number of parameters to the operating system through system calls

a) **Registers**
b) **Block or table in main memory**
c) **Stack**
d) **Block in main memory and stack**

d) Block in main memory and stack

Three general methods used to pass parameters to the OS

1. **Simplest: pass the parameters in registers**
2. **In some cases, may be more parameters than registers. Parameters stored in a block, or table, in memory, and address of block passed as a parameter in a register.**
3. **Parameters placed, or pushed, onto the stack by the program and popped off the stack by the operating system**

Paper II November 2020

Assuming that the system call fork() never fails, consider the following C programs P1 and P2 executed on a UNIX/Linux system:

```
/* P1 */
int main(){
    fork();
    fork();
    fork();
    printf("Happy \n");
}
```

```
/* P2 */
int main(){
    fork();
    printf("Happy \n");
    fork();
    printf("Happy \n");
    fork();
    printf("Happy \n");
}
```

Statement I: P1 displays "Happy" 8 times.

Statement II: P2 displays "Happy" 12 times.

In the light of the above statements, choose the correct answer from the options given below:

a) **Both Statement I and Statement II are true**

b) **Both Statement I and Statement II are false**

c) **Statement I is correct but Statement II is false**

d) **Statement I is incorrect but Statement II is true**

c) Statement I is correct but Statement II is false

First program P1 :- since 3 fork call are there hence 2^3-2 =7 child processes will be created total process 7+1=8, so total 8 times Happy will be printed

Second program P2 :- First fork () call creates total 2 processes [happy printed 2 times]

Second fork() will cause happy prints 4 times (2x2=4)

Third fork() will cause happy printed 8 times (2x4=8)

so total 14 times Happy will be printed

Paper II November 2020

Which of the following statements with respect to multiprocessor system are true:

A. Multiprocessor system is controlled by one operating system.

B. In Multiprocessor system, multiple computers are connected by the means of communication lines.

C. Multiprocessor system in classified as multiple instruction stream and multiple data stream system.

Choose the correct answer from the options given below:

a) (A) only

b) (A) and (B) only

c) (A) and (C) only

d) (B) and (C) only

c) (A) and (C) only

A multiprocessor is a computer system having two or more processing units i.e multiple processors each sharing main memory and peripherals, in order to simultaneously process programs.

- **Multiprocessor systems are controlled by one or more operating systems. Hence Statement I is correct.**
- **In a Multiprocessor system, multiple processors are connected under single computer systems. Hence Statement II is incorrect.**
- **Multiprocessor system is classified as multiple instruction streams and multiple data streams systems. Hence Statement III is correct.**

Paper II November 2021

Match List I with List II:

List I	List II
System calls	Description
A. fork()	I. Sends a signal from one process to another process
B. exec()	II. Indicates termination of the current process
C. kill()	III. Loads the specified program in the memory
D. exit()	IV. Creates a child process

A-II, B-III, C-IV, D-I

A-IV, B-I,C-II, D-III

A-IV, B-III, C-I, D-II

A-IV, B-III, C-II, D-I

c) A-IV, B-III, C-I, D-II

Paper II October 2022

If an operating system does not allow a child process to exist when the parent process has been terminated, this phenomenon is called as

a) Threading
b) Cascading termination
c) Zombie termination
d) Process killing

b) Cascading termination

Cascading termination refers to termination of all child processes if the parent process terminates Normally or Abnormally. Some systems don't allow child processes to exist if the parent process has terminated. Cascading termination is normally initiated by the operating system.

Paper II March 2023

In Linux Operating System, when is invoked, it is passed a set of flags that determine how much sharing is to take place between the parent and child tasks.

1. fork()
2. clone()
3. pthread()
4. thread()

2. clone()

The main use of clone() is to implement threads: multiple threads of control in a program that run concurrently in a shared memory space.

Paper II March 2023

The main function of the microkernel is to provide a communication facility between the _____ program and the various _____ that are also running in user space.

1. Virtual, Processes
2. System, Processes
3. Client Services
4. Virtual, Services

3. Client Services

The main function of the microkernel is to provide a communication facility between the client program and the various services that are also running in user space. Communication is provided by message passing.

Paper II March 2023
Select the correct order of events after power is initialized on a system.

A. Bootstrap loader is loaded from the disk
B. Kernel is loaded onto the memory
C. Firmware ROM loads boot block

Choose the correct answer from the options given below

1. B, C, A
2. C, A, B
3. A, B, C
4. A, C, B

2. C, A, B

The correct order of events after power is initialized on a system is as follows:

C. Firmware ROM loads boot block
A. Bootstrap loader is loaded from the disk
B. Kernel is loaded onto the memory

Process Management

By Aditi Mam

June 2014 Paper III

Consider a uniprocessor system where new processes arrive at an average of five processes per minute and each process needs an average of 6 seconds of service time. What will be the CPU utilization?

(A) 80 %
(B) 50 %
(C) 60 %
(D) 30 %

(B) 50 %

Given in system there are on an average 5 new processes arrive per minute.

