Page NO.

1. 1 to 24 – Mid Term
2. 25 to 59- Final Term

MD IFTAKHAR KABIR SAKUR

25th BATCH

COMPUTER AND COMMUNICATION ENGINEERING

International Islamic University Chittagong

COURSE CODE: CCE-4705

COURSE TITLE: Operating System

COURSE TEACHER:

Mohammad Nadib Hasan

Lecturer
Computer and Communication Engineering

Operating System

CCE-4705

□ **Operating system:-** It's a collection of software that mangges Computer, hardware, resources & provides various services for computer programs. It acts as an intermediary between the user & and computer hardware

□ **Abstract view:-**

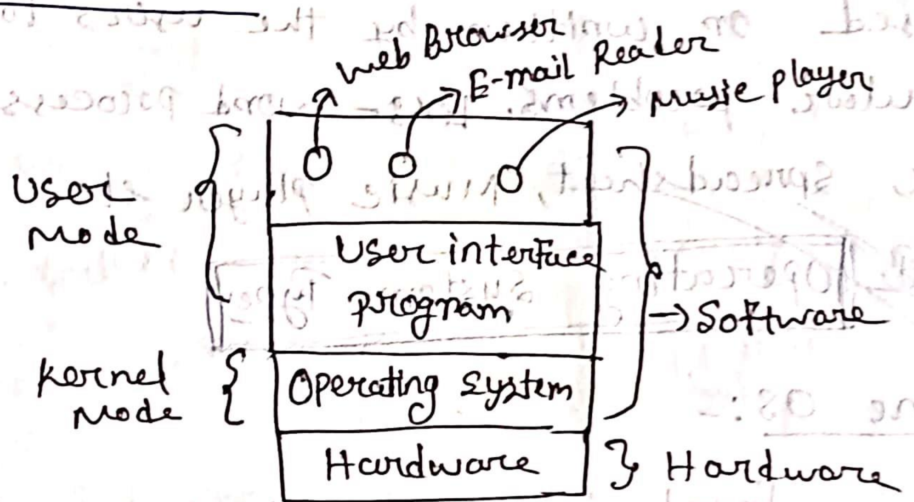


Fig:- Abstract view.

Hardware:- Ic, chips, wires, disks, a key board, A monitor & similar physical devices.

~~Software~~:-

Operating System:- Run on hardware & provides base for software.

Most computer have 2 options or 2 modes of operations

(i) Kernel Mode:-

In kernel mode it has complete access of all the hardware & can execute any instruction that the machine is capable of executing.

(ii), User Mode:- Software runs in this mode. Here we find command interpreter (shell), compilers, editors & other system programs.

Application programs:- It is above all. These programs are purchased or written by the users to solve their particular problems. Ex:- word processing, web browser, spreadsheet, music player etc.

Operating System Types

(1) Mainframe OS:-

→ This OS is found in room sized computers which are found in major corporate offices. It has different from personal computer based on the I/O.

→ It does three major services!

(2) Batch OS:- This one processes routine jobs without any interactive user presents, such as claim processing in an ~~to~~ insurances & sales reporting etc.

(ii) Transaction processing System :-

Handles large number of small requests for ex:- check processing at a bank & airline reservation.

(iii) Time sharing :- Allows multiple remote users to run jobs on the computer at once, such as ~~querying~~ a querying a database.

Ex:- OS/390 & descendant of OS/360.

② Server os:-

=> They run on servers. ~~High~~ very large personal computers, workstations or even mainframes.

=> Serve multiple users at once over a network & allow the users to share hardware & software resources.

=> The server provides : print service, file service & or web service.

=> Typically server OS are Solaris, FreeBSD, and Linux & windows server 200x.

Multiprocessor OS:-

~~Co~~ To get major group computing power is to Connect multiple CPUs into a single system. But it depends on what they share how they interact. These systems are called as Parallel Computers, Multicomputers, Multiprocessors.

-> It has special Features (communication, connectivity, consistency)

-> Ex:- Windows, Linux

□ personal computer OS:-

-> All modern computer have this ~~multiple~~ multiprogramming, often with more than one programs starting up at boot time.

-> It ~~support~~ provide good support to a single user.

-> Ex:- Windows, Linux Macintosh OS.

□ Handheld OS:

-> It is for the device which can fit in hand or in pocket.

It is a small computer or PDA (personal Digital Assistant) ~~is~~ that performs small number of operations. Ex:- Electronic Address

Memo pad.

The functions are like Telephony, photography & other functions.

-) Difference between personal Computer OS & Handheld don't have GB harddisk.

-) Eng- palm os, Symbian os.

Embedded OS:-

=) This runs on computer that are not generally computers. And do not accept user installed software.

-) Untrusted s/w will never run on it.

-) NO need for protections between applications.

Ex:- QNX, Vx works, etc.

Sensor Node OS:-

=> Networks of tiny sensors nodes are being & developed for numerous purposes. These nodes are tiny computers that communicate with each other & with a base station using wireless communication.

=) These sensor networks works to protect buildings helps Border guards, Detect Fires in forests, Measure temperature & weather forecasting glean information about enemy movement in Battle Fields.

=) Each sensor Node have CPU, RAM, ROM like real computer.

=) All the programs are loaded in advanced

=) Ex:- Tiny O S.

◫ Real-Time OS:-

-) Time is a key parameter

-) It has fixed time constraints! Specific process must be done within given time or the system will fail.

Types:- 2 types:-

① Hard Real Time System:-

-) This is used in Companies (industries), military & similar applications areas.

-) A certain thing will occur in a certain time.

② Soft real time systems:-

-) Missing an occessional deadline, while not desirable is acceptable but it does not do any permanent damage.

-) Exa:- Digital Audio, Digital telephone & multimedia systems. For Ex:- e-cos.

[Q] Give the Features Of Batch Operating system:-

The OS which processes jobs in batch system without requiring a constant user intervention. (Earlier computers)

1) Job scheduling:- Does multiple jobs ~~so~~ in a sequential Order. But there are criterias for this. Like priority, execution time, and resource ~~ability~~ availability.

2) No user interaction:- Little user interaction during job execution. Once Jobs are submitted next part comes autometically.

3) Job Control Language (JCL):- Often use scripted language JCL.

4) Claim processing in insurance & sales reporting etc.

5) To improve utilization this Concept was developed.

6) Jobs with similar needs are batched together & were run through the Computer as a group.

7) Sort program into batches.

8) The operator then loads a special program, which reads ~~from~~ ~~first~~ the first job from magnetic tape & run it.

→ The output is written in second magnetic tape instead of printing.

→ After finishing one job the OS automatically reads the next one.

→ when whole batch is done the operator removes Input & output tapes & replace with the next batch. And brings output tape for Offline printing.

→ With the use of this type of OS, the user no longer has direct access to machine.

→ Advantage:-

→ Disadvantage:-

Scheduling Algorithm

way of selecting a process from ready queue & put it in the cpu.

(Interrupt) ← (i) pre-emptive (ii) Non-preemptive (No Interrupt. complete full tasks)

- FCFS:- First Come ~~Fast~~ First Service

Process	AT	BT
P1	0	2
P2	1	

pre-emptive:-

- → SRTF (Shortest Remaining Time First)
- → LRTF (Longest Remaining Time First)
- → Round-Robin (RR)
- → priority Based.

Non-preemptive:-

- → FCFS (First Come First Serivice)
- → SJF (Shortest Job First)
- → LJF (Longest Job First)
- → HRRN (Highest Response Ratio Next)

Scheduler:- The port of an operating system that makes the choice is called scheduler.

Arrival Time (A.T):- Time at which, process enter the ready queue

Burst Time (B.T):- Time required by a process to get executed.

Completion Time (C.T):- Time at which, process Completes its execution.

Total Around Time (T.A.T):- T.A.T = C.T − A.T

Waiting Time (W.T):- W.T = T.A.T − B.T

R.T (Response Time):-

In non-preemptive response time is equal to waiting Time(W.T).

