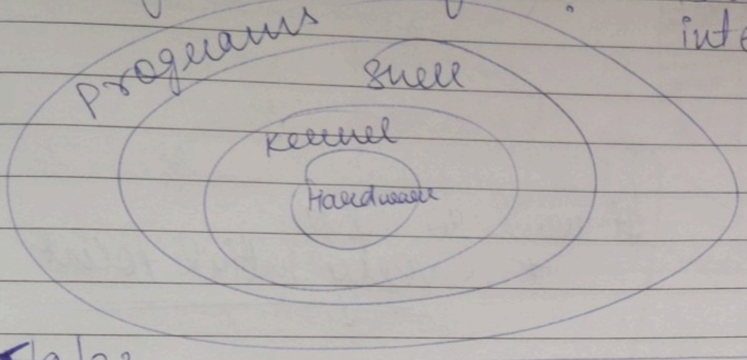
Date 1/9/22

Page

Why CLI is fast? = Bcoz no graphical intervention. (of)



IPC - Inter Process Communica^n

5/9/22

* NUMA - Non uniform memory access.
(अपने Processor को छोड़ के दूसरे के पास जाए)

Actual CPU Burst time (of previous process)

$$T_{n+1} = \alpha t_n + (1-\alpha) T_n$$ → Predict of prev. process

Predicted; smoothing factor or weighing "

$0 < \alpha < 1$

Exponential Averaging.

$$T_n = \alpha t_{n-1} + (1-\alpha) T_{n-1}$$

weightage; Predicted at $(n-1)^{th}$ time

8/9/22

$$T_{n+1} = \alpha t_n + (1-\alpha)\left[\alpha t_{n-1} + (1-\alpha) T_{n-1}\right]$$

Predict

$$T_{n+1} = \alpha t_n + \alpha t_{n-1} + T_{n-1}$$ (struck out)

$$T_{n+1} = \alpha t_n + (1-\alpha)\alpha t_{n-1} + (1-\alpha)^2 T_{n-1}$$

$$T_{n-1} = \alpha t_{n-2} + (1-\alpha) T_{n-2}$$

FREEMIND

Date

Page

$$T_{n+1} = \alpha t_n + \alpha(1-\alpha)t_{n-1} + \alpha(1-\alpha)^2 t_{n-2} + (1-\alpha)^3 T_{n-2}$$

