Q1 a)

ans 1(i) An operating system is a construct that allows the user application programs to interact with the system hardware. Operating system itself does not provide any function but it provides an atmosphere in which different applications and programs can do useful work.

The operating system can be observed from the point of view of the user or the system. This is known as the user view and the system view respectively.

USER VIEW:
- → Depends on system interface by user
- → Different view experiences —

- If user is using a personal computer the operating system is largely designed to make interaction easy personal computer uses all the resources available and no sharing.
- OS is largely concerned to resource utilization.
- If user is on workstation then OS need to focus on both individual usage and sharing of resource.

SYSTEM VIEW:
- → Bridge between applications and hardware
- → It is most intimate with hardware used to control.
- → Different type of system view for OS.

- System view OS as resource allocator.
  - as a control program
  - as a way for easier hardware
- To easy communication
- Considered to run all time in background.

Types of operating System.

1) Batch operating System → Does not interact with computer directly. There is an operator, which takes similar jobs having the same requirement and group them into batches.

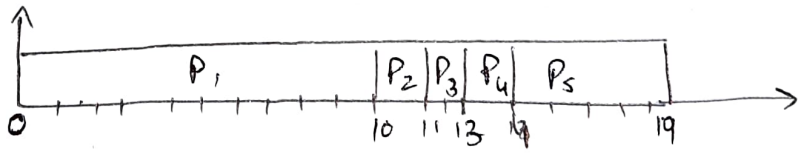
2) Time sharing operating → Time given for execution, then OS switches over to next task. These are also known Multitasking system.

3) Multiprogramming System: Various autonomous - inter-connected computers communicate using a shared communication network.

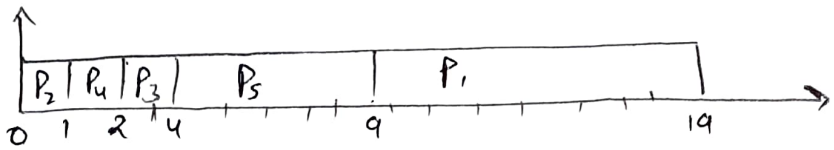
4) Multiprocessing System: These system run on a server and provide the capability to manage data, users, groups, security, application and other network function.

Q-1 (i)

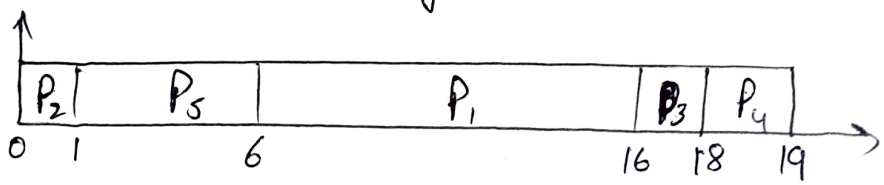
Ans 1 (ii) • a) FCFS - Gantt chart



SJF - Gantt chart.



non-Premetive priority - Gantt chart.



b) Turn Around Time (TAT)

FJFS

Process	TAT
P_1	10 − 0 = 10
P_2	11 − 0 = 11
P_3	13 − 0 = 13
P_4	14 − 0 = 14
P_5	19 − 0 = 19

Total TAT = 67

SJF

Process	TAT
P_1	19 − 0 = 19
P_2	1 − 0 = 1
P_3	4 − 0 = 4
P_4	2 − 0 = 2
P_5	9 − 0 = 9

Total TAT = 35

Non Preemptive Priority

Process	TAT
P_1	16 − 0 = 16
P_2	1 − 0 = 1
P_3	18 − 0 = 18
P_4	19 − 0 = 19
P_5	6 − 0 = 6

Total TAT = 60

c) Waiting Time

Process	FIFS WT	SJF WT	NP Priority WT
P_1	0	9	6
P_2	10	0	0
P_3	11	2	16
P_4	13	1	18
P_5	14	4	1
Total	48	16	41

d) Average TAT = $\frac{\text{Total TAT}}{\text{no. of Process}}$ Average WT = $\frac{\text{Total Wait}}{\text{no of Process}}$

FIFS	13.6	9.6
SJF	7	3.2
NP priority	12	8.2

Q-2(i)

Ans 2(i) (i) Race condition → Potential IPC problem when two or more process interact via shared data. Final outcome depends on exact instruction sequence.

Depends on which process "wins" the race.

(ii) Critical Section → Critical section of program is where global shared memory is being accessed. Process has exclusive access to share modified data while in critical region.

