Shri Shankaracharya Institute of Professional Management & Technology, Raipur

---

February-2022- Class Test-2

Date: 09/02/2022

Student Name: V OM SAI NAGESHWAR SHARMA

Roll No.:

3	0	3	3	0	2	2	2	0	0	2	0

Enrollment No.:

B	J	4	5	9	9

Course: B.Tech **Semester:** 3rd

Branch: Computer Science And Engineering

Subject Name: Operating System

Subject Code:

B	0	2	2	3	1	5	(	0	2	2	)

Mobile No.: 8602727389

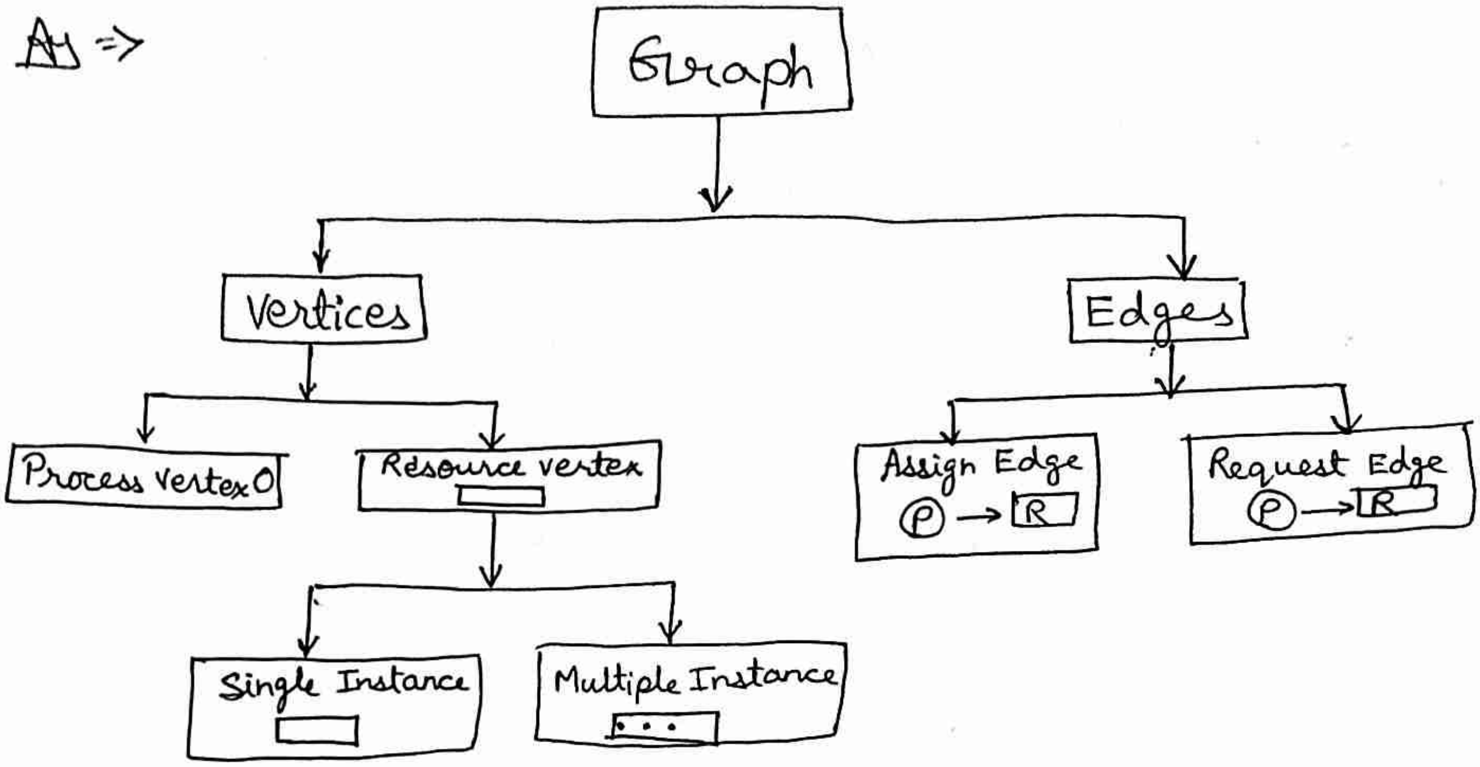
Email id: om.sharma@ssipmt.com

Signature

PART - I

(A)
Q1 >

Ans =>



The Resource Allocation Graph, also known as RAG is a graphical representation of the state of a system. It has all the information about the resource allocation to each process and the request of each process.

Q(b)>

Ay=> Thrashing: Thrashing is when the page fault and swapping happens very frequently at a higher rate, and then the operating system has to spend more time swapping these pages. This state in the operating system is known as thrashing. Because of thrashing, the CPU utilization is going to be reduced or negligible.

