MAIN MEMORY
NUMERICALS

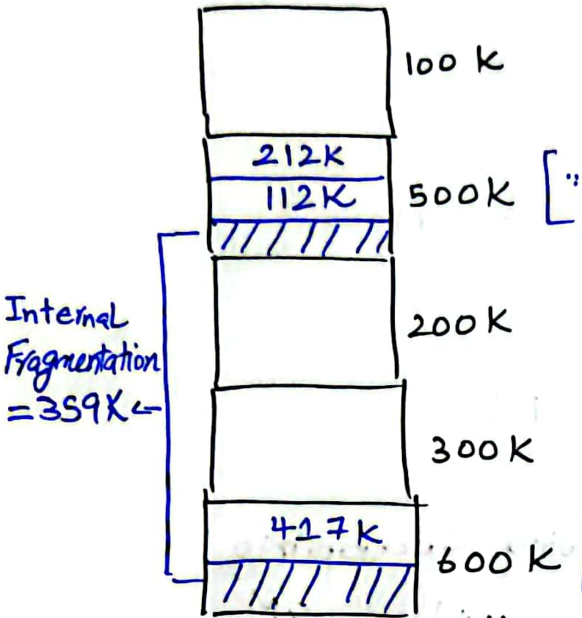
Qasim Hasan
21K-3210

TOPIC 1 :
Dynamic Storage Allocation Techniques

Q:// Memory Partitions : 100K , 500K , 200K , 300K , 600K
Processes : 212K , 417K , 112K , 426 Kb

i:) First - Fit :

212K put in 500K
417 put in 600K
112K put in new 288K
426K must wait



[∵ New Partition ⇒ 500 − 212 = 288 K]
NP ⇒ 288 − 112 = 176 K]

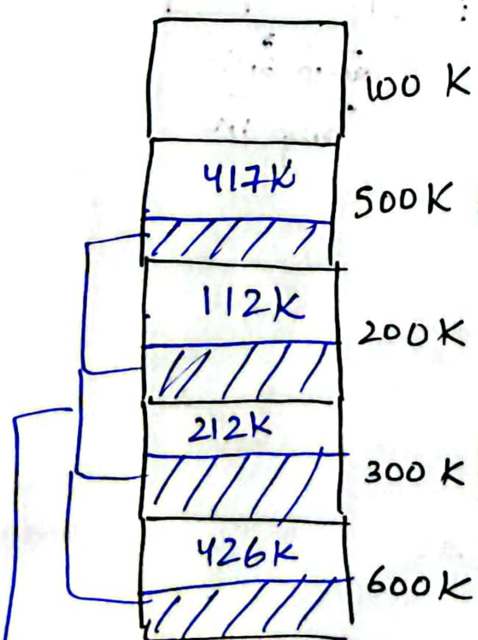
External Fragmentation / Free Space = 600K left after Processes

Total Memory left = 959K

[∵ New Partion ⇒ 600 − 417 = 183 k]

ii:) Best - Fit : (minimum Internal fragmentation)

212K put in 300K
417K put in 500K
112K put in 200K
426K put in 600K



[∵ NP ⇒ 500 − 417 = 83K]

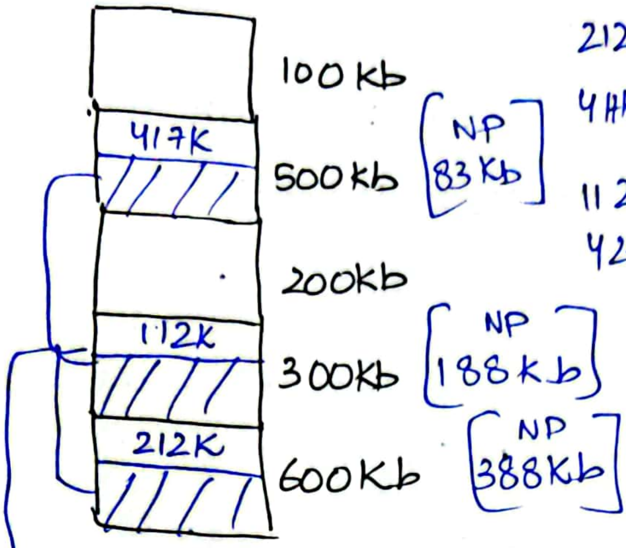
[∵ NP ⇒ 200 − 112 = 88K]

[∵ NP ⇒ 300 − 212 = 88K]

[∵ NP ⇒ 600 − 426 = 176K]

Internal Fragmentation = 435
Free Memory after Individual = 100K
Total Memory left = 535K

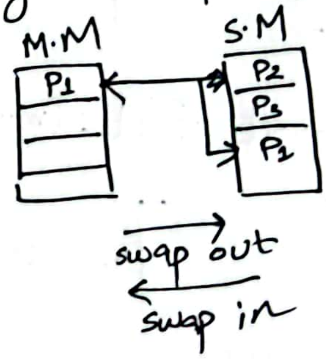
Worst fit : (Largest Size)



