Mansi Uniyal
19EE10039.

Q1. (a) How I/O is handled by OS?

Ans. OS allocates buffer in memory and informs I/O device to use it to send data to CPU while using memory mapped I/O. I/O device operates asynchronously with CPU, interrupts CPU when finished.

Q1. (b) What is DMA? How it differs from normal I/O?
(Direct memory access)

Ans. It is a method to allow I/O device to send or recieve data directly to or from the main memory, by passing the CPU to speed up memory operations. DMA Controller manages this process.

- I/O is software control data transfer, whereas DMA is hardware —"— .
- I/O has slower data transfer speed.
- In I/O CPU is involved in complete transfer unlike DMA.
- I/O is used for small data transfer.

Q1. (c) What is dual mode of execution?

Ans. In Dual mode, it protects operating system from illegal users. We acomplish this by designating some of system instructions as priveleged one that can cause harm. ∴ Hardware only allows for execution of priveleged instruction in kernel mode.

MANSI UNIYAL
19EE0039

Q1. (d) Discuss Client-server computing & peer to peer computing.

Ans. Client-Server computing :-
Client requests resource and server provides that resource.
A server may serve multiple clients at same time while a client is in contact with only 1 server.
Both client & server communicate via a computer network but sometimes they may reside in same system.

Peer to peer computing :-
It contains nodes that are equal participants in data sharing.
All tasks are equally divided bet" all the nodes.
Nodes interract with each other as required as share resources.

Q1. (e) What is interrupt? How it will be handled?

Ans. Interrupt is a signal from a device attached to computer that causes the main program that operates computer (OS) to stop & figure out what to do next.

Q1. (e) Interrupt handling mechanism of OS accepts a number which is an address and then selects what specific action to be taken which is already mentioned in interrupt service routine. In most architecture, address is stored in a table called as vector table..

Q1. (f) What is cache coherency?

Ans. It is uniformity of shared resource data that ends up stored in multiple local cache. When clients in system maintain caches of common memory resource, problems may arise with incoherent data. (which is the case with CPU in multiprocessing system)

Q1. (g) How cluster system dif" from multiprocessor systems? What are objectives of OS in case of cluster system?

Ans. Cluster system are less tightly couple than multiprocessor system. Clustered system use messages to communicate while processor communicate using shared memory.

MANSI UNIYAL
19EE10039.

(4)

main goal of clustered system is to provide help in weather forecast, scientific computing and super computing system.

Q2 (a)

Ans: System call interface is middleware bet^n user program/application and system call.
System call interface maintains a table indexed according to members associated with each system call. It invokes the system call using API in OS kernel and returns status of signal call to user program.

MANSI UNIYAL

19EE10039.

Q2.(b) What is core dump & crash dump? Where are they used?

Ans. Crash dump:-

failure in kernel: error info is saved to log or memory file and state is saved.

It is created when process contents are written to file. this can happen automaticaly if application has been set up to create them upon as exception or if exception bubbles up to underlying OS.

Core dump:- It is printing or copying to more permanent medium (hard disk). contents of RAM at one moment in time. One can thin of it as full length snapshot of RAM. It is taken mainly for purpose of depugging a program.

OS captures memory of the process & stores in file for later analysis.

They both are methods used by OS in failure analysis.

Q2.(c) Micro kernels:- (6)

contains min. number of func" required, data or features.
It is small in size & has more security & reliability.
Eg:- MacOSX, Windows NT.

MANSI UNIYAL
19EE10039.

Q2 (c) Modular kernels:-
Some parts of system core gets allocated in independent files called modules, which can be added ~~to~~ at runtime. This has less security & reliability, decreased boot time and faster development.
Eg:- Linux.

Q2. (d)
Ans. Various user OS interfaces are:-
(i) Command line (CLI)
(ii) Graphical UI (GUI)
(iii) Form based interface (FBI)
(iv) Natural language interface (NLI)

Q2. (e) (vm).
Ans. Virtual machines creates virtual environment that stimulates a physical ~~environment~~ machine. It has its own CPU, memory, network interface & storage but are independent of physical hardware.

MANSI UNIYAL

(8)

MEE 10039.

Q2. (e) Benefits:- 1

- Can exist on same server.
- Each is isolated from another
- Can be saved to file and moved like files.

Q2. (f).

Ans. System programs help in building system software for OS. Program development and execution can be done easily in system program.

Eg:- file management system.

Q3 (a) (i)

Parent process creates new child process using fork() system call. This new child process is duplicate of parent and inherits parents address space.