For efficient operation —

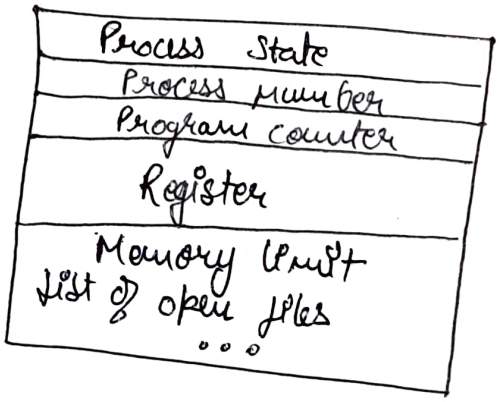
- No two processes can be inside their critical section.
- No assumption made about relative process speed
- No process can be indefinitely postponed from entering its critical section.
- No process stopped outside its critical section can block other process.

(iii) Sleep and Wake up → Essentially, when a process is not permitted to access its critical section, it uses a system call known as sleep, which causes that process to block. The process will not be scheduled to run again, until another process uses to wake up system call. In most cases, wakeup is called by a process when it leaves it critical section if any other process have blocked.

(iv) Sleeping Barber Problem → It has one barber, one barber chair, and n chairs for waiting customers, if any to sit in. If there is no customer present, Barber sit down in chair and falls asleep. Wakeup when customer arrives. Customer leave shop if all chairs are full. The problem is to program the ~~the~~ the barber and the customer without getting into race condition.

Q-2(ii)

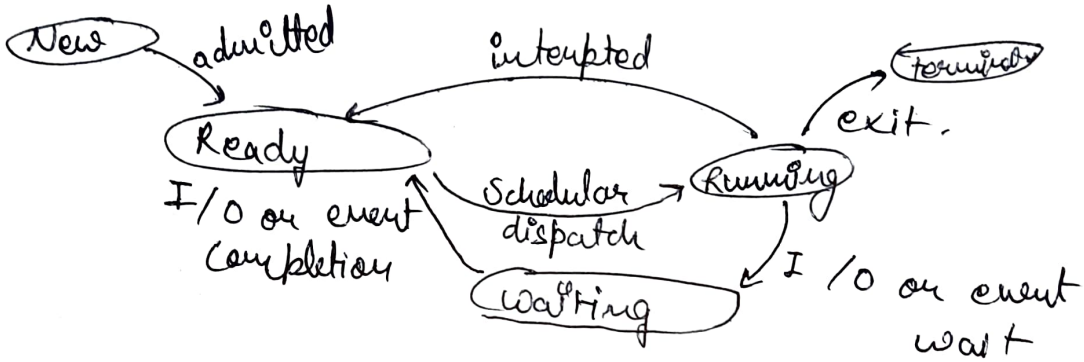
Ans 2(ii) Process life cycle can be defined by a state diagram which has states representing the execution states of process at various time and transition. That shows changes in execution status. To maintain the information about a process, the OS uses process control block or Task control Block.



Process control Block (PCB)
Task control Block.

Process life cycle consists of five stages which are —

- New → The process which is being created
- Running → Instruction being executed.
- Waiting → The process is waiting for an event to occur
- Ready → Waiting to assigned to a process
- Terminated → Process completed its execution.

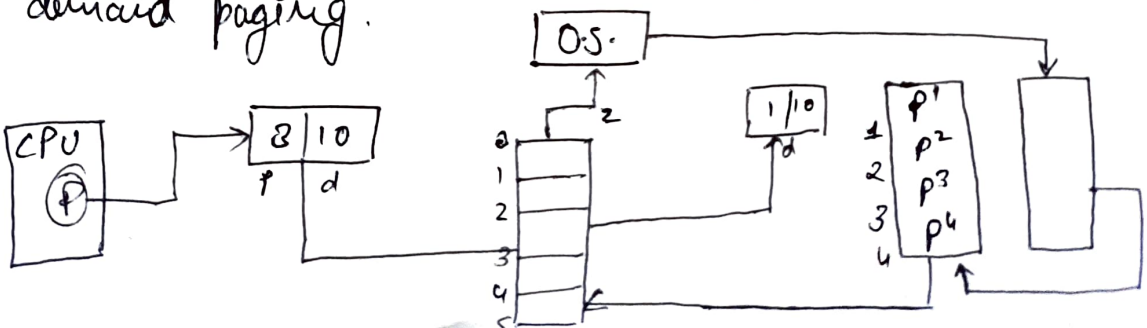


Q-3(i)

Ans 3(i) Virtual memory is a storage mechanism which offers user an illusion of having a very big main memory. It is done by treating a part of secondary memory as the main memory. In virtual memory, the user can store processes with a bigger size than the available main memory.