Predicted

So on

It will go upto very initial point

Previous Processes

Q → $\alpha = 0.5$ $T_1 = 10$ or $\tau_1 = 10$

Actual Burst Time t_1 t_2 t_3 t_4
4 8 6 7

$T_5 = ?$

→ $T_2 = \alpha t_1 + (1-\alpha) T_1$

$T_2 = 0.5 \times 4 + (0.5) \times 10 = 7$

$T_3 = \alpha t_2 + (1-\alpha) T_2$

$= 0.5 \times 8 + 0.5 \times 7 = 4 + 3.5 = 7.5$

$T_4 = \alpha t_3 + (1-\alpha) T_3$

$= 0.5 \times 6 + 0.5 \times 7.5$

$= 3 + 3.5 = 6.5$ 6.75

$T_5 = \alpha t_4 + (1-\alpha) T_4$

$= 0.5 \times 7 + 0.5 \times 6.75$

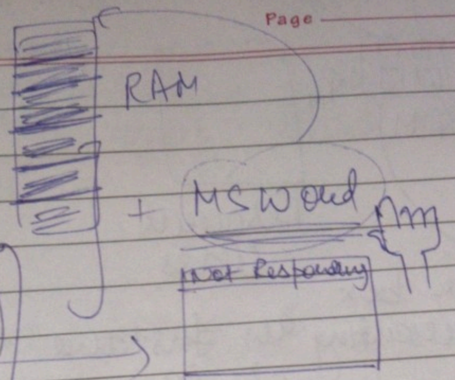
$= 3.5 + 3.385$

$= 6.885$

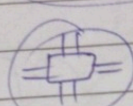
Ans

☆ DeadLock :-

ये सब Process चल रहे हैं।
- Chrome
- Notepad
- Copy-Paste

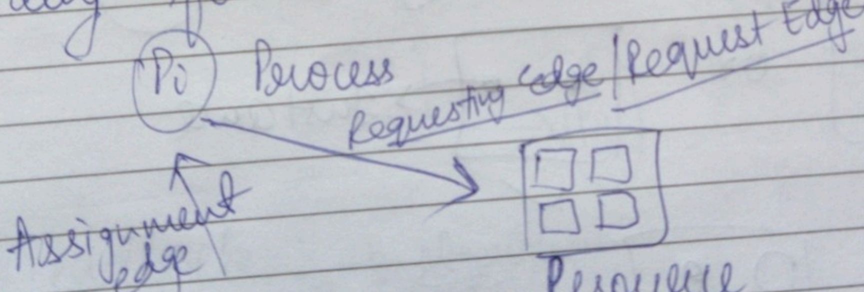


we were use → Resources of System –
RAM, Registers (11), L1/L2 Cache (11)
(ये तीनों Process RAM का use करते थे)



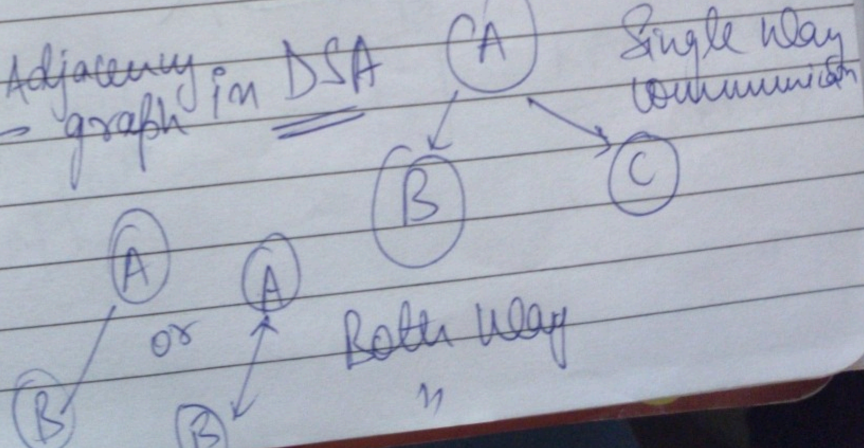
* जिस भी App को Resource चाहिए होगा वो Request करेगी OS को

