Harsh Gupta
(1802713039)

Operating System (PUT)

(Section - A)

(Answer - 1)

Inter process Communication (IPC) :- It is a mechanism which allows processes to communicate each other and synchronize their actions. The communication between these process can be seen as a method of co-operation between them. Processes can communicate with each other using these two ways.

Context Switching :- It is a procedure that a computer's CPU (central processing unit) follows to change from one task (or process) to another while ensuring that the tasks do not conflict.

(Answer - 2)

A kernel is the central part of an OS. It manages the operations of the computer and the hardware, most notably memory and CPU time. A micro kernel which only contains basic functionality. A monolithic kernel, which contains many device drives.

(Answer - 3)

Batch processing is a technique in which an OS collects the programs and data

together in a batch before processing starts. An OS does the ~~same~~ following activities related to batch processing - The OS defines a job which has predefined sequence of commands, programs and data as a single unit.

(Answer-4)

The major diff between program and process is that program is a group of instruction to carry out a specified task whereas the process is a program in execution. While a process is an active entity, a program is considered to be a passive one.

Value of counting semaphore = 7
After 20 P operations value of semaphore =
7 - 20 = -13

After 15 P operations value of semaphore =
-13 + 15 = 2

so The resulting value of semaphore is 2.

(Answer-5)

A state is safe if the system can allocate all resources requested by all processes (up to their stated maximum) without entering a deadlock state.

If a safe sequence does not exist, then the system is an unsafe state, which may lead to deadlock.

Harsh Gupta
(1802713039)

Date _____ Page no. _____

(Answer - 6)

Busy waiting means that the process is waiting for a condition to be satisfied in a tight loop. without relinquishing the processor Alternatively, a process could wait by relinquishing the processor, and block on a condition and wait to be awakened at some app. time in the future.

To avoid busy waiting, a semaphore may use an associated queue of process that are waiting on the semaphore, allowing the semaphore to block the process and then ~~wake~~ it wake it when the semaphore is incremented.

(Answer - 7)

Logical memory 8 pages $= 2^3$
Size of word $= 2^{10}$
Total logical address space = 10 + 3 = 13 bits

Physical memory 32 frames $= 2^5$
Size of word • 1024 $= 2^{10}$
Total physical address space = 5 + 10 = 15 bits

(Answer - 8)

(i) Overlays allocation Storage :-

- no internal fragmentation
- less overhead and segment taskes only.

Harish Gupta
(1802713039)

(ii) Compaction :-
- minimize the probability of external fragmentation
- Store the bigger processes in the memory

(iii) Page table:- of
- Allocation memory is easy and cheap
- more efficient swapping.

(iv) Segment table :-
- It solves the problem of internal fragmentation
- Segment table consume less space.

(Answer-9)

Seek time :-
Seek time is the time taken in locating disk ~~arm~~ arm to a specified track where the read/write request will be satisfied.

Latency time :-
It is the time taken by the desired sector to rotate itself to the position from where it can access read/write heads.

(Answer-10)

Disadvantages of single contiguous memory allocation
- longer memory access time
- guarded page tables
- inverted page tables.

Harsh Gupta
(1802713039)

(Section-B)

(Answer-11)

	Allocated			Maximum			Available		
Process	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	2	2	3	3	6	8	7	7	10
P2	2	0	3	4	3	3			
P3	1	2	4	3	4	4			

(i) Need Matrix = [Maximum] - [Allocated]

$P_1 = [368] - [223]$

$= [145]$

$P_2 = [433] - [203]$

$= [230]$

$P_3 = [344] - [124]$

$= [220]$

Need		
R1	R2	R3
1	4	5
2	3	0
2	2	0

Need Matrix :-

Process	Need		
P_1	1	4	5
P_2	2	3	0
P_3	2	2	0

(ii) Safe sequence will be $< P_3, P_2, P_1 >$

The system is in safe state where safe state if system can

Harsh Gupta
(1802713039)

Date Page no.