So,Arrival rate = 5 process / min

In every 12 second a ($\frac{60}{5}$)new process arrives in system or we can say every process stays 12 second with the CPU.

Given service time = 6 seconds

CPU Utilization = $\frac{Service\ Time}{Staying\ Time} * 100$

$= \frac{6}{12} * 100 = 50\%$

Paper II December 2014

Consider the following justifications for commonly using the two-level CPU scheduling :

I. It is used when memory is too small to hold all the ready processes.
II. Because its performance is same as that of the FIFO.
III. Because it facilitates putting some set of processes into memory and a choice is made from that.
IV. Because it does not allow to adjust the set of in-core processes.

Which of the following is true ?

(A) I, III and IV
(B) I and II
(C) III and IV
(D) I and III

(D) I and III

he two-level CPU scheduling is used when memory is too small to hold all the ready processes because it facilitates putting some set of processes into memory and a choice is made from that.

Paper II December 2014

Which of the following conditions does not hold good for a solution to a critical section problem ?

(A) No assumptions may be made about speeds or the number of CPUs.
(B) No two processes may be simultaneously inside their critical sections.
(C) Processes running outside its critical section may block other processes.
(D) Processes do not wait forever to enter its critical section.

(C) Processes running outside its critical section may block other processes.

If a process is running outside the critical section we don't care whatever it is doing out there either it is blocking some processes or not it is not going to effect the processes which are running inside the critical section anyway.

Paper III December 2014

Monitor is an Interprocess Communication (IPC) technique which can be described as

(A) It is higher level synchronization primitive and is a collection of procedures, variables, and data structures grouped together in a special package.
(B) It is a non-negative integer which apart from initialization can be acted upon by wait and signal operations.
(C) It uses two primitives, send and receive which are system calls rather than language constructs.
(D) It consists of the IPC primitives implemented as system calls to block the process when they are not allowed to enter critical region to save CPU time.

(A) It is higher level synchronization primitive and is a collection of procedures, variables, and data structures grouped together in a special package.

monitor is a thread-safe class, object, or module that uses wrapped mutual exclusion in order to safely allow access to a method or variable by more than one thread. The defining characteristic of a monitor is that its methods are executed with mutual exclusion: At each point in time, at most one thread may be executing any of its methods. Using a condition variable(s), it can also provide the ability for threads to wait on a certain condition (thus using the above definition of a "monitor"). For the rest of this article, this sense of "monitor" will be referred to as a "thread-safe object/class/module".

Paper III June 2015

Dining Philosopher's problem is a:

(A) Producer-Consumer problem
(B) Classical IPC problem
(C) Starvation problem
(D) Synchronization primitive

(B) Classical IPC problem

The simple description of the problem:

- Five silent philosophers sit at a round table with bowls of spaghetti.Forks are placed between each pair of adjacent philosophers.
- Each philosopher must alternately think and eat.
- However, a philosopher can only eat spaghetti when he has both left and right forks.
- Each fork can be held by only one philosopher and so a philosopher can use the fork only if it is not being used by another philosopher.
- After he finishes eating, he needs to put down both forks so they become available to others.
- A philosopher can take the fork on his right or the one on his left as they become available, but cannot start eating before getting both of them.
- Eating is not limited by the remaining amounts of spaghetti or stomach space; an infinite supply and an infinite demand are assumed.
- The problem is how to design a discipline of behavior (a concurrent algorithm) such that no philosopher will starve; i.e., each can forever continue to alternate between eating and thinking, assuming that no philosopher can know when others may want to eat or think.

Paper II August 2016 (Re-test)

Five jobs A, B, C, D and E are waiting in Ready Queue. Their expected runtimes are 9, 6, 3, 5 and x respectively. All jobs entered in Ready queue at time zero. They must run in ..... order to minimize average response time if 3 < x < 5.