Time at which process get CPU First time = R.T − A.T

FCFS (Non-preemptive)

				→CT−AT	TAT−BT	
process	AT	BT	CT	TAT	W.T	R.T
P1	0	2	2	2	0	0
P2	1	3	5	4	1	1
P3	4	4	9	5	1	1
P4	7	5	14	7	2	2
				18	4	

So, Avg TAT = $\frac{18}{4}$ = 4.5 unit

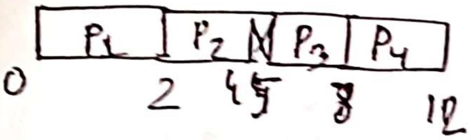
Avg W.T = 4/4 = 1 unit

P1	P2	P3	P4

0 2 5 9 14

FCFS - 2 :-

Process	A.T	B.T	C.T	T.A.T	W.T	R.T
P_1	0	2	2	2	0	0
P_2	1	2	4	3	1	1
P_3	5	3	8	~~2~~3	0	0
P_4	6	4	12	~~8~~6	2	2
				14	3	3

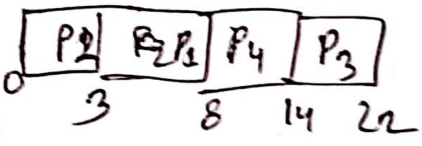


Avg. WT = $3/4$ = 0.75

Avg. T.AT = $14/4$ = 3.5

SJF (Shortest Job First) :- [Burst Time ছোট হলে আগে]

Process	AT	B.T	C.T	TAT	W.T	R.T
P_1	0	5	8	8	3	3
P_2	0	3	3	3	0	0
P_3	2	8	22	20	12	12
P_4	3	6	14	11	5	5
				42	20	



Avg TAT = $\frac{42}{4}$ = 10.5

Avg W.T = $\frac{20}{4}$ = 5

(Pre-emptive) Shortest Remaining First ~~Job~~ (SRTF)

(AT- [illegible] time is
OS [illegible]

P	A.T	B.T	C.T	T.AT	W.T	R.T
P_1	0	12	27	27	15	0-
P_2	2	4	6	4	0	2-2=0
P_3	3	6	12	9	3	6-3=3
P_4	8	5	17	9	4	12-8=4
				49	22	

P_1	P_1	P_2	P_2	P_2	P_2	P_3	P_3	P_3	P_3	P_3	P_3	P_4	P_1	P_1

0 1 2 3 4 5 6 7 8 9 10 11 12 17 27 ~~27~~

~~Avg = wt = 49/4~~

Avg = TAT = 49/4 = 12.25

Avg WT = 22/4 = 5.5

SRTF-2

CT-AT

P	A.T	B.T	C.T	T.A.T	W.T	R.T
P_1	0	5	9	9	4	0
P_2	1	3	4	3	0	0
P_3	2	4	13	11	7	7
P_4	4	1	5	1	0	0
				24	11	

P_1	P_2	P_2	P_2	P_4	P_1	P_1	P_1	P_1	P_3	P_3	P_3	P_3
0–1	1–2	2–3	3–4	4–5	5–6	6–7	7–8	8–9	9–10	10–11	11–12	12–13

$\therefore$ Avg $\rightarrow$ TAT $= \frac{24}{4} = 6$

Avg - WT $= \frac{11}{4} =$ ~~2.25~~ 2.75

Avg. RT $= 7/4 = 1.75$

(Ans)

Round Robin

⊛ Time quantum = 4 unit

P.T	A.T	B.T	~~TAT~~ CT	TAT	W.T	R.T
P_1	0	5 ①	17	17	12	0
P_2	1	6 ②	23	22	16	~~5~~ 4-1=3
P_3	2	3	11	9	6	8-2=6
P_4	3	1	12	8	7	
P_5	4	5	~~16~~ 24	20	15	
P_6	6	4	21	15	11	

P_1	P_2	P_3	P_4	P_5	P_1	P_6	[illegible]	P_2	P_5

0 4 8 11 12 16 17 21 [illegible] 23 24

P_1 P_1 P_1 P_1

→ P_2 P_3 P_4 P_5 P_1

P_2 P_2 P_2 P_2 P_3 P_4 P_5 ~~P_6~~ P_1 P_6 P_2

→ P_3 P_3 P_3 P_4 P_5 P_1 P_6 P_2

→ P_4 P_5 P_1 P_6 P_2

→ P_5 P_5 P_5 P_5 P_1 P_6 P_2

→ ~~P_1 P_6 P_1 P_6 P_1 P_2~~ P_5 P_1 P_6 P_2

→ ~~[illegible] [illegible]~~ P_2 → P_6 P_6 P_6 P_6 P_2 P_5

→ ~~P_2~~ P_2 ~~P_5~~ → P_2 P_2 P_5

→ P_5 → P_5

㊟ Why we need Round Robin ?

=> SJF এর ক্ষেত্রে কোনো program run এর জন্য Last moment পর্যন্ত wait করতে হয়। তো কোনো Important moment এ কোনো program execute করতে হলে Round Robin use করা হয়।

Priority Scheduling

P.T	B.T	Priority	W.T
P1	10	3	6
P2	1	1	0
P3	2	4	16
P4	1	5	18
P5	5	2	1

~~P2~~	~~P5~~	~~P1~~	~~P3~~

~~0 4 6 18~~

~~16 20~~

0 1 6

P2	P5	P1	P3	P4

0 1 6 16 18 19

2) Priority Column এর value অনুসারে Chart তৈরি হবে এবং ঐ priority এর সাথে B.T সমান হবে

7) Operating Systems-

It is a collection of software that controls the hardware & programs of computers. It act as an intermediary between computer hardware & the user of the computer.

Figure:-

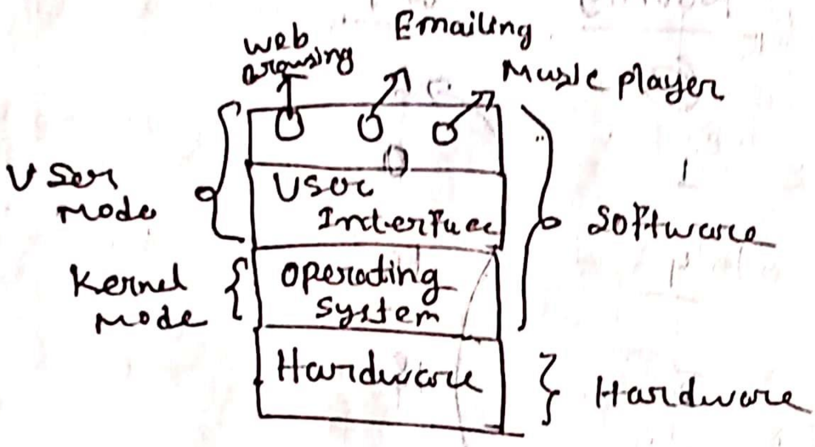


Fig:- operating system

OS Types:-

① Mainframe Operating system
② Personal Computer " "
③ Embedded " "
④ Server " "
⑤ Multiprocessor " "
⑥ Sensor Node " "
⑦ Real Time " "

Previous Question Solve

Spring -22

1) (a) Previous

(b) Justify the statement "OS" can be viewed as a government resource allocator & a control program.

=) ① Resource Allocation:-

-> Government Analogy:- Government allocates resources (Funds, Infrastructure, Service) to different sectors of society based on priority & needs.

-> Operating System Role:- The OS allocates resources efficiently. It manages task, priorities, & ensures each process receives each share to execute effectively.

② Control program:-

-> Government Analogy:- Establishes rules, regulations, policies to maintain law & order, ~~message~~ manage public services, and ensure the well-being of citizens.

-> Operating System Role:- It also has control program that oversees the execution of processes, enforce rules for memory protection, manages access to files & devices, and ensures security by controlling ~~cs~~ user access.

3] Enforce policies

→ Govt:- Makes policies to guide economic, social, and environmental activities. so, that they get better output from these.

→ OS :- OS enforce policies to resource allocation, security, user Access tool. It enforces rules about process execution, manages memory protection to prevent unauthorized access.

4] Fairness & optimization:-

→ Govt:- Govt distribute resources fairly, reduce inequalities, and optimize the functioning of societies.

→ OS:- OS aim to provide fair resource allocation, manage task scheduling to optimize CPU usage, and balance system load to prevent bottlenecks

5] Conflict Resolution:-

→ Government:- Mediate conflict between different groups or individuals to maintain social harmony.

→ OS:- In multitasking environment conflict may arise between resources. And ~~so~~ OS solve this.

3(a) principal problem of Non-preemtive : And how preemptive solved this.

-) ~~Pro~~ problem of ~~P~~ Non-preemptive:-

1| Inefficient Resource Utilization :- poor resource utilization.

2| Unresponsiveness:- High priority is blocked by lower one then High priority will have to wait.

3| Inability to Handle Real Time Requirements!

&

=) How preemptive solved this:-

⇒ 1| Efficient Resource Utilization:- By allowing interruption & switch to another process when a higher priority task becomes ~~or~~ available.

2| Responsiveness:- Higher-priority task can be scheduled to run as soon as they are ready.

3| Real Time Requirements:- Real Time requirements. High-priority tasks can interrupt lower-priority tasks.

4| Fairness:- Can be used to enforce fairness by assigning time slices to process In a Round-Robin way.