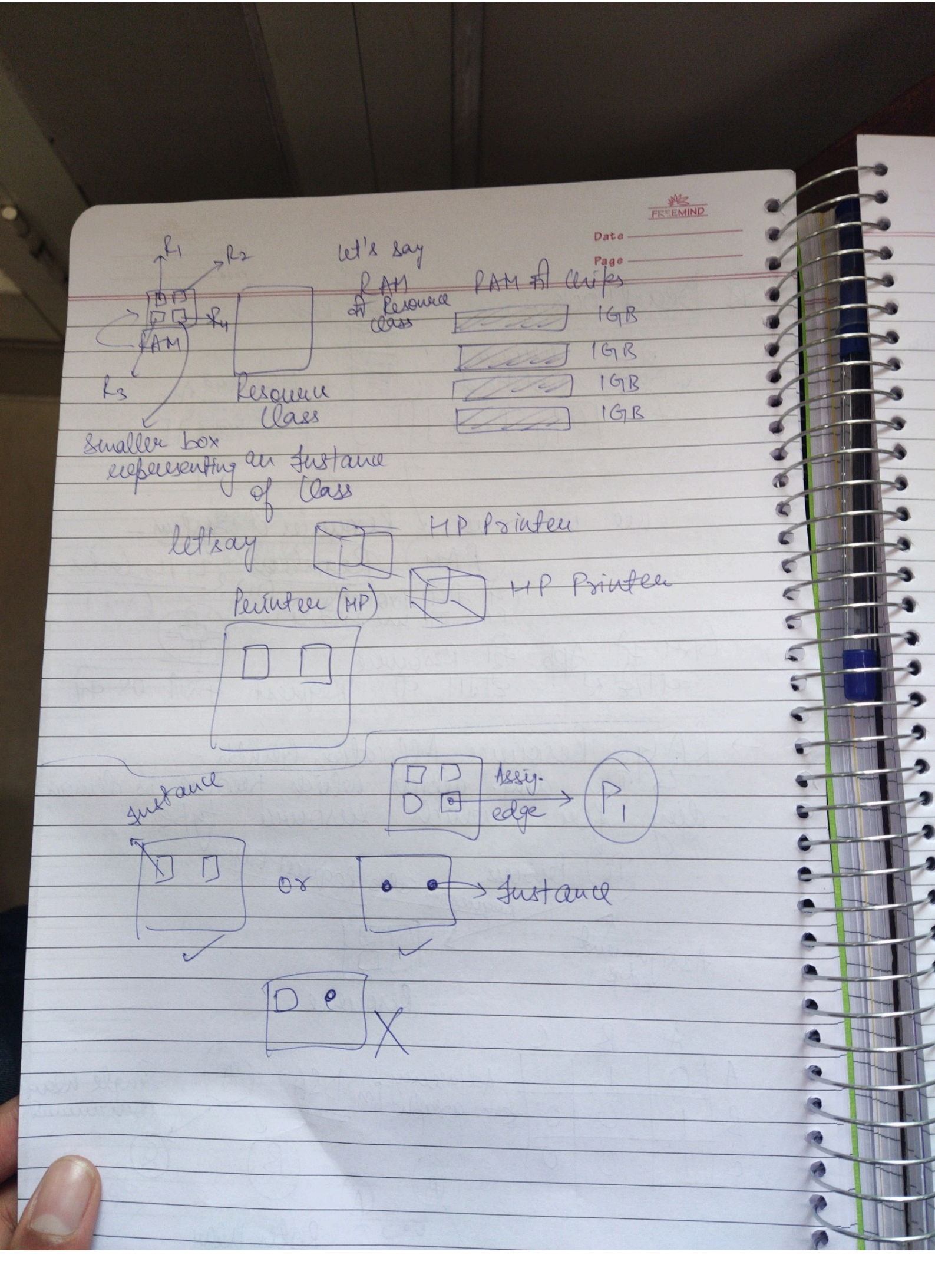
→ RAG – Resource Allocation Graph –
↳ This diag. depicts which process is demanding for which resource type.



	A	B	C
A	0	1	1
B	1	0	0
C	1	0	0

← Adjacency graph in DSA



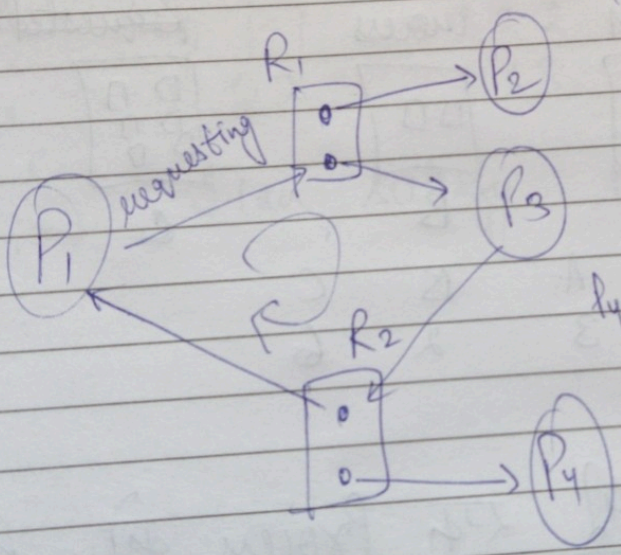

FREEMIND
Date
Page
R1
R2
R4
RAM
R3
Resource Class
Let's say RAM of Resource Class
RAM of chips
1GB
1GB
1GB
1GB
Smaller box representing an instance of class
let's say
HP Printer
HP Printer
Printer (HP)
instance
Assig. edge
P1
or
instance
X