(A) B, A, D, E, C
(B) C, E, D, B, A
(C) E, D, C, B, A
(D) C, B, A, E, D

(B) C, E, D, B, A

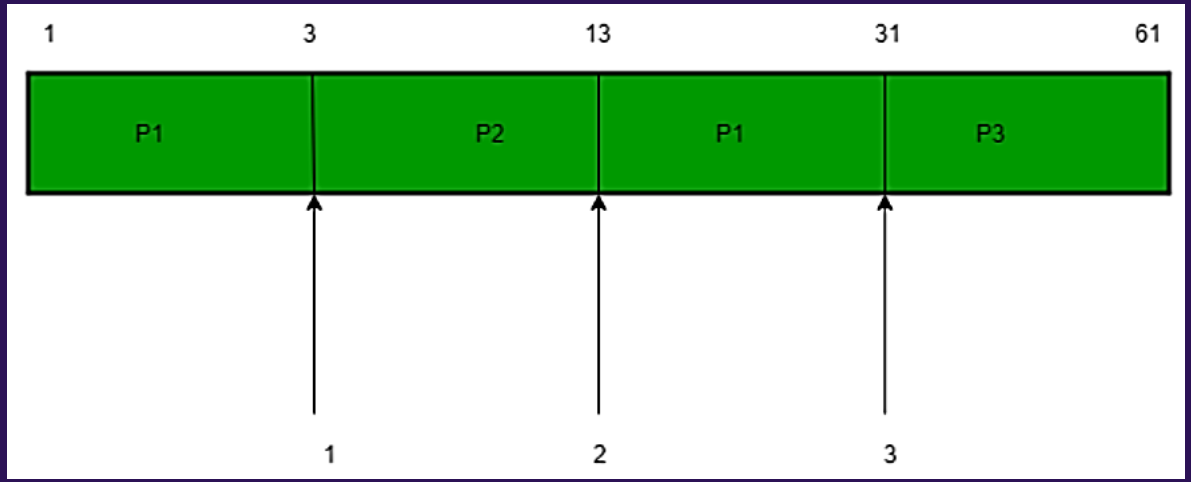
			Option A		Option B		Option C		Option D	
P	AT	BT	CT_A	WT_A	CT_B	WT_B	CT_C	WT_C	CT_D	WT_D
A	0	9	15	6	27	18	27	18	18	9
B	0	6	6	0	18	12	18	12	9	3
C	0	3	27	24	3	0	12	9	3	0
D	0	5	19	14	12	7	9	4	27	22
E	0	4	23	19	7	3	4	0	22	18
				63/5		40/5		43/5		52/5
				AWT=12.6		AWT=8		AWT=8.6		AWT=10.4

Paper II August 2016 (Re-test)

Consider three CPU intensive processes P1, P2, P3 which require 20, 10 and 30 units of time, arrive at times 1, 3 and 7 respectively. Suppose operating system is implementing Shortest Remaining Time first (pre-emptive scheduling) algorithm, then ..... context switches are required (suppose context switch at the beginning of Ready queue and at the end of Ready queue are not counted).

(A) 3
(B) 2
(C) 4
(D) 5

(A) 3



Paper II January 2017

There are three processes P1, P2 and P3 sharing a semaphore for synchronising a variable. Initial value of semaphore is one. Assume that negative value of semaphore tells us how many processes are waiting in queue. Processes access the semaphore in following order:

(a) P2 needs to access
(b) P1 needs to access
(c) P3 needs to access
(d) P2 exits critical section
(e) Pi exits critical section

The final value of semaphore will be:

(1) 0
(2) 1
(3) -1
(4) -2

(1) 0

initial value of semaphores S=1

P2 needs to access decreases it to 0 (no one is waiting)
P1 needs to access decreases it to -1 (one process is waiting)
P3 needs to access decreases it to -2 (2 process r waiting)
P2 exits critical section increases it to -1 (one process is waiting)
Pi exits critical section increases it to 0 (no process is waiting)

Paper II January 2017

Which of the following scheduling algorithms may cause starvation?

a. First-come-first-served
b. Round Robin
c. Priority
d. Shortest process next
e. Shortest remaining time first

(1) a, c and e
(2) c, d and e
(3) b, d and e
(4) b, c and d

(2) c, d and e

First-come-first-served- No starvation
Round Robin- No starvation
Priority- starvation if higher priority process called again and again, lower priority starves
Shortest process next -starvation possible
Shortest remaining time first -starvation possible as its a preemptive version of shortest process next casuing apotential for longer processes to starve.

Paper III January 2017

Some of the criteria for calculation of priority of a process are:

a. Processor utilization by an individual process.
b. Weight assigned to a user or group of users.
c. Processor utilization by a user or group of processes.

In fair share scheduler, priority is calculated based on:

(1) only (a) and (b)
(2) only (a) and (c)
(3) (a), (b) and (c)
(4) only (b) and (c)

(3) (a), (b) and (c)

As the name of the scheduler tells about itself Fair-share means every process has an equal share of the CPU.

Fair-share scheduling is a scheduling algorithm for computer operating systems in which the CPU usage is equally distributed among system users or groups, as opposed to equal distribution among processes.

Paper II November 2017

Two atomic operations permissible on Semaphores are ..... and .....

(1) wait, stop
(2) wait, hold
(3) hold, signal
(4) wait, signal

(4) wait, signal

Semaphores are integer variables that are used to solve the critical section problem by using two atomic operations, wait and signal that are used for process synchronization.

Paper III November 2017

The Bounded buffer problem is also known as .....

(1) Producer - consumer problem
(2) Reader - writer problem
(3) Dining Philosophers problem
(4) Both (2) and (3)

(1) Producer - consumer problem

- The Bounded buffer problem is also known as the producer-consumer problem. The Producer-Consumer problem is a classic multi-process synchronization problem, which means we're attempting to synchronize several processes.
- In the producer-consumer dilemma, there is only one Producer who produces certain goods, and there is only one Consumer who consumes the Producer's products. Both producers and users share the same memory buffer, which has a fixed capacity.
- The Producer's job is to make the object, store it in the memory buffer, and then start making more. The Consumer's responsibility is to consume the object from the memory buffer.