To replace address place of child process, exec() system call is used

(ii) User can make parent process wait using wait() system call, until child process terminates. It can also allow both the proces to run simultaneously.

(iii) During termination ~~of~~ a child process returns status data to parent via wait().

If no parent is waiting then proces is zombie process.

If parent terminates without ~~save~~ wait, proces becomes orphan. Some Os don't allow such orphan child to exist.

Q3 (b) IPC communication first est. a region of shared memory which resides in address space of process creating shared memory.

Other process in communication must attach this shared region to address space.

Now process can read.

MANSI UNIYAL
19EE10039

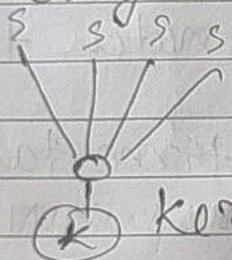
Q4.(a)

Ans. Threads are considered as light weight processes because they take less time for creation and lesser time to terminate than full process.

Advantage :- (Multiprocess)

(i) Responsiveness
(ii) Resource sharing
(iii) economy
(iv) Scalability

Q4. (b): Many to one model.



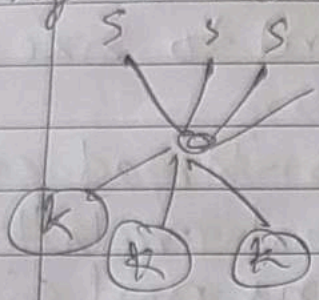
It maps many users levels threads to one kernel thread.

Pros & cons :-

- Thread management is done by thread library in user space so it is efficient.

- The entire process will block if a thread makes a blocking system call.
- As only 1 thread can access the kernel at a time, multiple threads are unable to run in parallel on multicore systems.

Many to many model.



multiplies many user level threads to smaller or equal number of kernel threads.

Pros & cons :-

- Allows developer to create as many user thread as wishes, it doesn't results in true concurrency as the kernel can schedule only one thread at a time.
- It suffers no short comings, Develops can create as many user thread and corresponding kernel threads to run in parallel on multiprocessor.

MANSI UNIYAL (11)
19EE10039.

Q4(c) (i) thread pool.
Creates a number of threads in pool where they wait for work.

Pros and cons :-
- Faster to service a request with an existing thread than create a new thread
- Allows number of threads in application to be bound to size of pool.
- Windows API supports thread pools.
Sophisticated thread pool architecture :- dynamically vary pool size.

They achieve concurrency of execution in computer program. Reuses prev. used threads to execute current tasks and offer solution to problem of thread cycle overhead and resource thrashing.

MANSI UNIYAL (12)

PEE1033.

Q4 (c)(ii) thread cancellation

Terminating thread before completion. A thread that is to be cancelled is often refer to as target thread. Cancellation of target thread can be asynchronous or deferred cancellation.

Asynchronous :- 1 thread immediately terminates target thread

deferred :- target thread periodically checks wether it should terminate.

(iii) Scheduler activations. (upcall handler)

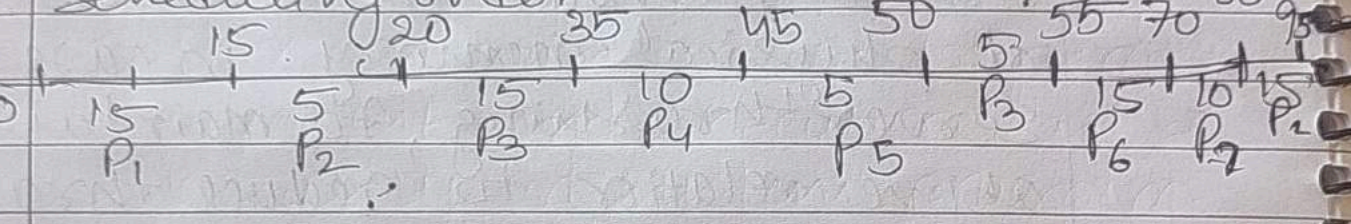
It communicates bet" kernel and thread library which may be required by many to many and 2 level models. Kernel provides application with set of virtual processors (LWP) and application can schedule user threads onto available virtual processor. Called as Upcall. It is handled by thread library with upcall handler.