Various ways for implementing it are:-

a) Demand Paging → The process of loading the page into memory on demand (whenever page fault occurs) is known as demand paging.



→ If the CPU tries to refer to page that is currently not available in the main memory, it generates an interrupt indicating a memory aries fault.

→ OS puts the interrupted process in a blocking state.

b) Swapping:- It is a process out means removing all of its pages from memory or making them so, that they will be removed by the normal page replacement process.

c) Thrashing → At a given time, only a few pages of any process are in the main memory and therefore more processes can be maintained in memory. Pages not swapped in and out of memory.

Q3(ii)

Ans 3(ii)

↳ Segment table

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

i) 0, 430 → Base = 219 ; offset = 430 ; Length = 60

offset < length, ∴ It is valid physical address

Address = 219 + 430 = 649

ii) 1, 10 → Base = 2300 ; offset = 10 ; length = 14

offset < length, ∴ It is valid physical address

Address = 2300 + 10 = 2310

iii) 4, 12 → Base = 1952; offset = 12; length = 96

offset < length ; ∴ It is valid physical address

Address = 1952 + 12 = 1964

iv) 2,500 : Illegal address since size is 100 and offset is 500. ∴ It is INVALID.

Q-41i)

Ans 4(i)

Need Matrix.

	A	B	C	D
P_0	0	0	0	0
P_1	0	7	5	0
P_2	1	0	0	2
P_3	0	0	2	0
P_4	0	6	4	2

i) To check if system is in a safe state, we need to find safety sequence such that it satisfy the criteria needs.

Need ≤ Avail.

① P_0
as need (P_0) ≤ Available, we select P_0 [0000] ≤ [1520]

After Completition [Available] = [1532]

② P_1 Need(P_1) ≥ Available = [0750] > [1532]
*Not fulfilling criteria.

③ P_2 Need(P_2) ≤ Available
[1002] [1532] After Completion [Available] = [2886]

④ P_3 Need(P_3) ≤ Available
[0020] [2886] After [Available] = [2 14 11 8]

⑤ P_4 Need(P_4) ≤ Available
[0642] [2 14 11 8] After [Available] = [2 14 12 8]

(6) Repeat:

Need(P_1) ≤ Available

↳ [3 14 12 12]

Process completed.

Safe Seq = $\langle P_0, P_2, P_3, P_4, P_1 \rangle$

System is in safe state.

ii) A request for Process P_2 arrives for (0,4,2,0)

Need(P_1) > Req(P_1)

[0 7 5 0] > [0,4,2,0] → true.

Available > Req(P_2)

[1 5 2 0] > [0 4 2 0] – true.

Updating values

	Allocation	Max	Need	Available
P_1	1 4 2 0	1 7 5 0	0 3 3 0	1 1 0 0

on verification we can see that still the safe sequence remains the same.

Q-4 (ii)

Ans 4 (ii)

(1) FCFS scheduling algorithm.

- Total disk distance:

(152 − 86) + (147 − 86) + (147 − 913) + (1774 − 913) + (1850 − 1774) + (1850 − 950) + (1525 − 850) + (1728 − 1525) + (1728 − 1000)

= 6423 cylinders.

(2) SSTF scheduling algorithm.

Total distance = 1830 cylinders.

(3) SCAN Algorithm

Total distance → 9760 cylinders.

(4) C-SCAN Algorithm

Total distance → 9973 cylinders

(5) LOOK Algorithm

Total distance → 3462 cylinders

(6) C-LOOK Algorithm

Total distance = 3503 cylinders.

Ans 5 i) FIFO (First in First out) page trace analysis.

Page Request	a	b	a	c	a	b	d	b	a	c	d
Page fault	*	*		*			*		*		
Page frame 1	a	a	a	a	a	a	d	d	d	d	d
Page frame 2		b	b	b	b	b	b	b	a	a	a
Page frame 3				c	c	c	c	c	c	c	c
Stack 1 (Top)	a	b	b	c	c	c	d	d	a	a	a
Stack 2		a	a	b	b	b	c	c	d	d	d
Stack 3 (oldest)				a	a	a	b	b	c	c	c
Stack 4 (swapped out)							a	a	b	b	b

5 page faults in 11 steps yields:

Success rate of $6/11 \Rightarrow \frac{6}{11} \times 100 = 55\%$.