3(b) SRTF, → SJF Scheduling & which one is difficult ?

=>

Both SJF & SRTF algorithms involves selecting the process with the shortest burst Time, but SRTF adds the complexity of preemption & maintaining accurate Burst Time.

But SRTF also more careful considering of Content switching, Burst Time update.

And the choice of algorithm depends on the specific goals..

(3 OR)

a

P.T.O

3(a)(OR) Distinguish between process & Thread :-

Process	Threads
1) An independant unit of execution. of an OS. Has its own memory space, code, data, etc.	1) unit of execution within a process. Thread share same ~~t~~ resources of process.
2) Each process has its own memory specs, which includes the program's Code, data. & stack.	2) Threads within the same Process share the same memory spec.
3) ~~Processes~~: processes have their own system ~~Spec~~ resources. Such as Files, Network & devices etc.	3) Threads within a process share resources (Files, Network connections & devices)etc.
4) Process's scheduled managed by OS's process scheduler.	4) Scheduled by OS's thread scheduler.
5) Communicate using inter-Process Communication	5) Communicate easy way through shared memory.
6) Can involve higher overhead.	6) . Less overhead.

Aut-22

(1) (a) Distributive OS. And the Advantage Of it.

→ Distributed OS is a type of OS that runs on multiple Interconnected computers, Often referred to as nodes or hosts, and enables them to work together as a unified system.

Advantage:-

1] Resource Sharing:- Ability to share hardware resources

2] Fault tolerance:- IF one node Fails, the system can automatically redirect tasks to other available nodes to ensure continuity

3] Scalability:- Adding more nodes to network. And it makes possible to accomodate Increased workloads without a complete overhaul of the system

4] performance:- Task Can be executed In parallel mode, which helps to improve performance

5] Geographical Distribution:- Allows users from different locations to Collaborate & access resources remotely.

6] Cost Efficiency:- Instead of investing in cheap device distributed systems can utilize a network of cheaper & more readily available hardware.

7] Ease of maitanence

8] Load Balancing:-

9] Flexibility

10] High Availability

1(b)

prev

2(a)

prev

2(b)

prev

2(c)

Preemptive are complex:-

→ Scheduler interrupts the currently running process or thread.

→ Interruption Handling is complex.

→ Synchronization & Data sharing

→ priority Management

→ Real Time Consideration

→ Fairness

→ Rescheduling

→ Multicore

→ Debugging & testing

→ Predictability & performance.

3(b)

prev

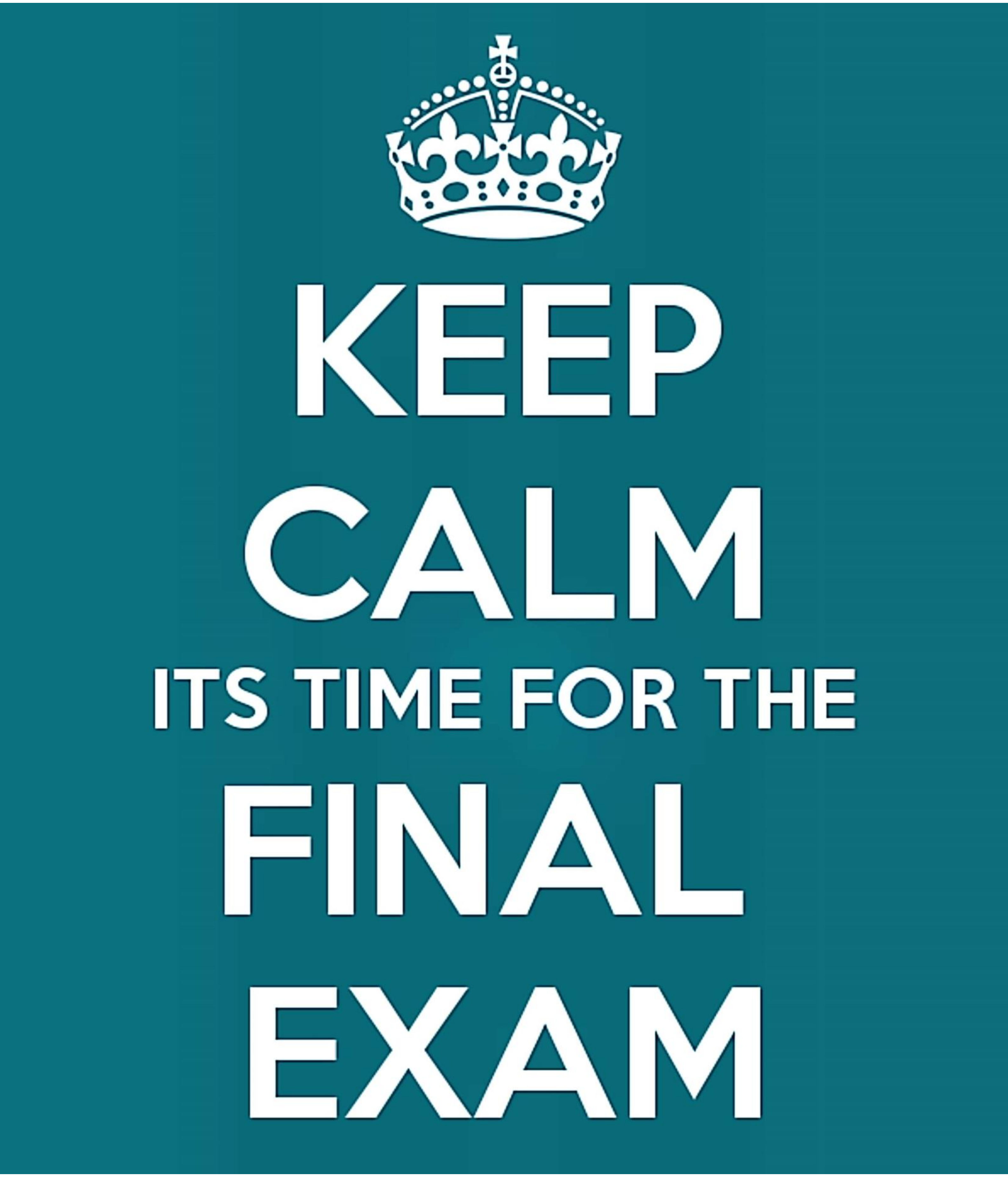
3(a) OR prev

3(b) OR

~~Recent~~

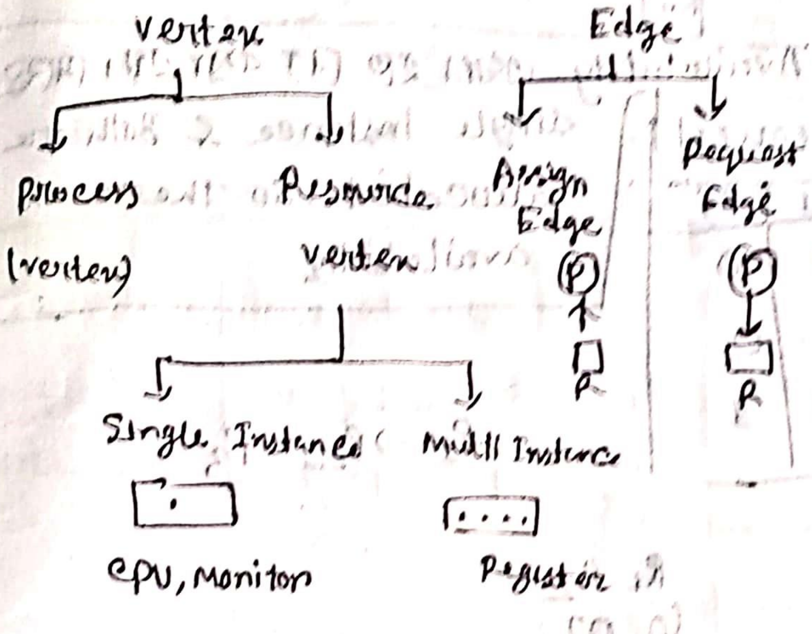
FCFS & SJF এর মধ্যে উদাহরণ দিতে হবে।

Then সবচে W.T কম SJF এটা prove করতে হবে

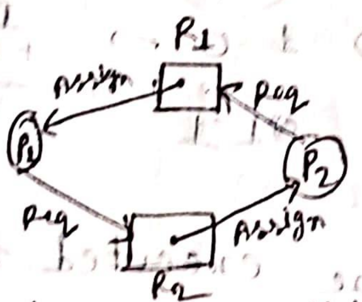

KEEP
CALM
ITS TIME FOR THE
FINAL
EXAM

FINAL

Resource Allocation Graph:-



Ex-01



Find if ~~it~~ there is a deadlock or not.