(Answer-13)

The different types of OS are:-

1. Batch Operating System:
   - Users of Batch OS don't interact with computer directly.
   - Each user prepares his job on an offline device like punch cards and submit it to the computer operator.
   - programmers, leave their programs with the operator and the operator then sorts program with similar requirements into batches

2. Time sharing operating System:-
   - Time sharing OS allows the many users to share the computer simultaneously.
   - A time shared OS uses CPU scheduling and multiprogramming to provide each user with a small portion of a time shared computer.

3. Distributed operating System:-
   - It uses multiple processors to serve multiple real-time applications and multiple users
   - Data processing jobs are ~~distributed~~ distributed among the processors accordingly.
   - They are referred as ~~tightly~~ loosely coupled system
   -

4. Interactive Operating System :-

- It allows the users to interact directly with the OS whilst one or more programs are running
- There will be an user interface to allow this to happen.

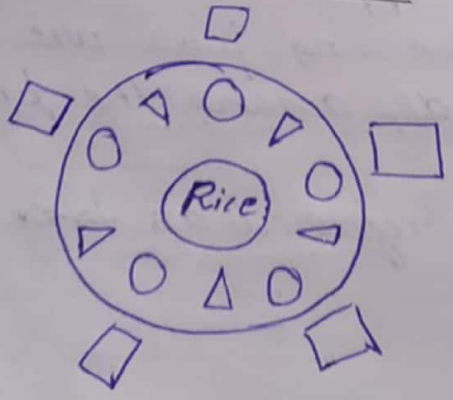
5. Network Operating system :-

- Network OS runs on a server and provides the server the capability to manage data
- Primary purpose is to allow shared files and printer access among multiple computers.
- eg :- UNIX, LINUX.

6. Real time Operating system :-

- A real time system is defined as data processing system in which the time interval required to process and respond to inputs is so small that it controls the environment.
- The response time is very less
  eg :- In scientific experiments, robot control system etc.

(Answer-15)



Harsh Gupta
(1802713039)

Date _____ Page no. _____

Dining philosopher's problem is a classic synchronization problem. There is a dining table with 5 chairs, at each chair is a plate and between each plate is a chopstick, in middle is bowl of rice

In order to eat, a philosopher must sit at the table and pick the left and right chopstick and eat the rice.

So we have to ~~fix~~ find a sol so that while one philosopher is eating other must not starve.

Solution:-

we can say for one case, that first a philosopher must pick left chopstick first and then the right But in some case deadlock will occur.

Using Semaphores:-

→ each philosopher is a process
one semaphore per fork:

fork : array [0---4] of semaphores

→ initialization:
fork[i], count = 1
for i := 0--4

process Pi:
repeat
think;

Harsh Gupta
(1802713039)

```
wait ( fork [i]);
wait( fork [ (i+1) mod 5]);

eat
 signal (fork [i+1 mod 5]);
  signal [fork [i]);
forever.
```

→ Now to remove the deadlock situation, we may fix that the last philosopher i.e. S4 must follow another

i.e. first he should pick right fork and then left fork

```
i.e. = wait (fork [i+1 mod 5]);
       wait (fork [i]);
```

∴ Deadlock will get removed

→ Another solution can be :-

odd numbered philosopher picks up first the left and the right

even numbered philosopher picks right then left.

Harsh Gupta
(1802713039)

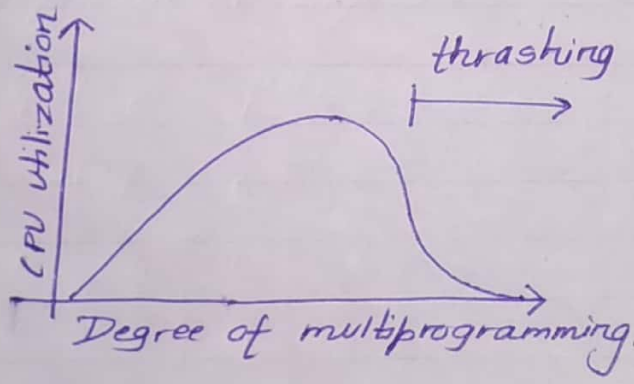
(Operating System)

(Section - C)

(Answer - 16 (a))

Thrashing:-

1. Thrashing occurs when a system spends more time in processing page faults than executing transcations.
2. While processing page fault is necessary to be in order to appreciate the benefits of virtual memory thrashing has a negative effect on the system.
3. While the transactions in the system are waiting for the paging device. CPU utilization, system throughput and system response time decreases, resulting in below optimal performance of a system.
4. Therefore to increase CPU utilization multiprogramming should be reduced.