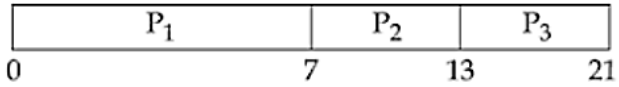
Paper II July 2018

Consider the following three processes with the arrival time and CPU burst time given in milliseconds:

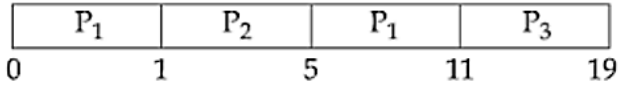
Process	Arrival Time	Burst Time
P_1	0	7
P_2	1	4
P_3	2	8

The Gantt Chart for preemptive SJF scheduling algorithm is .....

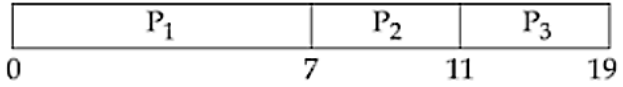
(1)



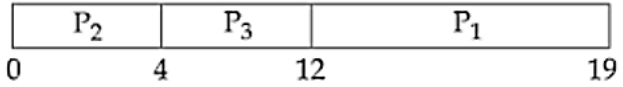
(2)



(3)

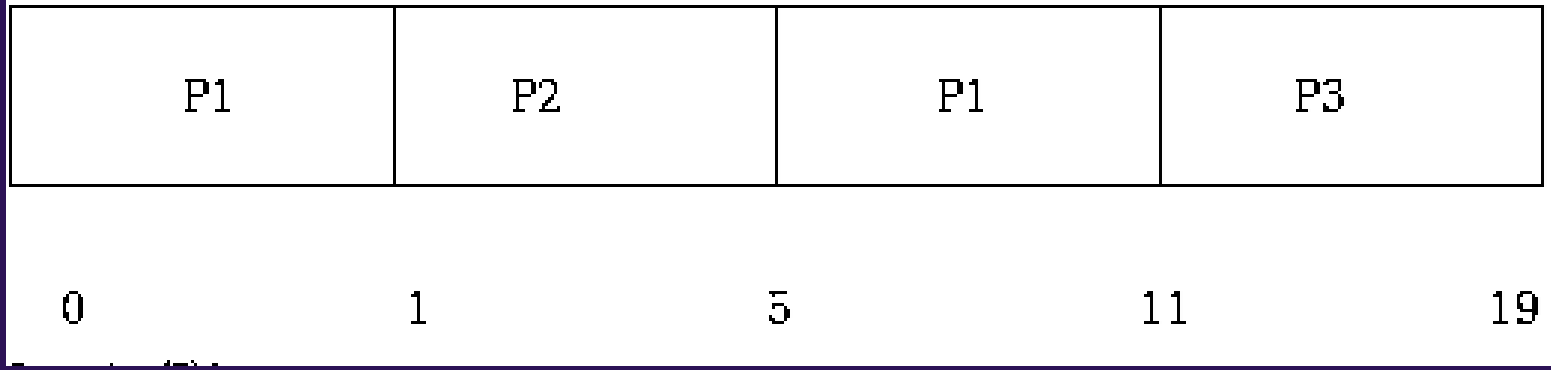


(4)



Answer: 2

Gantt chart for preemptive SJF scheduling algorithm is:



Paper II December 2018

To overcome difficulties in Readers-Writers problem, which of the following statement/s is/are true?

(i) Writers are given exclusive access to shared objects
(ii) Readers are given exclusive access to shared objects
(iii) Both Readers and Writers are given exclusive access to shared objects

Choose the correct answer from the code given below:

Code:
(1) (i) only
(2) (ii) only
(3) (iii) only
(4) Both (ii) and (iii)

(1) (i) only

In Readers-Writers problem, more than one Reader is allowed to read simultaneously but if a Writer is writing then no other writer or any reader can have simultaneous access to that shared object. So, Writers are given exclusive access to shared objects.

Paper II December 2018

Consider the following set of processes and the length of CPU burst time given in milliseconds:

Process	CPU Burst time (ms)
P_1	5
P_2	7
P_3	6
P_4	4

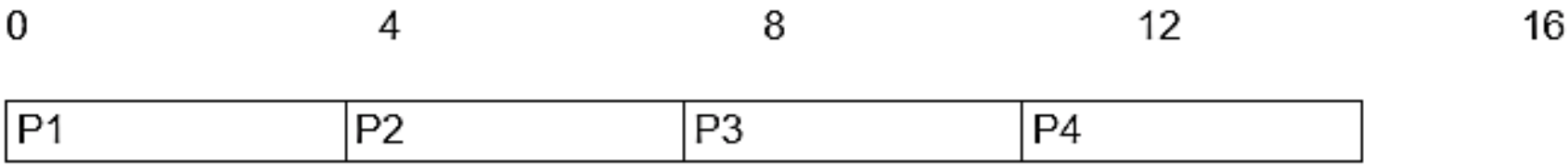
Assume that processes being scheduled with Round-Robin Scheduling Algorithm with time quantum 4 ms. The the waiting time for P4 is ..... ms.