Three main causes of Thrashing are as follows :-

1. High degree of Multiprogramming.
2. Less number of frames compared to the process requirement.
3. The process scheduling scheme which swaps in more processes when CPU utilization is low.

Q(C) >

Ans => **Translation Look-aside Buffer:**

It is a memory ~~cha~~ cache that is used to reduce the time taken to access a user memory location. It is a part of the chip's memory - management unit (MMU). The TLB stores the recent translations of virtual memory to physical memory and can be called an address - translation cache.

Q(D) >

Ans => A file is a collection of released information defined by its creator.

Commonly Files represent program & data. Data files may be numeric alphabetic or alphanumeric.

A files is a sequence of bits, bytes, lines or records whose meaning is defined by its creator & user.

The various attributes of a file :-

1. Name.
2. Identifier.
3. Type.
4. Location.
5. Size.
6. Protection.
7. Time, date & user Identification.

Part - II

Q(B) >

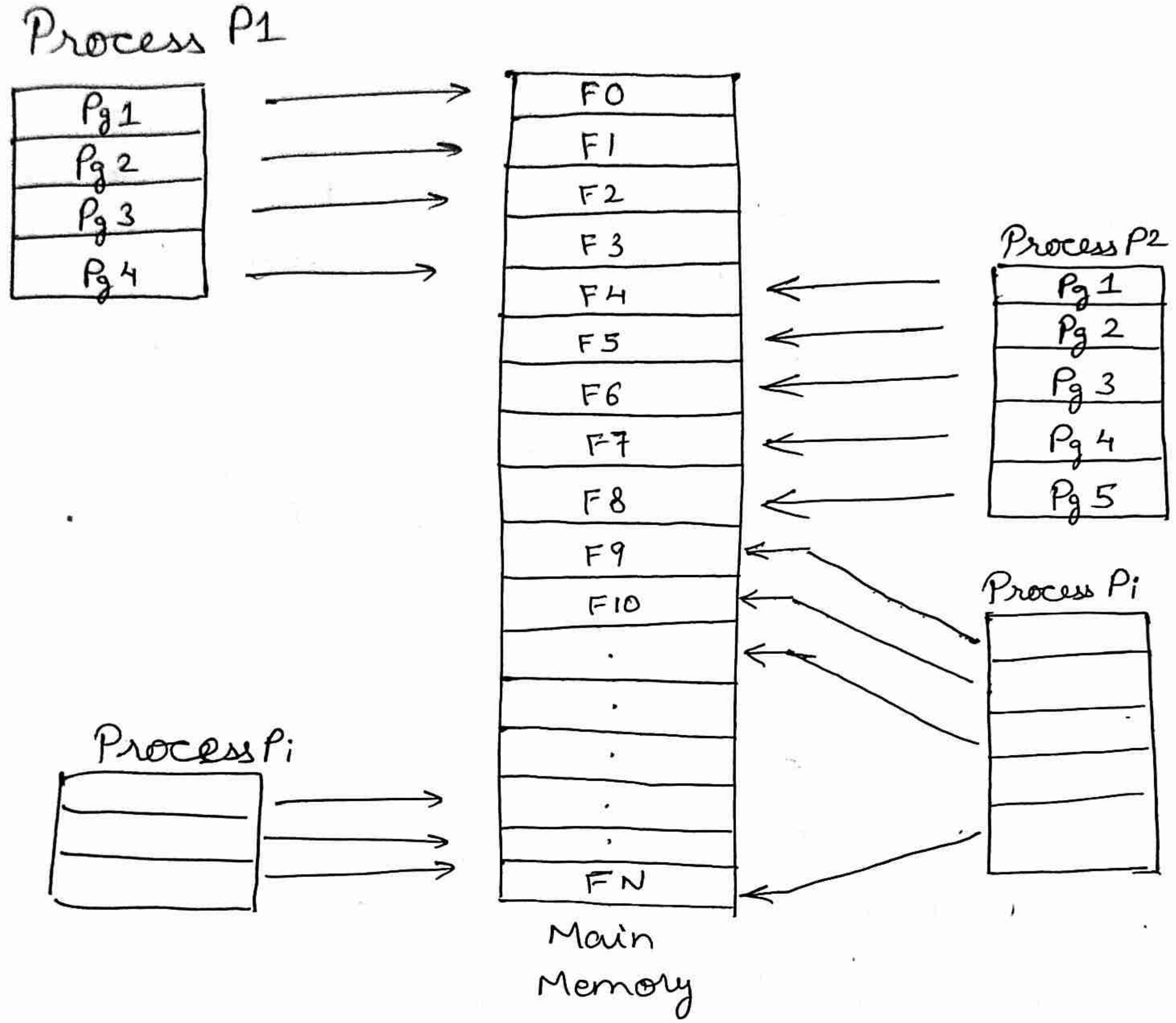
Ans => The physical memory is conceptually divided into a number of fixed size blocks, called frames.