Causes:-

1. A system thrashes if physical memory is too small to hold the working sets of all the process running
2. The upshot of thrashing is that pages always spend their time waiting for the backing store to fetch their pages.
3. Usually there is a ~~process~~ phase transition from no thrashing to thrashing.
4. Scheduler brought in new process whenever CPU utilization goes down.

5. Eventually the size of the working sets become larger than physical memory and processes start to page.

6. A process is thrashing if it is spending more time in paging than executing.

Detection of thrashing :-
A system can detect thrashing by evaluation the level of CPU utilization as compared to the level of multiprogramming.

Elimination of thrashing :-
The data cache pre-fetching and single assignment data move principle enables elimination of cache thrashing / thrashing.

(Answer - 17(a))

Optimal page replacement Algorithm :-

Request	4	7	6	1	7	6	1	2	7	2
Frame 3			6	6	6	6	6	2	2	2
Frame 2		7	7	7	7	7	7	7	7	7
Frame 1	4	4	4	1	1	1	1	1	1	1
Miss/Hit (M/H)	M	M	M	M	H	H	H	M	H	H

Number of pages faults in optimal page Replacement algo = 5

Harsh Gupta
(1802713039)

FIFO Page Replacement Algorithm:-

Request	4	7	6	1	7	6	1	2	7	2
Frame 3			6	6	6	6	6	~~6~~	7	7
Frame 2		7	7	7	7	7	~~7~~	2	2	2
Frame 1	4	4	~~4~~	1	1	1	1	1	1	1
Miss/Hit (M/H)	M	M	M	M	H	H	H	M	M	H

No. of pages faults in FIFO page Replacement algorithm = 6

LRU page Replacement Algorithm:-

Request	4	7	6	1	7	6	1	2	7	2
Frame 3			6	6	6	6	6	~~6~~	7	7
Frame 2		7	7	7	7	7	~~7~~	2	2	2
Frame 1	4	4	~~4~~	1	1	1	1	1	1	1
Miss/Hit (M/H)	M	M	M	M	H	H	H	M	M	H

No. of page faults in LRU page Replacement algorithm = 6

Harsh Gupta
(18027130 39)

Date Page no.

(Answer - 18(b))

Caching :-

- A cache is a region of fast memory that holds copies of data. Access to the cached copy is more efficient than access to the original
- For instance, the instruction of the currently running process are stored on disk, cached in physical memory and copied again in the CPU's secondary and primary caches.
- The difference between a buffer and a cache is that a buffer may hold the only existing copy of a data item, whereas a cache, by definition holds a copy on faster storage of an item that resides elsewhere.

Spooling :-

- A spool is a buffer that holds outputs for a device, such as printer, that cannot accept interleaved data streams.
- Although a printer can serve only one job at a time. Several applications may wish to print their output concurrently without having their output mixed together.
- The operating system solves this problem by intercepting all output to the printer.
- Each application's output is spooled to a separate disk file.
- When an application finishes printing, the spooling system queues the corresponding spool file for output to the printing, the spooling system copies the queue spool files to the printer one at a time.

Harsh Gupta
(1802713039)

Error handling :-

- An operating system that uses protected memory can guard against many kinds of hardware and application errors, so that a complete system failure is not the usual result of each minor mechanical glitch.
- Devices and I/O transfers can fail in many ways, either for transient reasons, as when a network becomes overloaded, or for "permanent" reasons, as when a disk controller becomes defective.
- Operating system can often compensate effectively for transient failures
- As a general rule, an I/O system call ~~with~~ will return one bit of information about the status of the call, signifying either success or failure.