Failure rate of $5/11 \Rightarrow \frac{5}{11} \times 100 = 45\%$.

(ii) LRU (Least Recently used) page trace analysis.

Page Request	a	b	a	c	a	b	d	b	a	c	d
Page Fault	*	*		*			*			*	*
Page Frame 1	a	a	a	a	a	a	a	a	a	a	a
Page Frame 2		b	b	b	b	b	b	b	b	b	d
Page Frame 3				c	c	c	d	d	d	c	c
Stack (1) (Top)	a	b	a	c	a	b	d	b	a	c	d
Stack 2		a	b	a	c	a	d	b	b	a	c
Stack 3 (old)				b	b	c	a	a	d	b	a
Stack 4 (Swapped out)							c	c	c	d	b

6 faults in 11 steps yields:

Success rate $= 5/11 \times 100 = 45\%$.

Failure rate $= 6/11 \times 100 = 55\%$.

The FIFO (first in First out) yielded a more favorable success rate than the LRU (least Recently Used) method. but we cannot find the optimal method by taking first just one sample.

iii) Optimal Method for page trace analysis.

Page Request	a	b	a	c	a	b	d	b	a	c	d
Page Fault	*	*		*			*	*			*
Page Frame 1	a	a	a	a	a	a	a	a	a	a	a
Page Frame 2		b	b	b	b	b	d	b	b	b	b
Page Frame 3				c	c	c	c	c	c	c	c
Stack 1 (Top)	a	b	a	c	a	b	d	b	a	c	d
Stack 2		a	b	a	c	a	a	a	b	a	a
Stack 3 (old)				b	b	c	c	c	a	b	b
Stack 4 (Swapped out)							b	d	d	d	c

Q5(ii)
Ans5(ii)

a) Semaphore and Monitor :

Monitor :- High type level synchronisation construct. abstract data type. The monitor type contains shared variable and set of procedures.

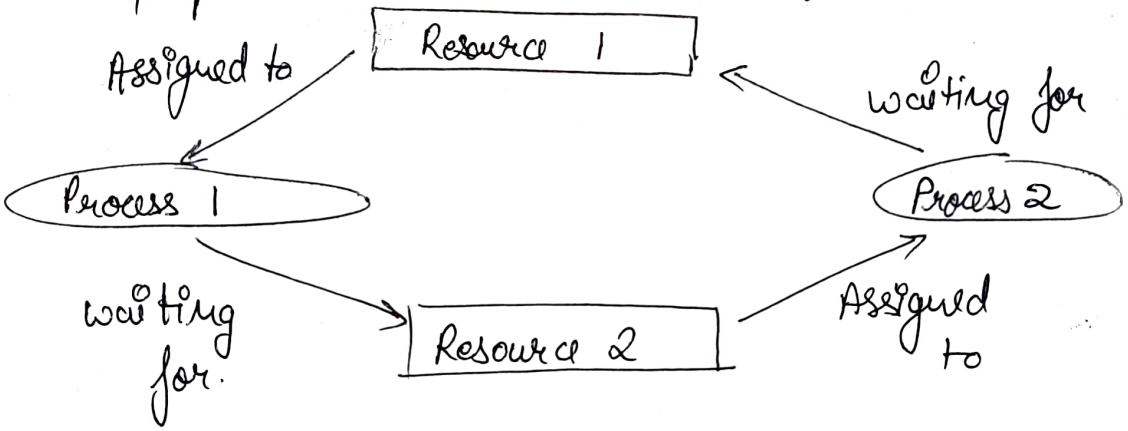
Any process wishes to access the shared variables in monitor, need to access through procedures. Process line up in a queue and are only provided access.

Semaphore → lower level object. Semaphore is a non-negative integer variables. The value indicated by the numbers to resource available in system. Modified by wait() and signal() operations.

↳ Binary Semaphore
↳ Counting Semaphore.

b) Deadlock Detection →

* If a single instance :- Here, for detection, we can run an algorithm to check for the cycle in the Resource Allocation Graph.



* If multiple instenses :- Detection is necessary but not sufficient condition for Deadlock detection.

Deadlock Recovery :-

A traditional operating system such as windows doesn't deat with deadlock recovery as it is a time and space consuming process. Real-time operating systems use Deadlock recovery.