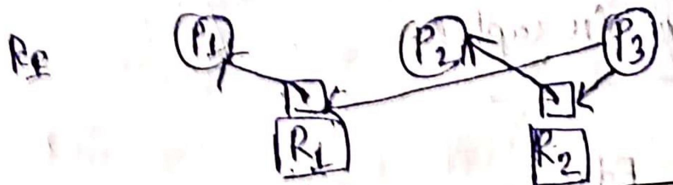
(Circular wait)

	Allocate		Request	
	R1	R2	R1	R2
P1	1	0.	0	1
P2	0	1	1	0

$$\therefore \text{Availability} \quad \begin{array}{cc} R_1 & R_2 \\ (0, & 0) \end{array}$$

As P1 or P2 request can't be fullfilled it is a deadlock.

Example-02 Multiple



→ Availability এখান হতে বের করতে হবে। যেহেতু single instance & Both are allocated. So there no availability

	Allocatie		Request	
	R_1	R_2	R_1	R_2
P_1	1	0	0	0
P_2	0	1	0	0
P_3	0	0	1	1

$P_1 \to P_2 \to P_3$

Availability (0, 0)

After P_1 process availability => 1 0

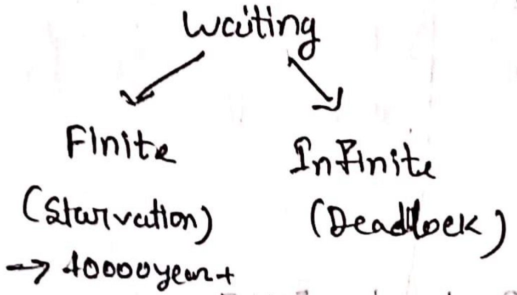
" P_2 " " ~~0~~ 1 1

R_1 R_2
(0 0)
1 0 →

$$\begin{array}{r} 1\ 0 \\ +\ 0\ 1 \\ \hline 1\ 1 \end{array}$$

So, all will be executed successfully

So, it is not a deadlock.

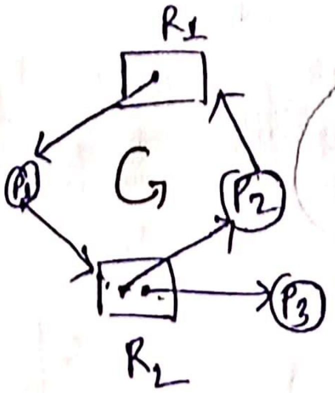


=> If RAG has circular (cycle & single instance there will be dead lock.

=> ~~If RAG doesn't have circular cycle & single instance there will be no deadlock~~

Ib) Multi instance RAG !-

1)



	Allocation		Request	
	R1	R2	R1	R2
P1	1	0	0	1
P2	0	1	1	0
✓P3	0	1	0	0

P3 → P1 → P2

Availability (R1 0 , R2 0)

0 1 ⟶ 0 1

1 0 ⟶ 1 0

1 1 ⟵ 1 1

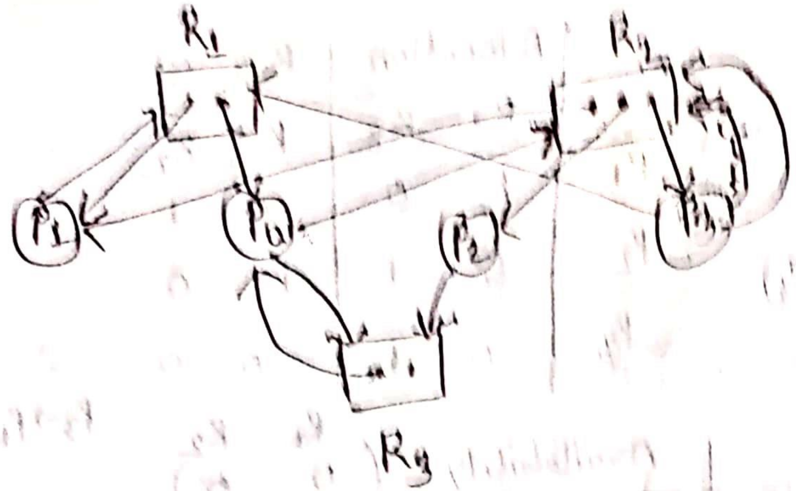
=) If the circular cycle but ~~the~~ multi instance There will be NO deadlock.

[P2 Need 1 for R1 & available was (1,1) so, R1 will get 1 for that.]

So there will be NO dead Lock As all of them will be executed

(Ans)

Example-02.



P	Allocate R1	R2	R3	Request R1	R2	R3
✗ P0	1	0	1	0	1	1
✓ P1	1	1	0	1	0	0
✗✓ P2	0	1	0	0	0	1
✓ P3	0	1	0	~~0~~1	1	0

P2 → P0 → P1 → P3

So, availability =

$$\begin{array}{ccc} R_1 & R_2 & R_3 \\ (0 & 0 & 1) \end{array}$$

0 1 0 → 0 1 1

0 1 1
1 0 1 → (1, 1 2)

1 1 2
1 1 0 → ~~1~~ 2 2 2

2 2 2
0 1 0

2 3 2 → Last availability

Condition:-
In Multiple Instance & circular wait there will be always no deadlock.

As all the ~~Resource~~ processes are terminating so there is no deadlock.

Bankers Algorithm

→ [Total Resource - Allocation]

→ [Max Need - Allocation]

process	Allocation			Maximum Need			Available			Remaining Need	Total Resources
	A	B	C	A	B	C	A	B	C		
✓P1	0	1	0	7	5	3	3 2	3 0	2 0		A = 10
✓P2	2	0	0	3	2	2	~~8~~3 2 5	~~8~~4 0 4	2 0 2		B = 5
P3	3	0	2	9	0	2	5	3	2		C = 7
P4	2	1	1	4	2	2	2 7	1 4	1 3		
P5	0	0	2	5	3	3	0 7	0 4	2 5		
	7	2	5								

P~~1 → P2 →~~ P2 → P4 → P5

Exe-01

Deadlock Handling Method

(1) Ignore (2) prevention (3) Avoidance (Banker Algorithm is used to do this)
(4) Detection & Recovery.

Ex:-01

(Banker's Algorithm)

MN-Allo

Process	Allocation			Maximum Need			Available			Remaining Need			Total Resources
	A	B	C	A	B	C	A	B	C	A	B	C	
P1	1	0	1	4	3	1	3 1	3 0	0 3	3	3	0	A=8
P2	1	1	2	2	1	4	4 1	3 0	3 1	1	0	2	B=4
P3	1	0	3	1	3	[illegible]	5 1	3 1	4 2	0	3	0	C=6
P4	2	0	0	5	4	1	6 2	4 0	6 0	3	4	1	
	5	1	6				8	4	6				

8-5=$\boxed{3}$
4-1=3
6-6=0

Safe sequence:-

$P_3 \rightarrow P_1 \rightarrow P_2 \rightarrow P_4$

There are 6 processes & 4 Resources. Now verify this with Banker's Algorithm.

Process	Current Allocation				Maximum Need/Demand				Need Available Matrix (Remaining)				Available Need Matrix				Total Resources
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	
P_0	2	0	2	1	9	5	5	5	6 0	3 1	5 1	4 1	7	5	3	4	A=15
P_1	0	1	1	1	2	2	3	3	6 1	4 0	6 0	5 1	2	1	2	2	B=6
P_2	4	1	0	2	7	5	4	4	7 1	4 1	6 0	6 0	3	4	4	2	C=9
P_3	1	0	0	1	3	3	3	2	8 4	5 1	6 0	6 2	2	3	3	1	D=10
P_4	1	1	0	0	5	2	2	1	12 2	6 0	6 2	8 1	4	1	2	1	
P_5	1	0	1	1	4	4	4	4	14 1	6 0	8 1	9 1	3	4	3	3	
P_6	9	3	4	6					15	6	9	10					

Safe sequence:-

~~$P_1 \rightarrow P_3 \rightarrow P_4 \rightarrow P_2 \rightarrow P_1 \rightarrow P_5$~~

$P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_0 \rightarrow P_4 \rightarrow P_5$

Memory Management

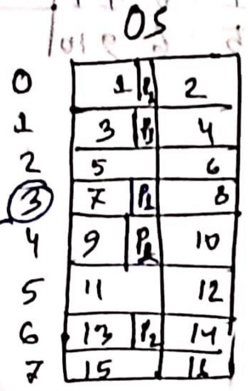
Virtual Memory:- It is a memory management technique where secondary memory can be used as if it were a part of main memory.

Importance:-

1| It provides illusion to the programmers that a process who sizes larger than the size of MM can also be executed.