Date 9/9/22
Page

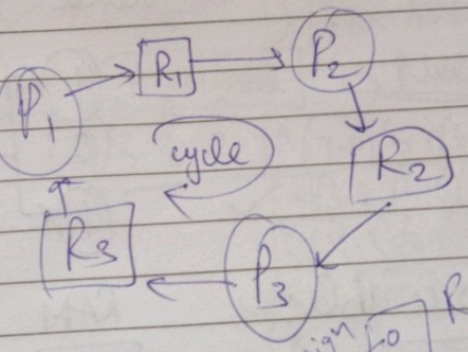
If in Resource Allocation graph में Cycle होती है then it may lead to deadlock situation.

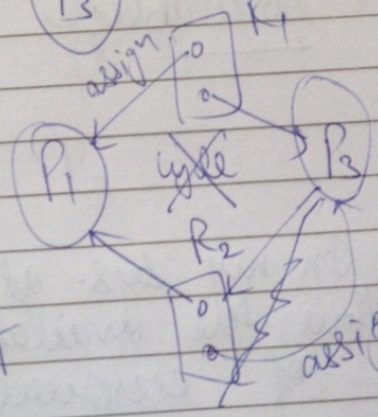
RAG → Cycle → Deadlock may occur

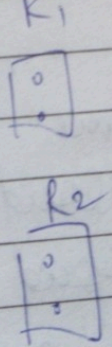
cycle
If ^ can be broken then no deadlock
If " cannot " then surely deadlock will occur



अब P_4, P_2 चला गया exit हो गया







final state
All the process ended
RAM free हो गई

→ Deadlock को 2 Algorithms

Prevent
Static approach

Avoid
Dynamic approach

FREEMIND

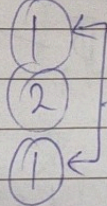
Date

Page

→ Banker's Algorithm :-

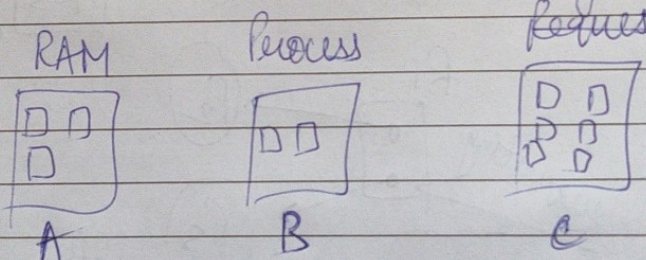
① Safe Sequence Algo./Module
↳ we compute safety sequence

② Resource Request Algorithm
→ my algo will see whether system can allocate resources as requested



~~चलती~~ है ① Module

Available:



In my sys. at n^{th} time the availability of resources →	A	B	C
	3	2	6

Allocations : किसी एक Process को कितने Resources Allocate किए हैं।

	Allocation →	A	B	C
P_1 Chrome	P_1	2	1	0
P_2 MS Word	P_2	1	4	3
P_3 Notepad	P_3	0	0	0

Max : किसी भी Process की Max. demand क्या है i.e ज्यादा से ज्यादा कितने Resources चाहिए हो सकते हैं।

FREEMIND

Date

Page

Let's say → MAX

	A	B	C
P_1	5	3	2
P_2	0	1	0
P_3	2	1	3

Process	Allocation A B C	MAX A B C	Available A B C
P_1	2 0 0	7 5 3	3 3 2
P_2	3 0 2	3 2 2	
P_3	2 1 1	9 0 2	
P_4	0 0 2	2 2 2	
P_0	0 1 0	7 5 3	
P_1	2 0 0	3 2 2	
P_2	3 0 2	9 0 2	
P_3	2 1 1	2 2 2	
P_4	0 0 2	4 3 3	

(MAX - Allocation) → Need

Available A B C	Need
3 3 2	✗ (7-0)(5-1)(3-0) = 143
✓ →	(3-2)(2-0)(2-0) = 122
	✗ (9-3)(0-0)(2-2) = 600
✓ →	0 1 1
	✗ 4 3 1

A B C
(3, 3, 2) I can satisfy P_1 or P_3