(1) 0
(2) 4
(3) 12
(4) 6

(3) 12

Time quantum is 4, assuming every process arrived at $t = 0$

When every process uses 1 time quantum,



P_4 finished at $t = 16$

$$waiting\,time = 16 - 4 = 12\,ms$$

Paper II June 2019

At a particular time of computation, the value of a counting semaphore is 7. Then 20 P (wait) operations and 15V (signal) operations are completed on this semaphore. What is the resulting value of the semaphore?

(a) 28
(b) 12
(c) 2
(d) 42

(c) 2

since P(S) decreases the counting semaphore value
The initial value of semaphore is 7
After finishing 20 P operations S=-13 (i.e 7-20 =-13)
Now S=-13
V(S) increases the counting semaphore value
After finishing 15 V operations S=-13+15=2
Therefore Answer is 2

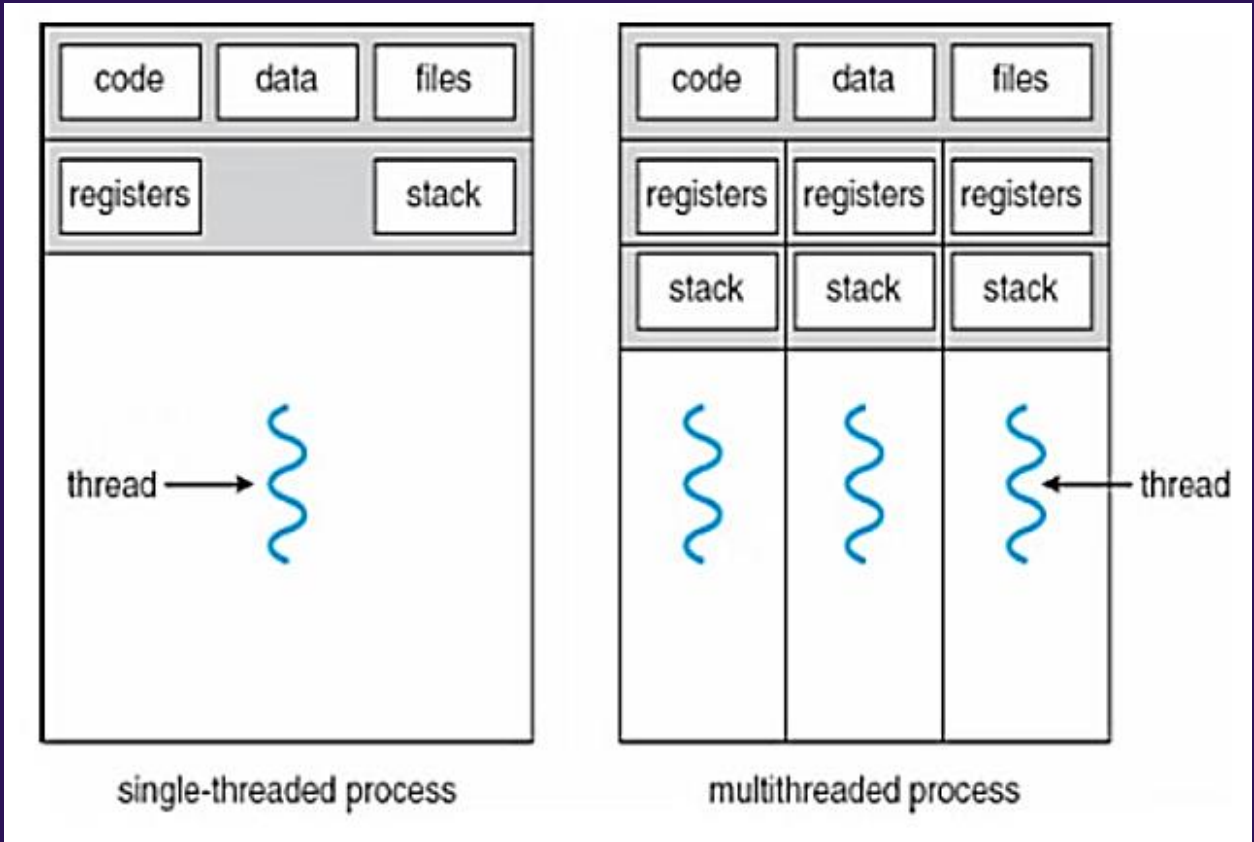
Paper II June 2019

Which of the following are NOT shared by the threads of the same process?