The logical address space is also splitted into fixed size blocks called, pages.

when a process is to be executed, its pages are loaded into any available memory frame from the backing store.

The backing store is also divided into fixed size blocks that are of same size of the frames.

i.e. size of frame = size of pages for a particular hardware.



Allocation of memory consist of finding a sufficient number of unused page frames for loading the pages of a requesting process for its execution.

Since, each page is mapped seperately different page frames allocated to a single process need not occupy contigeous areas of Main memory.

Paging is a fixed size partitioning scheme

In paging, secondary memory & main memory are divided into equal fixed size partitions.

The partitions of secondary memory are called "pages".

The partitions of main memory are called as "frames".

Each process is divided into parts where size of each part is same as page size.

The pages of process are stored in the frames of main memory depending upon availability.

....

Q(c) >

Ans => Belady's anomaly: It states that it is possible to have more page faults when increasing the number of page frames while using FIFO ~~and~~ method of frame management. Laszlo Behady demonstrated ~~that~~ this in 1969. Previously, it was believed that an increase in the number of page frames would always provide the same number or fewer page faults.

For optimal

~~1 0 2 2 1 7 6 7 0 1 2 0 3 0 4~~

1	0	2	2	1	7	6	7	0	1	2	0	3	0	4	5	1	5	2	4	5	6	7	6	7	2	4	2	7	3	3	2	2
1	1	1	1	1	1	6	6	6	6	2	2	3	3	3	5	5	5	5	5	5	6	6	6	6	6	4	4	4	3	3	3	3
	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	4	4	4	4	4	4	7	7	7	7	7	7	7	7	7	7	7
		2	2	2	7	7	7	7	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
F	F	F	X	X	F	X	X	X	F	F	X	F	F	F	F	X	X	F	X	X	F	F	X	X	X	F	X	X	F	X	X	X

Fault = 15

For LRU

0	2	2	1	7	6	7	0	1	2	0	3	0	4	5	1	5	2	4	5	6	7	6	7	2	4	2	7	3	3	2	3
1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	4	4	4	7	7	7	7	7	7	7	7	7	7	7
	0	0	0	0	7	7	7	7	2	2	2	4	4	4	4	4	2	2	2	6	6	6	6	6	4	4	4	3	3	3	3
		2	2	2	2	6	6	1	1	1	3	3	5	5	5	5	5	5	5	5	5	5	5	2	2	2	2	2	2	2	2
F	F	F	x	x	F	F	x	F	F	F	x	F	x	F	F	F	x	F	F	x	F	F	x	x	F	F	x	x	F	x	x

Fault = 19

Fault Rate = $\frac{19}{33}$

∴ optimal is better ~~for~~ for all.

F.IFO

1 0 2 2 1 7 6 7 0 1 2 0 3 0 4 5 1 5 2 4 5 6 7 2 4 2 7 3 3 2 3

1 1 1 1 1 7 7 7 7 1 1 1 1 0 0 0 0 1 1 1 1 5 5 5 ~~5~~2 2 2 2 2 2 2 2

0 0 0 0 0 6 6 6 6 2 2 2 2 4 4 4 4 2 2 2 6 6 6 4 4 4 4 4 4 4

2 2 2 2 2 0 0 0 3 3 3 3 5 5 5 5 4 4 ~~7~~4 7 7 7 7 7 3 3 3 3

F F F x x F F x F F x F F F F F x F F F F x x F F x x F x x ~~F~~x

No. of page faults = 21

Fault ratio - $\frac{21}{33}$

Q3(A)

Ans =>

Process	Allocation				Maximum				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	2	0	2	1	9	5	5	5	6	3	5	4
P1	0	1	1	1	2	2	3	3				
P2	4	1	0	2	7	5	4	4				
P3	1	0	0	1	3	3	3	2				
P4	1	1	0	0	5	2	2	1				
P5	1	0	1	1	4	4	4	4				
	9	3	4	6								

Remaining Need (Max - Allocation)

	A	B	C	D
P0	7	5	3	4
P1	2	1	2	2
P2	3	4	4	2
P3	2	3	3	1
P4	4	1	2	1
P5	3	4	3	3

Safe Sequence

$i = 0$

P_0

If Need $\leq$ Available

7534 $\leq$ 6354

False

$i = 1$

P_1

Need $\leq$ Available

2122 $\leq$ 6354 (True)

True

Avail = Avail + Allocation

= 6354 + 0111

= 6465

$i = 2$

P_2

Need $\leq$ Available

3442 $\leq$ 6465 (True)