MANSI UNIYAL
19EE10039

Q5(a)

Process	Priority	Burst	Arrival
P_1	8	15	0
P_2	3	20	0
P_3	4	20	20
P_4	4	20	25
P_5	5	5	45
P_6	5	15	55

(i) Scheduling order



(ii) Turn around time = Burst Time + Waiting time

$P_1 = 15$
$P_2 = 85$
$P_3 = 35$
$P_4 = 55$
$P_5 = 5$
$P_6 = 15$

(iii) Waiting time

$P_1 = 0$
$P_2 = 15+10+5+5+15+10+15 = 75$
$P_3 = 10+5 = 15$
$P_4 = 10+5+5+15 = 35$
$P_5 = 0$
$P_6 = 0$.

MANSI UNIYAL (14)
19EE10039

Q5 (b)

(i) Time quantum = 1ms
0.1ms overhead is incurred as CPU switches every 1ms.
for every 1.1 ms CPU utilizes only 1ms.
CPU utilization = $\frac{1}{1.1} \times 100 = 3\%$.

(ii) Time quantum = 10ms
CPU bound process can use full 10ms time slot, whereas I/O bound process can have it only 1ms because another I/O bound is in queue that will snatch time from it.

CPU bound proces time takes 10ms
10 I/O bound process would take
$10 \times 1 = 10$ ms.

CPU would be utilized for 20ms out of 21.1 ms.
Total time = $10 \times 1.1 + 10.1$
$= 21.1$ ms.
CPU utilization = $\frac{20}{21.1} \times 100 = 95\%$.

MANSI UNIYAL
19EE10039

Q5(c)
Ans. $T_{n+1} = \alpha t_n + (1-\alpha) T_n$.

$\alpha = 0$

(i) $T_0 = 100$ ms.

$T_{n+1} = 100$ ms.

So in this case the formula always makes prediction of 100 ms. for next CPU burst.

(ii) $\alpha = 0.99$

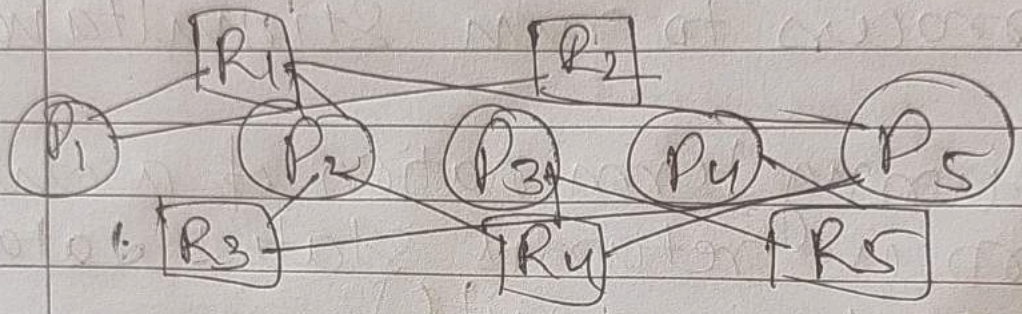
$T_0 = 10$ ms.

$T_{n+1} = 0.99\, t_n + 0.01 \times 10$ ms

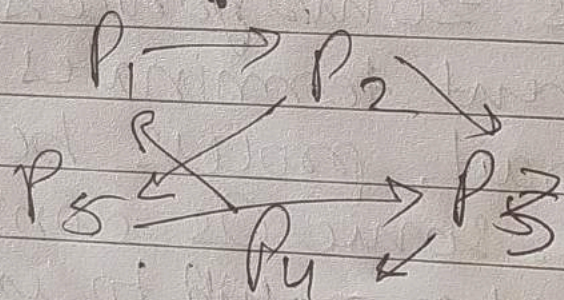
So, most recent behaviour t_n of process is given much higher weight than past history associated.

Q7 (a) (i)

Reserve Allocation Graph.



Wait for graph.

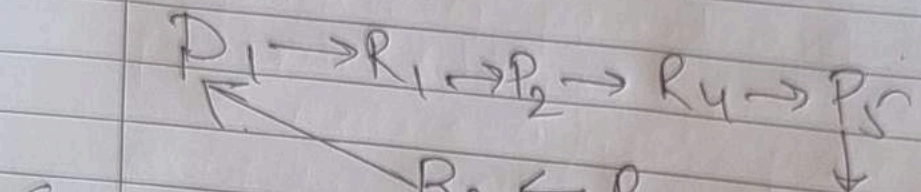


MANSI UNIYAL
19EE10039.

Q7 (a) (ii)

reserve allocation graph does contain cycle.

(iii) cycle path



Sequence of P_1 P_4 P_2 P_3 P_4.

Available.	A	B	C	D
t=0	0	4	2	2
1	1	7	3	3
2	2	9	6	7
3	6	10	9	8
4	10	11	10	9
5	11	11	11	10

All can be completed System is safe state.