(1) Stack
(2) Registers
(3) Address space
(4) Message queue

(a) (1) and (4)
(b) (2) and (3)
(c) (1) and (2)
(d) (1), (2) and (3)

(c) (1) and (2)


code
data
files
registers
stack
thread
single-threaded process
code
data
files
registers
registers
registers
stack
stack
stack
thread
multithreaded process

Paper II December 2019

A counting semaphore is initialized to 8. 3 wait () operations and 4 signal () operations are applied Find the current value of semaphore variable.

a) 9
b) 5
c) 1
d) 4

a) 9

wait() operation decrements the value of a semaphore.
signal() operation increments the value of a semaphore.

3 wait() implies -3 to value of semaphore.
4 signal() implies +4 to value of semaphore.

So 8 – 3 + 4 = 9

Paper II December 2019

Which of the following interprocess communication model is used to exchange messages among co-operative process?

a) Shared memory model
b) Message passing model
c) Shared memory and message passing model
d) Queues

c) Shared memory and message passing model

Co operative processes use shared memory method and message passing methods to exchange messages.

Paper II December 2019

Given CPU time slice of 2ms and following list of processes.

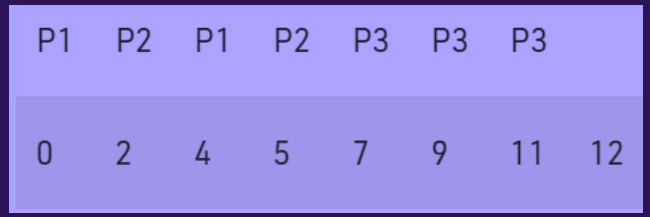
Process	Burst time (ms)	Arrival time
p_1	3	0
p_2	4	2
p_3	5	5

Find average turnaround time and average waiting time using round robin CPU scheduling?

a) 4, 0
b) 5.66, 1.66
c) 5.66, 0
d) 7, 2

b) 5.66, 1.66

Gantt Chart for given processes: P1,P2,P3



Process	Burst time (ms)	Arrival time	Completion time	Turn around time	Waiting time
P1	3	0	5	5	2
P2	4	2	7	5	1
P3	5	5	12	7	2

Now Average Turnaround Time = (5 + 5 + 7) / 3 = 5.66
Average Waiting Time = (2 + 1 + 2) / 3 = 1.66

Paper II November 2021

Consider the following 3 processes with the length of the CPU burst time given in milliseconds:

Process	Arrival Time	Burst Time
P1	0	8
P2	1	4
P3	2	9

What is the average waiting time for these processes if they are scheduled using preemptive shortest job first scheduling algorithm?

a) 2.66
b) 4.66
c) 5.5
d) 6

b) 4.66

Gantt Chart

0 1 5 12 21

Process	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P1	0	8	12	12	4
P2	1	4	5	4	0
P3	2	9	21	19	10

Average Waiting Time = (4+0+10)/3

= 14/3

= 4.66

Paper II October 2022
Match List I with List II:

	List I		List II
(A)	Firmware	(I)	Number of logical records into physical blocks
(B)	Batch file	(II)	ASCII format
(C)	Packing	(III)	Resource allocation
(D)	Banker's Algorithm	(IV)	ROM

Choose the correct answer from the options given below:

1. (A)-(II), (B)-(I), (C)-(IV), (D)-(III)
2. (A)-(II), (B)-(I), (C)-(III), (D)-(IV)
3. (A)-(IV), (B)-(II), (C)-(I), (D)-(III)
4. (A)-(IV), (B)-(I), (C)-(II), (D) (III)

Paper II October 2022
Match List I with List II:

	List I		List II
(A)	Firmware	(I)	Number of logical records into physical blocks
(B)	Batch file	(II)	ASCII format
(C)	Packing	(III)	Resource allocation
(D)	Banker's Algorithm	(IV)	ROM

Choose the correct answer from the options given below:

1. (A)-(II), (B)-(I), (C)-(IV), (D)-(III)
2. (A)-(II), (B)-(I), (C)-(III), (D)-(IV)
3. (A)-(IV), (B)-(II), (C)-(I), (D)-(III)
4. (A)-(IV), (B)-(I), (C)-(II), (D) (III)

Paper II October 2022
Match List I with List II:

	List I		List II
(A)	Least frequently used	(I)	Memory is distributed among processors
(B)	Critical Section	(II)	Page replacement policy in cache memory
(C)	Loosely coupled multiprocessor system	(III)	Program section that once begin must complete execution before another processor access the same shared resource
(D)	Distributed operating system organization	(IV)	O/S routines are distributed among available processors.

Choose the correct answer from the options given below

1. (A)-(III), (B)-(II), (C)-(IV), (D)-(I)
2. (A)-(I), (B)-(II), (C)-(III), (D)-(IV)
3. (A)-(II), (B)-(III), (C)-(I), (D)-(IV)
4. (A)-(II), (B)-(I), (C)-(III), (D)-(IV)

Paper II October 2022
Match List I with List II:

	List I		List II
(A)	Least frequently used	(I)	Memory is distributed among processors
(B)	Critical Section	(II)	Page replacement policy in cache memory
(C)	Loosely coupled multiprocessor system	(III)	Program section that once begin must complete execution before another processor access the same shared resource
(D)	Distributed operating system organization	(IV)	O/S routines are distributed among available processors.