2| More & more process will be able to come to the MM.

Working procedure

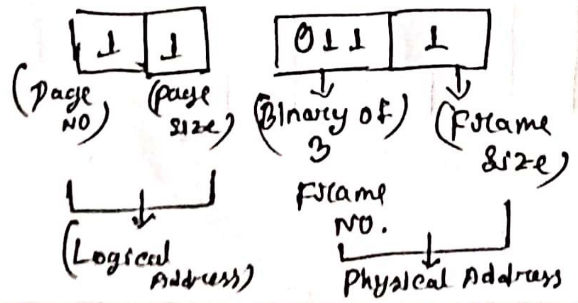


Suppose,
Memory Size = 16 Byte
Frame Size = 2 Byte
Total no. of frame required = $\frac{16}{2}$ = 8 Frame
Process size = 4 Bytes
Page size = 2 Bytes
No. of pages = $\frac{4}{2}$ = 2 Bytes

Questions:-

(1) 4 নং এর Byte কত?

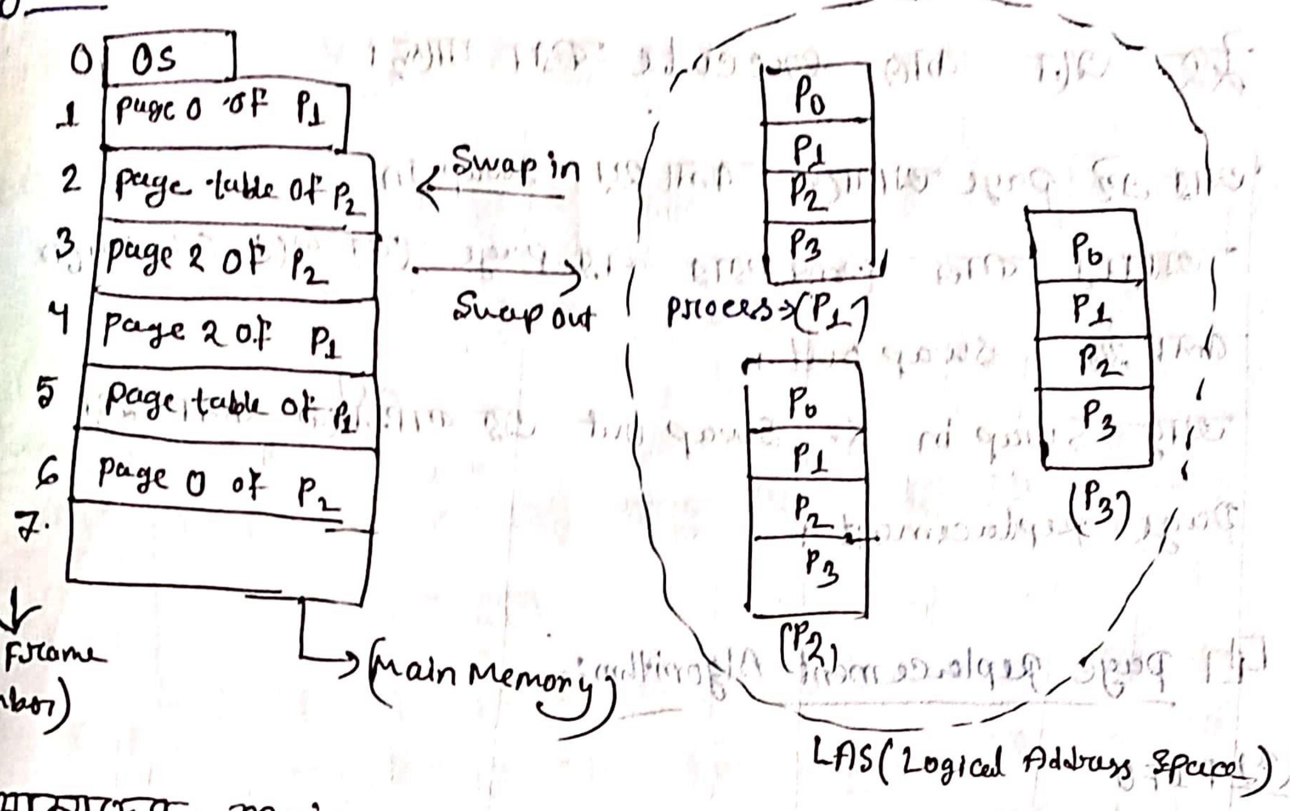
=>



=> page size & Frame size same

=> process-1, (3) নং Frame এ আছে এর binary (011).

Diagram:



=> আমাদের main Memory তে size limited, কিন্তু process এর size গুলো limited এবং এই size প্রতিনিয়তই বেড়ে চলেছে। আর virtual memory সেই illusion দেয় যার মাধ্যমে main memory এর চেয়েও বড় process সেই CPU-ই execute করে।

Mainly, process গুলো LAS (Hard Disk) এ থাকে। আর সেখানে multiple process গুলোকে page এ divide করা থাকে। Page এর size গুলো frame size এর সমান হয়ে থাকে। আর আমরা পুরো process কে Main Memory তে না নিয়ে বরং Required process অংশকেই Main Memory তে নিতে হয়। যেমন,

Frame-1 এ process P_1 এর page 0 কে Load করা হয়েছে,

Frame-2 তে process P_2 এর page table load করা হয়েছে

এভাবেই Importance এর উপর নির্ভর করে Main Memory তে আনা হয়েছে।

আর এর ফলে full process না এনেও important part টুকু এনে কাজ execute করা যাচ্ছে।

আর এই page আনাকে বলা হয় swap in

আবার কাজ শেষ করার পর page বের করে দেওয়াকে বলা হয় swap out ।

আর Swap in & swap out এর কাহিনীকে বলা হয় Page Replacement ।

Page Replacement Algorithm:-

(a) FIFO

(b) LRU [Last Recently Used]

(c) Optimal

Question:-

Reference String:- 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6

Frame Size:- 3 Frame / ~~4 Frames~~

Solve this Using FIFO, LRU, Optimal

FIFO:- (যে আগে আসবে সে যাব)
3 Frames

F_1	1	1	1	4	4	4	4	6	6	6	6	3	3	3	3	2	2	2	2	6
F_2		2	2	2	2	1	1	1	2	2	2	2	7	7	7	~~8~~ 7	1	1	1	
F_3			3	3	3	3	5	5	5	1	1	1	1	6	6	6	6	6	3	
	M	M	M	M	H	M	M	M	M	M	H	M	M	M	H	M	M	H	M	

~~Total~~ page Hit = 4

$\therefore$ Hit Ratio = $\frac{4}{20} \times 100 = 20\%$

~~Total~~ page Miss = 20-4 = 16

$\therefore$ Miss Ratio = $\frac{16}{20} \times 100 = 80\%$.

(Ans)

~~[illegible]~~

নিয়ম: → FIFO তে সবার আগে যে আসবে তাকে change করতে হবে

→ যদি Hit হয়, তবে Count করার সময় যে সংখ্যা Hit হবে তাকে বাদ দিয়ে Count করতে হবে। এবং ঐ অনুসারে যে আগে আসবে তাকে বাদ দেওয়া হবে।

LRU [Least Recently used]

=> এখানে পুরাতনটাকে Replace করে নতুনটাকে আনা হবে।

8 ⓔ Reference String:- 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6

Frame size:- 3 Frames.

F1	1	1	1	4	4	4	5	5	5	1	1	1	7	7	7	2	2	2	2	2
F2		2	2	2	2	12	2	6	6	6	6	3	3	3	3	3	3	3	3	3
F3			3	3	3	31	1	1	2	2	2	2	2	6	6	6	1	1	1	6
	M	M	M	M	H	M	M	M	M	M	H	M	M	M	H	M	M	H	H	M

Page Hit = 5 $\therefore$ Hit ratio = $\frac{5}{20} \times 100\% = 25\%$.

page Miss = 20 − 5 = 15

∴ Miss Ratio = $\left(\frac{15}{20}\right)^{*}$ 100% = 75%

Optimal

নিয়ম:-

- → যেটা এবার আগে আসবে তাকে সরাতে হবে
- → যদি process এ থাকা সংখ্যা ও নতুন সংখ্যা same হয় তবে কোনো কিছুই পরিবর্তন হবে না। Hit হবে।

Optimal

(যেটা অনেক দূরে string সেটাকে replace করতে হবে)

⇒ Replace

Reference String:- 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6

Frame Size:- 3

F1	1	1	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3
F2		2	2	2	2	2	2	2	2	2	2	2	7	7	7	2	2	2	2	4
F3			3	4	4	4	5	6	6	6	6	6	6	6	6	6	1	1	1	6
	M	M	M	M	H	H	M	M	H	H	H	M	M	H	H	M	M	H	H	

page Hit = 8

∴ Hit Ratio = $\frac{8}{20} * 100 = 40\%$

page miss = 20−8 = 12

∴ Miss Ratio = $\frac{12}{20} \times 100 = 60\%$

FILE SYSTEM

1) File Attributes & File operations:-

File Attributes:-

1) Name → Name of the File

2) Extension Type :- কি extension দিয়ে save হবে।

3) Identifier:- চিনার মত এমন কিছু। যেমন University ID.