(iv) Reserve request < need.

If granted → available 0 1 1 2
path unresolvable (unsafe) state.

Request can't be granted.

MANSI UNIYAL

19EE10039.

Q3 (b)

Process	Allocation	max	Available
	A B C D	A B C D	A B C D
P_0	1 0 1 1	5 3 2 4	0 4 2 2
P_1	1 3 1 1	1 7 2 3	
P_2	4 1 3 1	6 4 5 1	
P_3	4 1 1 1	8 3 4 2	
P_4	1 2 3 4	2 7 4 5	

(i) Total reserves

A = 11 B = 11 C = 11 D = 10

(ii) Need matrix

P_0	4 3 1 3
P_1	0 4 1 2
P_2	2 3 2 0
P_3	4 2 3 1
P_4	1 5 1 1

Q2.

(c) Policies to recover.

(i) Abort all process
→ slow as CPU time wasted

(ii) Abort one by one.
→ can be done by selecting victim
→ Downside is that may cloud to starvation.

(iii) Rollback
→ return to safe state
→ CPU wastage.

MANSI UNIYAL

PEE 100 39.

Qs. (a)

(i) Page size = frame size

no. of page = $\frac{2^{32}}{4 \times 2^{10}} = 2^{20}$

frames = $\frac{512 \times 2^{20}}{2^{12}} = 512 \times 2^{8}$.

page table entries = 2^{20}

Inverted page entry = 512×2^{8}

$= 2^{17}$.

(ii) each PTE needs to store bits needed to represent no of frames.

no. of frames = 2^{17}

no. of bits needed = 17 bits

Storage = $2^{20} \times 17$ bits

pid = x bits

Space = $2^{17} \times (20 + x)$ bits

(iii) If no TLB, memory not needed

= 2 access

∴ 400 ns

(iv) $0.75(2+50) + 0.25(102) = 64.5$ ns.

Q8. (b) first fit:

500k, 600k, 200k, not possible

best fit:

300k, 500k, 200k, 600k

worst fit

600k, 500k, 300k, not possible

Best fit makes best use of memory

Q8. (c) L1 : seg = 0 offset = 384

L2 seg = 6 offset = 7

L3 seg = 1 offset = 15

L4 seg = 3 offset = 51.

L1 physical = 603

L3 physical = 2307

L2 physical = invalid

L4 physical = 1838.

MANSI UNIYAL
MEE10039.

Q9.(a)

Ans. p = page fault rate
= probability that a memory access reacts to a page fault.

Probability that memory costs
100 nsec = 1-p
Probability that page fault cost 20 ms
= 0.7p
Probability that page fault costs 80 ms
= 0.3p.

(1) $(1-p)100 + 0.7p \times 20 \times 10^6 + 0.3p \times 8 \times 10^6 = 200$.

(2) $(14 \times 10^5 + 24 \times 10^5 - 100)p = 100$.

$p = \frac{100}{16400100} = 61 \times 10^{-6}$

$\boxed{p = 0.000\,61\,\%}$

Q9.(b) Page fault is generated if processes present in system doesn't have enough pages with new ones, soon after that again page faults occurs requesting for just recently removed pages.

MANSI UNIYAL
19EE10039

MANSI
19EE1

Q9.(b) Max. amount of time will be spent for disk I/O as this process continues and CPU will be mostly idle waiting for new pages. Moreover, overall performance is degraded as OS will load new processes from disk to increase multiprograming for enhancing CPU utilization. This phenomena is called thrashing.

Q9.(b)

Working set model :-
Set of pages referred in recent time interval is known as working set of process at that time.
OS: {0, 2, 0, 0, 2, 2, 2, 4, 4}.
for this history working set model is {0, 2, 4}.

Tracking of working set :-
This is to decide which page from present working set has to be replaced if the page fault occurs. This is based on page usage in previous hiss relation between WSM and size of physical memory.