Choose the correct answer from the options given below

1. (A)-(III), (B)-(II), (C)-(IV), (D)-(I)
2. (A)-(I), (B)-(II), (C)-(III), (D)-(IV)
3. (A)-(II), (B)-(III), (C)-(I), (D)-(IV)
4. (A)-(II), (B)-(I), (C)-(III), (D)-(IV)

Paper II March 2023

Consider an operating system capable of loading and executing a single sequential user process at a time. The disk head scheduling algorithm used is first come first served (FCFS). If FCFS is replaced by shortest seek time first (SSTF) and the vendor claims 50% better benchmark results. What is the expected improvement in the I/O performance of user programs?

1. 50%
2. 100%
3. 25%
4. 0%

4. 0%

The I/O performance of a user program determined by many input and output devices not only by the disk. When we uses FCFS and replace it by SSTF it improves only disk driver performance not the entire I/O performances. So I/O improvement performance of user program is 0%.

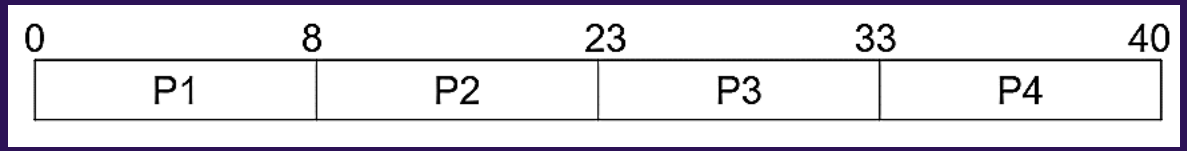
Paper II March 2023

For the following set of processes scheduled using FCFS policy, determine the average waiting time Assume that the processes arrived in the order P1, P2, P3, P4.

Process	Burst Time (ms)
P1	8
P2	15
P3	10
P4	7

1. 8
2. 16
3. 32
4. 48

2. 16



Process	BT	TAT = CT – AT	WT = TAT – BT
P1	8	8	0
P2	15	23	8
P3	10	33	23
P4	7	40	33

Waiting time = (8+23+33)/4 = 64/4 =16

Paper II March 2023

Which of the following statements are true?

A. Shortest remaining time first scheduling may cause starvation
B. Preemptive scheduling may cause starvation
C. Round robin is better than FCFS in terms of response time

Choose the correct answer from the options given below

1. A only
2. B, C only
3. A, B only
4. A, B, C

4. A, B, C

Shortest remaining time first scheduling may cause starvation
Preemptive scheduling may cause starvation
Round robin is better than FCFS in terms of response time
All statements are true.

Paper II March 2023

An OS follows round-robin scheduling with time quantum of 4ms. Assuming that the CPU is free now and there are 20 processes waiting in the ready queue the, maximum amount of time that a process waits before getting into the CPU is

1. **80ms**
2. **76ms**
3. **84ms**
4. **None of the above**

2. 76ms

Given that the time quantum is 4ms and there are 20 processes waiting in the ready queue, we can calculate the maximum wait time as follows:

Maximum wait time = (Number of processes in the ready queue - 1) * Time quantum
Maximum wait time = (20 - 1) * 4
Maximum wait time = 19 * 4
Maximum wait time = 76ms

Paper II March 2023

Which of the following statement is/are false?

A. The processor has direct access to both primary and secondary memory
B. Primary memory stores the active instructions and data for the program being executed on the process
C. Secondary memory is used as a backup memory
D. Memory system is implemented on a single level memory

Choose the correct answer from the options given below:

1. A and B only
2. B and C only
3. A, B and C only
4. A and D Only

4. A and D Only

Statement A: "The processor has direct access to both primary and secondary memory." This statement is false. The processor (CPU) has direct access to primary memory (RAM), but it does not have direct access to secondary memory.

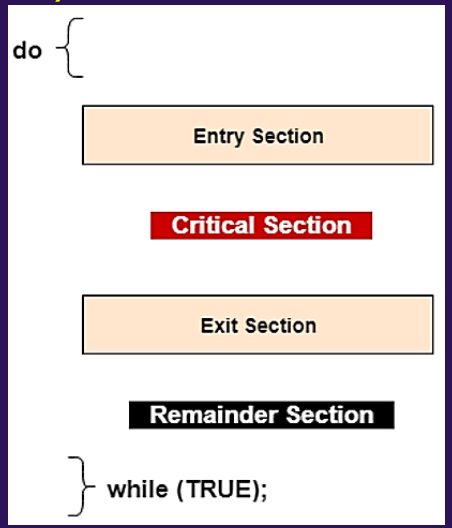
Statement D: "Memory system is implemented on a single level memory." This statement is false. The memory system is implemented as a hierarchy of multiple levels, including primary memory (RAM) and secondary memory (e.g., hard disk). The memory hierarchy is designed to optimize the trade-off between speed, cost, and capacity.