4) Location → where will it be saved.

5) Size → File ~~s~~ size

6) Modified date, created date → কবে তৈরি এবং Last কবে modified হয়েছে।

7) protection permission → কে Access করতে পারবে।

8) Encryption & compression → Encrypt করা ও extract করা।

9) File Attributes actually take 'system reserved' space to store metadata

Operation:-

i) Create

ii) Reading

iii) writing

iv) Deleting

v) Closing

vi) Copying

vii) Repositioning (যখন data কে point out করে Data escape করতে চাইলে এর use হয়)

(d) File Attributes:-

(1) protection -> who can access

(2) password -> Needed to access file

(3) Creator -> ID of the person who created

(4) Owner -> Current owner.

Flag:-

(5) Read only flag -> 0 for read/ write, 1 for read only.

(6) Hidden flag -> 0 for normal, 1 for do not display the listing.

(7) System flag -> 0 for normal, 1 for system file.

(8) Archive flag -> 0 has been backed up, 1 for needs to be backed up.

(9) Random Access flag -> 0 for sequential access only, 1 for needs to be backed up.

(10) Lock flag -> 0 for unlock, 1 for lock.

(*) provide info required to find the keys.

(11) Record length -> Number of bytes in a record

(12) Key position -> Offset of the key within each record

(13) Key length -> Number of bytes in key field.

(*) When it was created, was accessed & modified:-

(14) Creation time:- Date & time the file was created.

(15) b. Last Access -> Date & time the file was accessed

(16) Time of last change -> Date & time the file was changed

Based on Size

(17) current size → Number of bytes in the file

(18) Maximum size → Number of bytes the file may grow to.

File operation

(1) Create → The file is coming & to set some attributes

(2) Delete → Free up disk space

(3) Open → Allow system to fetch the attributes

(4) Close → No longer needed then close it.

(5) Read → Bytes come from current position.

(6) Write → Datas are written to the file.

(7) Append → Only add data to the end of the file.

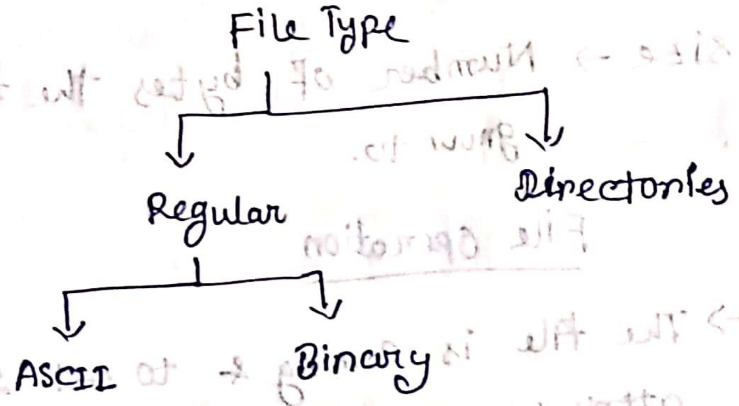
(8) Seek → Seek repositions the file pointer to a specific place in the file.

(9) Get attributes → Need to read the file attributes to do their work.

(10) Set attributes → attributes can be changed or setted after file creation.

(11) Rename:- change the name of the existing files

File Types & File Access:-



① Regular File:-

→ Contain user information

→ Has no other predefined internal structure as a randomly accessible sequence of bytes.

→ Application programs are responsible for understanding the structure.

② Directories:-

→ Maintain the structure of the file system.

→ To keep track of files, file systems normally have directories or folder.

ASCII File:-

→ Consist of line of text.

→ Can be displayed, printed, edited.

→ Easy to connect the output of one program to the input of another.

→ C/C++/perl/HTML files, all are ASCII files.

Binary Files:-

- Formatted Information that only specific applications & processors can understand.
- Must run on appropriate software or processor.
- Ex:- Executable files, Compiled programs, Spreadsheets, Compressed files, graphic files etc

Device files:-

- In Linux & unix every hardware device is treated as a file.
- A device file ~~he~~ is an interface for a driver that appears in a file system as if it were an ordinary file.
- This allows software to interact with device driver using standard input/output system calls, which simplifies many tasks.

Character Special Files:-

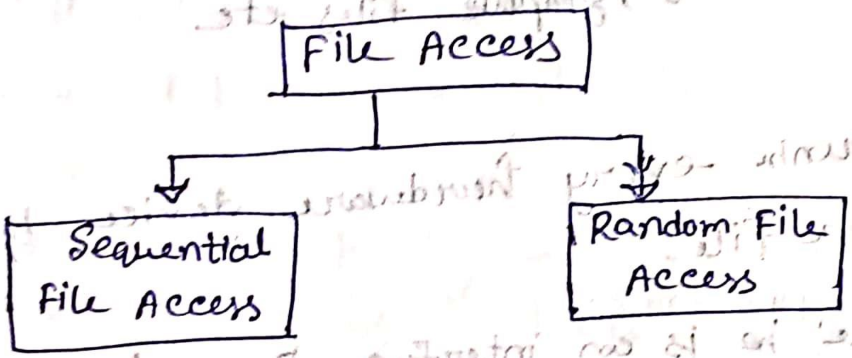
- Device files which talks to devices in a characters by character (1 byte at a time)
- This special files are related to input/output & use to model serial Input/output devices, such as terminals, printers & networks.

Block Special Files :-

→ Talks to devices 1 block at a time (1 block = 512 bytes to 32 KB).

→ Block special files are used to model disks, CD/DVD ROM, memory regions etc.

- - - - -



Sequential File Access :-

→ process could read all the bytes or records from a file in order.
Starting from the beginning till the end & can't ~~stop~~ skip in between them.

→ Could be read as often as needed.

→ Convenient when storage medium was magnetic tape or CD-ROM.

Random File Access :-

→ Files whose bytes or records can be read in any order are called Random File Access.

→ ~~Exe~~ Essential for many applications. For Example :- Data base system.

⇒ If an airline customer calls up & wants to researve a seat on a particular flight, the reservation program must access the record for that flight without reading thousands of other flights records.

Directory structure

⇒ To keep track all files file system Normally have directory.

⇒ And directories are system files for maintaining the structure of the file system.

(1) Single level Directory system:-

→ One directory keeps all the files.

→ Easy to find files

→ Simple & quick

→ Used in telephones.

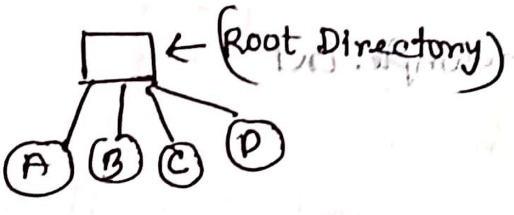


Fig:- Single-level Directory system.

(iii) Hierarchical Directory System:-

→ For so many files this is needed.

→ It structures & maintains an organized way.

→ User can create arbitrary number of subdirectory

→ Specifying their names.

Two methods are used:-

(i) An Absolute path name:-

→ The paths will be having entire directory structure from the root ~~directly~~. directory.

Ex:- C:\users\username\Sakur\Document\Sakur.txt

(ii) Relative path name:-

→ Only the current working directory will be there. The file "example.txt" is located within the 'Documents' ~~directly~~ directory, which is a subdirectory of current working directory.

ex: Documents/example.txt

Fig.