True

Avail = Avail + Allocation

= 6465 + 4102

= 10567

$i = 3$

P_3

Need $\leq$ Available

2331 $\leq$ 10567 (True)

(True)

Avail = Avail + Allocation

= 10567 + 1001 = 11565

$i = 4$

P_4 Need $\leq$ Available (True)

4121 $\leq$ 11568

Avail = Avail + allocation

= 11568 + 1100

= 12668

$i = 5$

P_5

Need $\leq$ Available (True)

3433 $\leq$ 12668 (True)

Avail = 12668 + 1011

= 13679

$i = 6$

P_0

Need ≤ Available (True)

$1534 \leq 13679$

Avail = Avail + allocation

$= 13679 + 2021$

$= 156910$

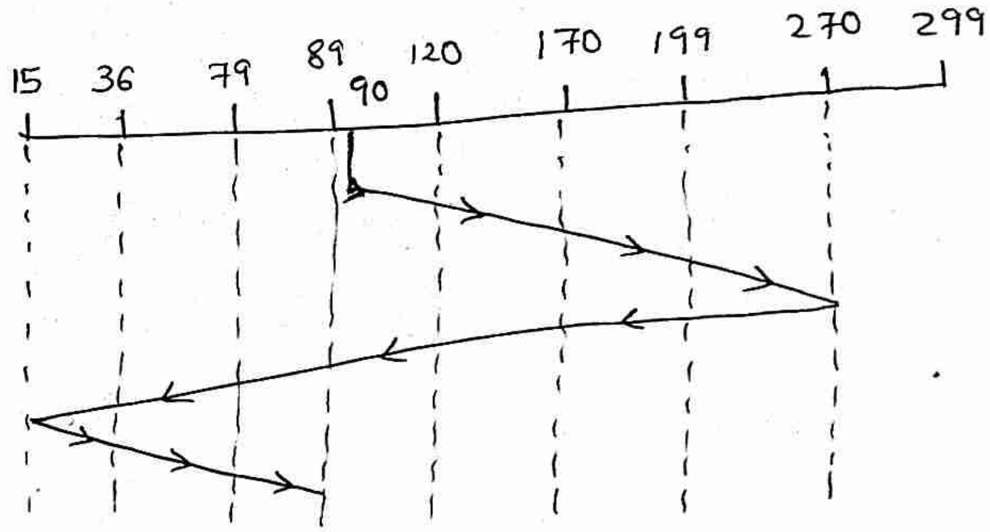
Safe Sequence

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

~~Safe~~ A state is safe if the system can allocate resources to each process (up to its maximum requirement) in some order and still avoid a deadlock. Formally, a system is in a safe state only, if there exist a safe sequence.

Q5(D)>

Ans => (i) SSTF :

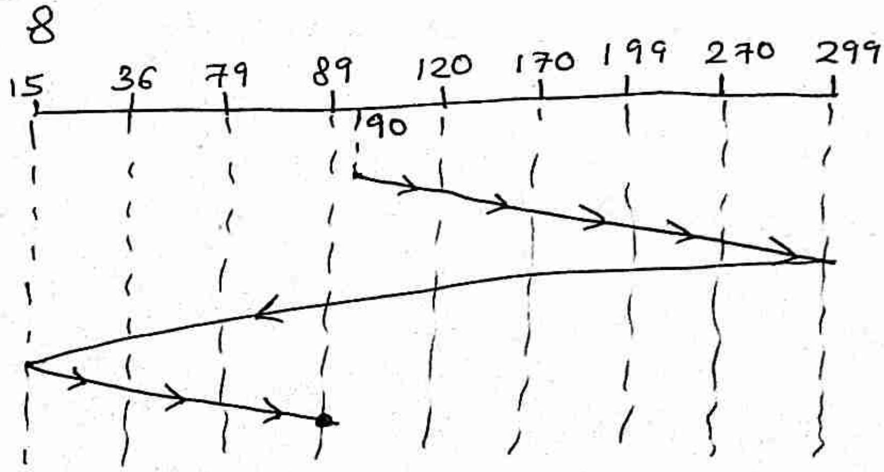


Average ~~cylinder~~ Head movements will be

$$= \frac{|90-120| + |120-170| + |170-190| + |190-270| + |270-89| + |89-79| + |79-36| + |36-15|}{8}$$

$$= \frac{30+50+29+71+181+10+43+21}{8}$$

$$= \frac{435}{8} = 54.37$$ ~~cylinders~~. Head.



Average ~~cylinder~~ Head movements will be

$$= \frac{|90-120| + |120-170| + |170-199| + |199-270| + |270-15| + |15-36| + |36-79| + |79-89|}{8}$$

$$= \frac{30 + 50 + 29 + 71 + 255 + 43 + 10}{8}$$

$$= \frac{509}{8}$$

$= 63.62$ ~~cylinders~~ Head