Paper II March 2023

In design protocol of critical section problem, each process must ask permission to enter critical section in ____ code; it then executes in the critical section; once it finishes executes in the critical section it enters the ____ code. The process then enters the ____ code.

1. **entry section, remainder section, exit section**
2. **entry section, exit section, remainder section**
3. **remainder section, entry section, exit section**
4. **remainder section, exit section, entry section**

2. entry section, exit section, remainder section



In the above diagram, the entry section handles the entry into the critical section. It acquires the resources needed for execution by the process. The exit section handles the exit from the critical section. It releases the resources and also informs the other processes that the critical section is free.

Paper II JUNE 2023

Consider the following table of arrival time and burst time for three processes P0, P1, P2:

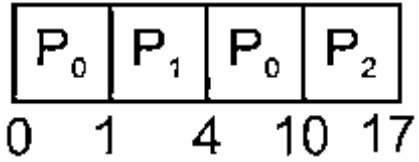
Process	arrival time	Burst time
P0	0 ms	7
P1	1 ms	3
P2	2 ms	7

The pre-emptive shortest job first scheduling algorithm is used. Scheduling is carried out only at arrival or completion of a process. What is the average waiting time for the three processes?

3 ms
3.67 ms
4.47 ms
4 ms

b) 3.67 ms

Gantt chart



Process	arrival time	Burst time	Waiting time
P0	0 ms	7	3
P1	1 ms	3	0
P2	2 ms	7	8

So, the average waiting time is:

(3ms + 0ms + 8ms) / 3 = 3.67ms (rounded to nearest hundredth)

Paper II JUNE 2023

Given below are two statements: one is labelled as Assertion A and the other is labelled as Reason R.

Assertion A: A process involves a library function to create a thread.
Reason R: The threads make system calls to convey their resource and I/O requirement to the Kernel.

In the light of the above statements, choose the correct answer from the options given below.

1. Both A and R are true and R is the correct explanation of A
2. Both A and R are true but R is NOT the correct explanation of A
3. A is true but R is false
4. A is false but R is true

b) Both A and R are true but R is NOT the correct explanation of A

Assertion A is true because a process can indeed use a library function to create a new thread. Libraries like the P threads library in C and Java's built-in multi-threading functionality offer such methods.

Reason R is also true since threads, like processes, can make system calls to request services from the operating system's kernel, which include tasks related to resource allocation and I/O operations.

Paper II December 2023

Arrange the following levels of interrupt protection within the Linux Kernel, in the order of increasing priority.

(A) user mode programs
(B) bottom half interrupt handlers
(C) kernel system service routines
(D) top half interrupt handlers

Choose the correct answer from the options given below :

(1) (A), (B), (D), (C)
(2) (A), (C), (B), (D)
(3) (A), (C), (D), (B)
(4) (D), (A), (C), (B)

(2) (A), (C), (B), (D)

(A) User Mode Programs: These programs are applications that are running on your computer. They do not interact directly with the system hardware or with other programs and have the least system privilege level. They request system resources and services from the kernel.

(C) Kernel System Service Routines: These routines facilitate the user mode programs by providing system services. They operate at a higher privilege level than user programs since they interact directly with system structures and hardware, but they are not as prioritized as interrupt handlers.

(B) Bottom Half Interrupt Handlers: These handlers deal with the processing of interrupts, but not immediately. They handle the less critical part of the interrupt handling, often the work that can be deferred to a later point when the system is less busy.

(D) Top Half Interrupt Handlers: These handlers are responsible for dealing with interrupts immediately after they occur to ensure uninterrupted functioning of the system. Therefore, they have the highest priority. After attending to the immediate needs of an interrupt, they may schedule the rest of the work to the bottom half handlers.

Paper II December 2023

Given below are two statements:

Statement (I): A thread is a dispatchable unit of work that does not executes sequentially and is not interruptible

Statement (II): It is not possible to alter the behaviour of a thread by altering its context when thread is suspended

In the light of the above statements, choose the most appropriate answer from the options given below:

(1) Both Statement I and Statement II are correct
(2) Both Statement I and Statement II are incorrect
(3) Statement I is correct but Statement II is incorrect
(4) Statement I is incorrect but Statement II is correct

(2) Both Statement I and Statement II are incorrect

Statement (I): "A thread is a dispatchable unit of work that does not execute sequentially and is not interruptible." This statement is incorrect. A thread is a basic unit of execution within a process, and it does execute sequentially. Threads can be interrupted, and their execution can be preempted by the operating system scheduler, allowing other threads to run.

Statement (II): "It is not possible to alter the behavior of a thread by altering its context when the thread is suspended." This statement is also incorrect. The context of a thread includes its register values, program counter, and other execution-related information. When a thread is suspended, its context can be saved, modified, and then restored when the thread resumes execution. This allows for changes in the behavior of the thread.