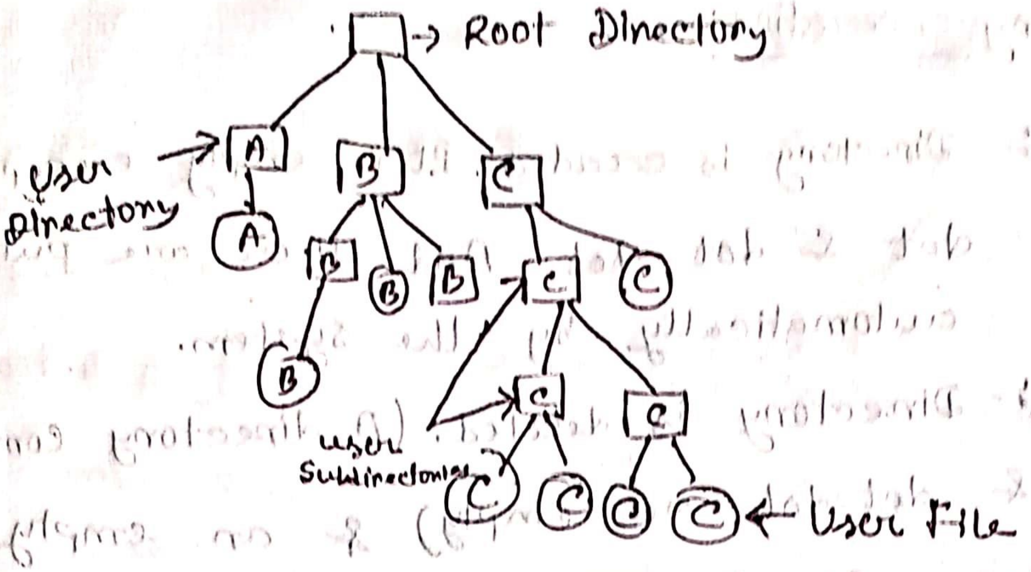


Fig:- A Hierarchical Directory system

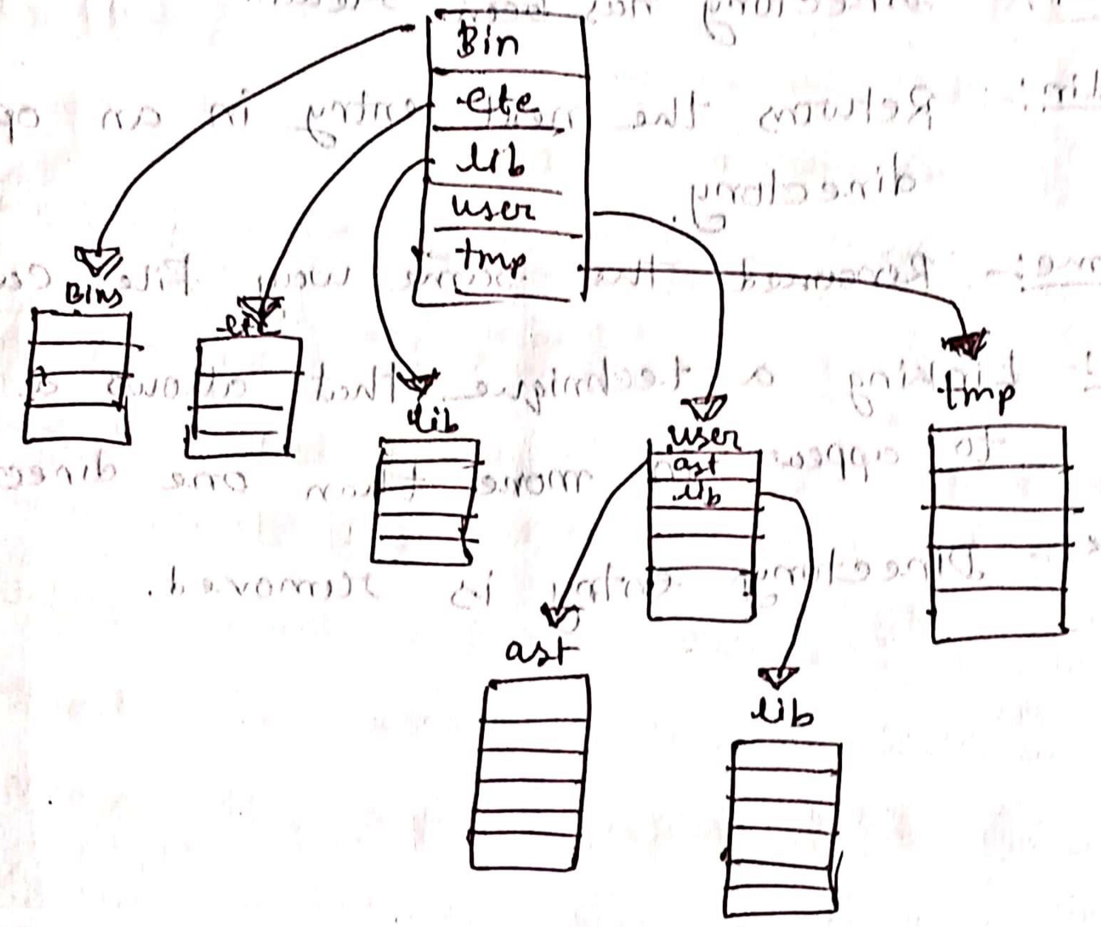


Fig:- A unix directory tree.

7] Directory operation:-

→ Create:- Directory is created. It is empty except for dot & dot dot. And these are put there automatically by the system.

→ Delete:- Directory is deleted. (A directory containing dot & dot dot are empty) & an empty directory can be deleted.

→ Opendir:- Directories can be read

→ Closedir:- Directory has been read.

→ Readdir:- Returns the next entry in an open directory.

→ Rename:- Renamed the same way files can be.

→ Link:- Linking a technique that allows a file to appear in more than one directory

→ Unlink:- Directory entry is removed.

Security of File System :-

principles of Security :-

(i) principles of least privileges:-

→ privilege means giving permission

→ This principles is about how privileges are granted.

→ A subject is given only those privileges that is required for completing task.

→ If no specific rights ~~to an object~~ is not granted.

→ only to append, not to rewrite.

→ once done take it from them.

(ii) principles of fail safe defaults:-

⇒ when subject, object create হয় তখন privilege ~~[illegible]~~, ~~ফি~~ ফিরত নিতে হয়।

⇒ Unless subject is given explicit to the object, it should be denied access to that object.

⇒ Means, the default access to object is none.

⇒ All the privileges are ~~go~~ authorized to trusted.

(iii) principles of economy mechanism :- (যত simple তত ভাল)

=) Simplifies the design & implementation of security mechanism.

=) Security mechanism should be as simple as possible

=) Fewer chances of errors

=) The checking & testing procedure becomes simpler.

(iv) principles of Complete Mediation :- (Eligible কিনা)

-) Checks if object is eligible to get the access

-) If yes, then it helps with resources.

-) If the subject reattempts to read operation then it checks if the subject is still allowed to read the object & then allows for reading.

(v) principle of open Design :-

-> This principles suggests that complexity doesn't add security.

-> This ~~security~~ principle states that the security of mechanism should not design on its design.

(vi) Seperation of privileges:-

-> Access of an object should not depend only on fulfilling a single condition.

-> Should be multiple conditions required & two or more system components work together to enforce security.

(vii) principles of least common Mechanism:-

=> Common Mechanism of multiple user should be kept minimum.

(viii) principles of user Acceptability:-

=> ~~what~~ According to this principle whatever the protection is used here should be kept as simple as possible.

=> Otherwise user might feel burden.

Domain protection Mechanisms:-

=) A Computer is a collection of processes & they all should be protected.

⇒ Each object have name & have different set of operations.

=) Unauthorized processes should be prohibited from access.

=) process should be able to access only those resources that it currently requires to complete its task.

=) This requirement is known as need to know principles.

=) operations that are possible depend on the object:-

(i) CPU:- Execution

(ii) File:- Read, write

(iii) Semaphore:- UP, Down

(iv) Tape Drives:- Read, write, Rewound.

Domain Structure

=) Set of access rights.

→ A domain is defined as a set of ~~<ot~~ <object, {access right set}> pairs.

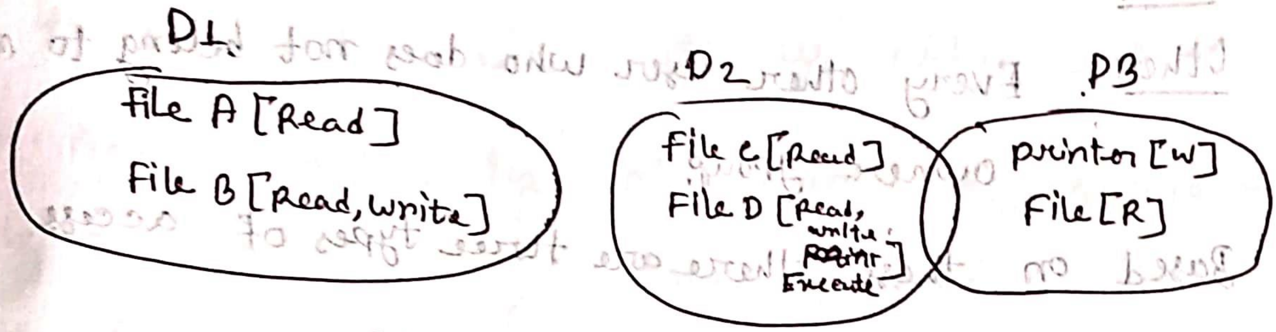


Fig:- Three protection Domains.