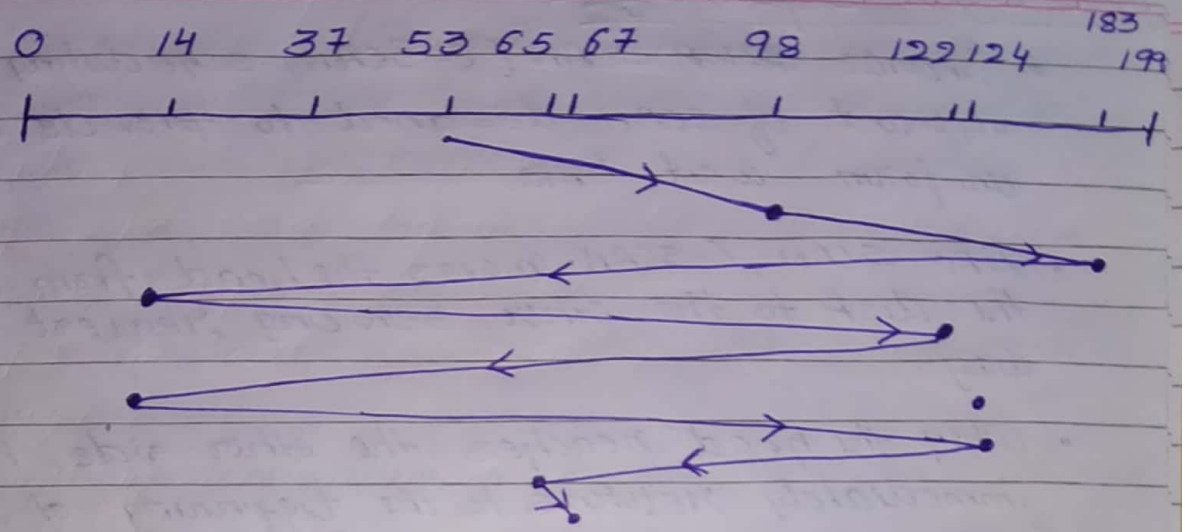
(Answer - 19(b))

(i) FCFS scheduling :-

- The simplest form of disk scheduling is the First-come First Served (FCFS) Algorithm.
- This algorithm is intrinsically fair, but it ~~generates~~ generally does not provide the fastest service.
- eg:- Consider, a disk queue with requests for I/O to blocks on cylinders
  98, 138, 37, 122, 14, 124, 65, 67 in that order
- If the disk head is initially at cylinder 53, it will first move from 53 to 98, then to 183, 37, 122, 14, 124, 65 and finally to 67

Harsh Gupta
(1802713039)

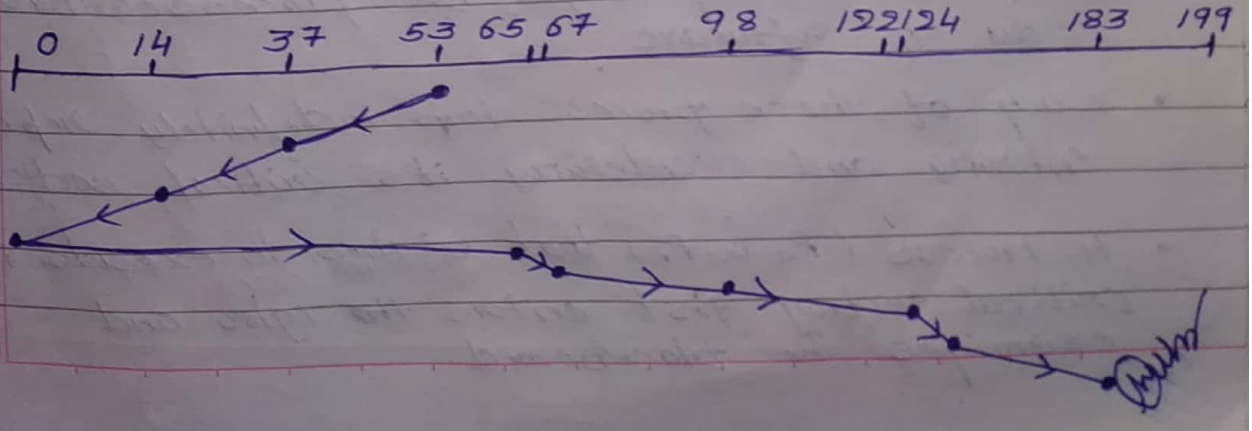
Date Page no.



queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53.