MANSI UNIYAL

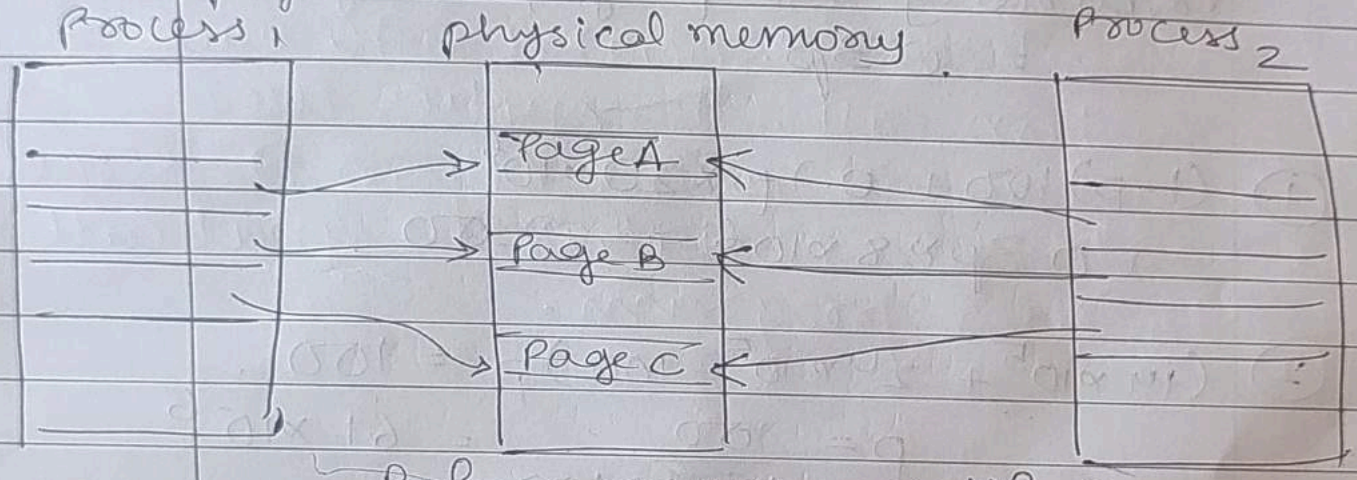
19EE10039

Q9.(b) If the size of sum of optimal WSM of all processes present in memory is greater than size of physical memory.
To avoid thrashing we need to ensure that size of sum of optimal WSM of all process present in memory should be less that size of physical memory. In case of server thrashing, to maintain above relation to reduce thrashing, we need to swap out some process to disk.

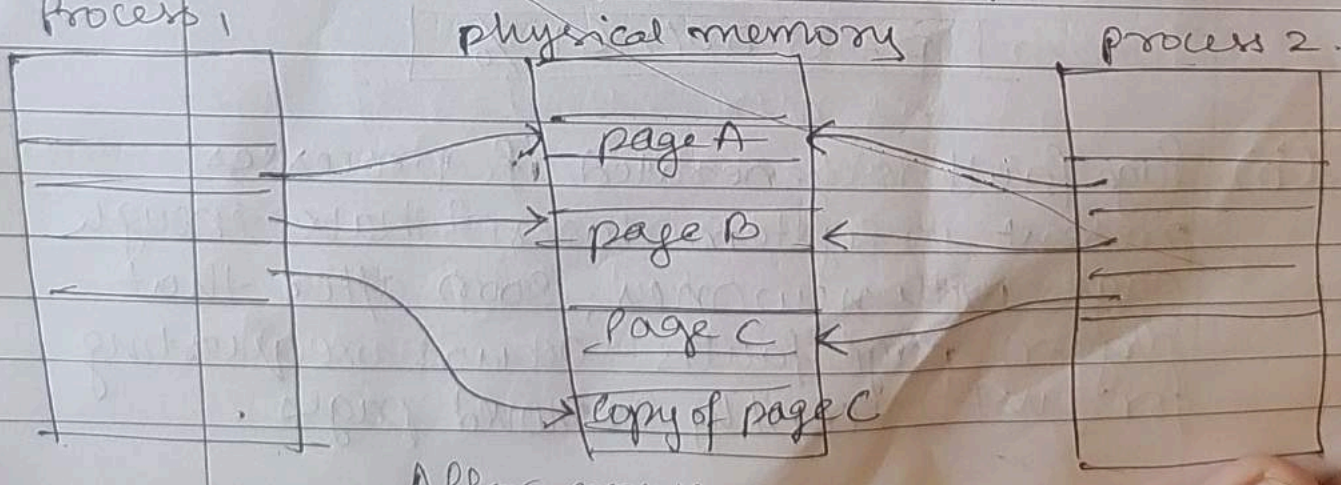
MANSI UNIYAL
19EE10039.

Q10 (a) Copy on write. (COW)

Ans. It allows parents & child processes to initially share same pages in memory If either modifies a shared page, only then is the page copied. It allows more efficient process creation as only modified pages are copied. Free pages are allocated from pool of zero fill on demand pages. Pool has free frames for fast demand page execution.



Before process 1 modified.



After process 1.