We can also call Domain as user

Object / Domain	File-A	File-B	File c	File-D	Printer
1	Read	Read write			
2			Read	Read write Execute	Write
3					write

(Access Matrix)

Access Control List

In OS like Linux, the file system gives three types of permission for a resource

User:- The user

Group:- To the group which the user belong

Other:- Every other user who does not belong to the owner group.

Based on these there are three types of access:

- Read
- Write
- Execute

For example:-

Here permission :- [illegible]--

→ rwxr-xr--

According to this,

user:- can read, write & execute the file (rwx)

group:- can read & execute (r-x)

other:- can only read (r--)

→ Now, we do not want all other users to read the file. So, we can change the file permission to this:- rwxr-x---

User:- can read, write & execute the file (rwx)

Group:- can read & execute (r-x)

Others:- can't do anything.

Suppose a new member arrived In the group. And he will be given permission only for some ~~some~~ files to read from directory. So, he can't be put in the owner group. If we do that the would be able to read all the files.

So, to solve this we will use ACL:-

ACL:- Access Control list, of permissions associated with a resource in a file system.

→ To see all the permissions associated with resource on linux,

command:-

```
→ $ getfacl <Filename>
```

Output:-

```
#File: <Filename>
#owner:. John
# group:- Sales
user:- rwx
group: r-x
other: ---
```

⊛ Setting permission for new member:-

```
$ ~~getfacl <Filename>~~
$ setfacl -m u: bob :r-- <Filename>
$ getfacl <Filename>
```

Output:-

```
# Owner: John
# group :- Sales
user: -rwx

user: bob: r--
group:: r-x
mask: rwx

other: ---
```

And once the user leave & then we can reset the permission again:-

```
$ setfacl -x user: bob <Filename>
$getfacl <Filename>
```

output:-

```
#owner: John
#group:- sales
user: rwx
group:- r-x
Other:.:- ---
```

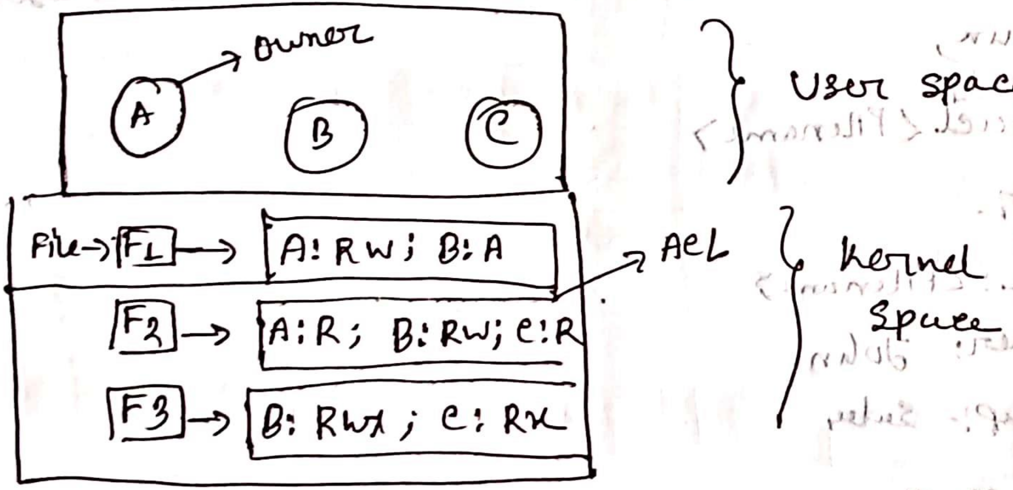


Fig:- Use of Access Control lists to manage Access File

Aut-22

S-2(b) Internal & ~~Extl~~ External fragmentation:-

Internal:- when memory is allocated to a process, but the allocated memory is not fully utilized by the process.

→ It arises when the allocated memory block is larger than what the process is ~~ne~~ actually needs

→ occurs in a single memory block

→ Reduces overall system efficiency & effective memory utilization.

→ Dynamic partitioning with compaction / paging helps to mitigate

→ Ex:- process needs 50 KB of memory, and it is allocated a memory block of 64 KB. There is 14 KB of internal fragmentation.

External Fragmentation:-

→ Free memory blocks are scattered throughout the system, making it challenging to allocate contiguous memory blocks to a process.

→ occurs due to allocation & deallocation of memory.

→ It affects entire memory space & is not confined to a specific memory block.

-) Reduces the available free memory for new processes.

-) Techniques like compaction or paging can help external fragmentation.

Ex:- Three free memory blocks of sizes 20KB, 15KB, & 25KB with allocated blocks in between.

Even if the total ~~of~~ free memory is sufficient it might be challenging to allocate a 30KB process due to fragmentation.

Q(a) Aut - 22

□ Physical & Logical Address:-

Logical Address:-

-) Known as virtual Address, generated by CPU during the execution.

-) Represents the location of data.

-) Generated by ~~data~~. CPU.

-) These are visible.

-) Access an array element is the use of logical address.

Physical Address:-

-) Actual ~~of~~ location in RAM

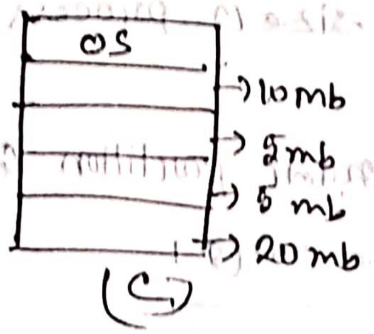
-) Represent actual location of data or an instruction in the physical memory of the computer.

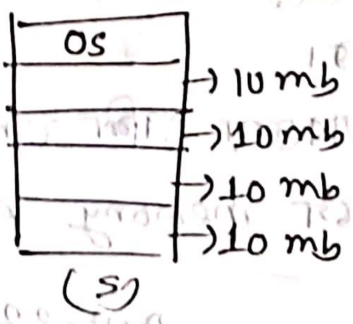
-) This is not visible. CPU & Memory Management Unit translates logical address to physical address.

Contiguous:- Serially Allocate করাতে হবে

Non-contiguous:- Separate blocks of memory to a process.

Contiguous Fixed / Static Memory partitioning:-

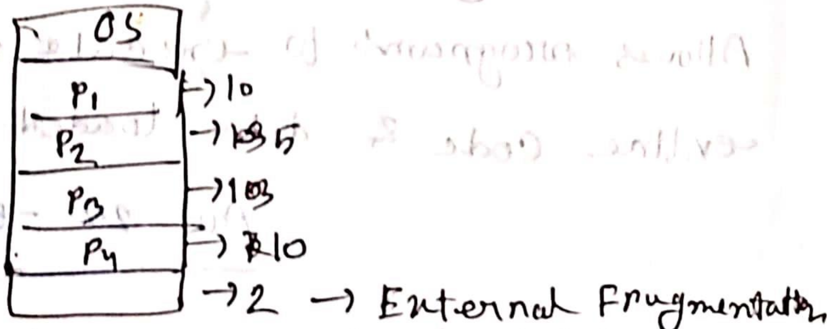




variable sized partitioning

Contiglous Memory Allocation এর একটি part। It is used to overcome the problem faced by Fixed partitioning. process এর উপর depend করে partition হয়।

P1 -> 10mb
P2 -> 5mb
P3 -> 13mb
P4 -> 10mb
P5 -> 7mb



Advantage:-

-> NO Internal Fragmentation
-> NO restriction on degree
-> NO limitation

Disadvantage:-

-> Difficult to Implemant
-> External Fragmentation

এ variable size partitioning Algorithm:

→ First Fit:- প্রথম থেকে search করা শুরু করে। প্রথম যে জায়গা খালি থাকে সেখানেই process Allocate হয়।

Next-Fit:- Last যে process Allocate হয়েছে তারপর হতে ~~Allocate~~ memory search

Best-Fit:- সবচেয়ে কম partition size এ process Allocate করে।

Worst-Fit:- সবচেয়ে বেশি জায়গাওয়ালা partition কে খুঁজে এবং তাতে আগে memory Allocate করে।

Aut-22 - 3(a)

Demand paging:- A memory management scheme used in OS. to optimize the use of physical ~~address~~ memory. by loading only the necessary portion of a program into memory when they are needed.

Allows programs to execute without having their entire code & data loaded into RAM.

Aut-22 - 5(a)

Grep → used to search pattern in a File

Cat → used to display the contents of Files

Cmp → Compare two files byte by byte

chmod → Change the permissions (mode) of a File or directory.

Spring-22