(ii) SCAN scheduling :-

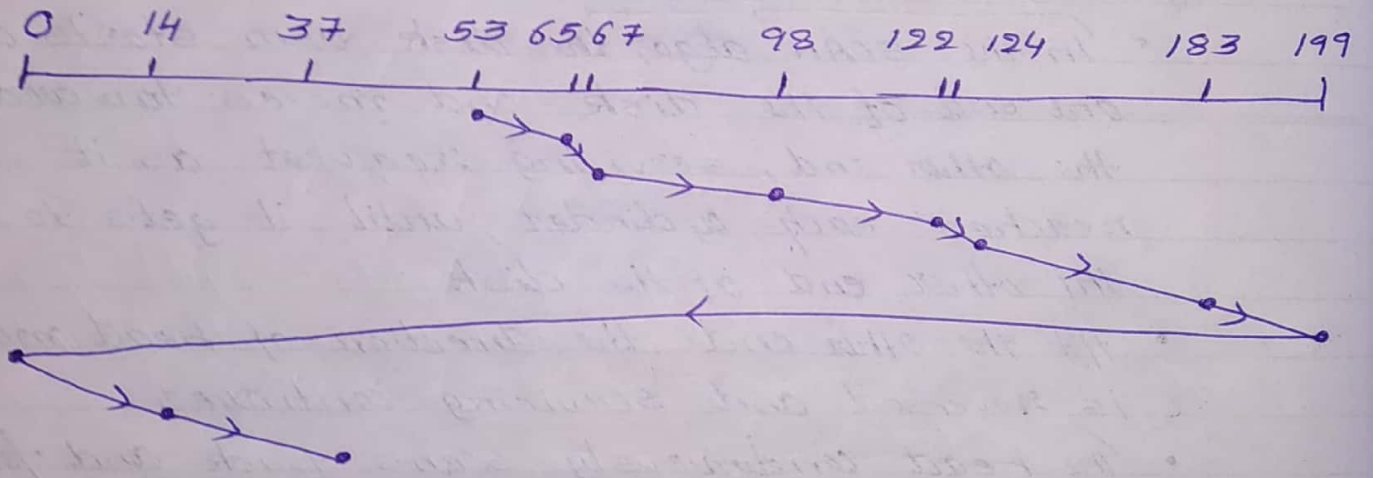
- In the SCAN algo, the disk arm starts at one end of the disk and moves towards the other end, servicing request as it reaches each cylinder, until it gets to the other end of the disk.
- At the other end, the direction of head mov. is reversed and servicing continues.
- The head continuously scans back and forth across the disk
- Assuming the disk arm is moving towards 0 and the initial head position is again 53.



Harish Gupta
(1802713039)

(iii) C-SCAN Scheduling :-

- Circular ~~SCAN~~ SCAN (C-SCAN) scheduling is a variant of SCAN designed to provide a more uniform wait time.
- Like SCAN, C-SCAN moves the head from one end of the disk to the other, servicing request along the way
- When the head reaches the other side, however, it immediately returns to the beginning of the disk without servicing any requests on the return trip
- The C-SCAN scheduling algorithm essentially treats the cylinder as a circular list that wraps around from final cylinder to the first one.



(Answer - 20(a))

- Dekker's solution is for two processes based solely on software
- Each of these process loop indefinitely, repeatedly entering and reentering its critical section.
- A Process (P_0 & P_1) that wishes to execute its critical section first enters the igloo and examines the blackboard.

Harsh Gupta
(1802713039)

Date Page no.

- The process number is written on the blackboard, and that process leaves the igloo and proceeds to critical section
- Otherwise the process will wait for its turn
- Process reenters in the igloo to check the blackboard
- It repeats this exercise until is allowed to enter its critical section. This procedure is known as busy waiting.

- Busy waiting occurs when a process is in its critical section, any other process that tries to enter its critical section must loop continously in the entry code.
- This kind of problem occurs in multiprogramming system

```
do {
        wait(mutex);    // critical section

        signal (mutex); // remainder section
} while ( TRUE);
```

- Busy waiting waste CPU cycle that some other process might be able to use productively.