212K put in 600K
414K put in 500k
112K is put in 388K
426K must wait

→ Internal Fragmentation = 659Kb
Free Space = 300kb
Total Free Space = 959 kb

TOPIC 2: Context Switch Time Incluing Swapping.



$$\text{Swap Time} = \frac{\text{Size}}{\text{Transfer rate}}$$

P=1000 Mb
Transfer=50 mb/sec

$$\text{Swap Time} = \frac{10\cancel{0}\cancel{0}}{5\cancel{0}}$$
$$= 2\text{sec}$$

Swap[out] = 2 sec
+
Swap[in] = 2 sec

⇒ Swap[Total] = 4 sec

TOPIC 3: Page Replacement Algorithms (VM)

1.) FIFO (highest Time)
2.) Optimal Solution →
3.) LRU Algorithm ←
4.) Second - Chance Algorithm

* Page Fault
✓ Hit

1.) FIFO:

Replace that has been in memory longest Time.

ex: 701203042303120

f_1	7	7	7	2	2	2	2	4	4	4	0	0	0	0	0
f_2		0	0	0	0	3	3	3	2	2	2	2	1	1	1
f_3			1	1	1	1	0	0	0	3	3	3	3	2	2
	*	*	*	*	✓	*	*	*	*	*	*	✓	*	*	✓

Page fault $= \frac{12}{15} \times 100$

$\boxed{\text{Page Fault} = 80\%}$

Page Hit: 3

Page fault $= 12$

Hit ratio $= \frac{\text{No of hits}}{\text{Total Pages}}$

$= \frac{3}{15} \times 100$

$\boxed{\text{Hit ratio} = 20\%}$

2.) Optimal: (Page will use late)

ex: 701203042303212017

	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7
f_1	7	7	7	2	2	2	2	2	2	2	2	2	2	2	2	2	2	7
f_2		0	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0
f_3			1	1	1	3	3	3	3	3	3	3	3	1	1	1	1	1
	*	*	*	*	✓	*	✓	*	✓	✓	*	✓	✓	*	✓	✓	✓	*

Page Hit : 9

Page False : 9

Page Hit $= \frac{9}{18} \times 100 = 50\%$

Page Fault $= \frac{9}{18} \times 100 = 50\%$

3.) Least Recently Used:

	7	0	1	2	0	3	0	4	2	3	0	3	1	2	0
f_1	7	7	7	2	2	2	2	4	4	4	0	0	0	2	2
f_2		0	0	0	0	0	0	0	0	3	3	3	3	3	0
f_3			1	1	1	3	3	3	2	2	2	2	1	1	1
	*	*	*	*	✓	*	✓	*	*	+	*	✓	*	*	*

Page Hit = 3

Page Fault = 12

Page Hit : $\frac{3}{15} \times 100 = 20\%$

Page Fault : $\frac{12}{15} \times 100 = 80\%$

4.) Second Chance Algorithm (Reference bit + FIFO)

ex:

	2	3	2	1	5	2	4	5	3	2	5	2
f_1	$2^{(0)}$	$2^{(0)}$	$2^{(1)}$	$2^{(1)}$	$2^{(0)}$	$2^{(1)}$	$2^{(0)}$	$2^{(0)}$	$3^{(0)}$	$3^{(0)}$	$3^{(0)}$	$3^{(0)}$
f_2		$3^{(0)}$	$3^{(0)}$	$3^{(0)}$	$5^{(0)}$	$5^{(0)}$	$5^{(0)}$	$5^{(1)}$	$5^{(1)}$	$5^{(0)}$	$5^{(1)}$	$5^{(1)}$
f_3				$1^{(0)}$	$1^{(0)}$	$1^{(0)}$	$4^{(0)}$	$4^{(0)}$	$4^{(0)}$	$2^{(0)}$	$2^{(0)}$	$2^{(1)}$
	*	*	✓	*	*	✓	*	✓	*	*	✓	✓

Page Hit : 5

Page Fault : 7

Hit : $\frac{5}{12} \times 100 = 41.6\%$

Page Fault : $\frac{7}{12} \times 100 = 58.4\%$

Second Chance

	3	6	8	7	0	6	3	3	9	2	8	7	2	1	1
f_1	$3^{(0)}$	$3^{(0)}$	$3^{(0)}$	$7^{(0)}$	$7^{(0)}$	$7^{(0)}$	$3^{(0)}$	$3^{(1)}$	$3^{(1)}$	$3^{(1)}$	$8^{(0)}$	$3^{(0)}$	$2^{(0)}$	$2^{(0)}$	$2^{(0)}$
f_2		$6^{(0)}$	$6^{(0)}$	$6^{(0)}$	$0^{(0)}$	$0^{(0)}$	$0^{(0)}$	$0^{(0)}$	$9^{(0)}$	$9^{(0)}$	$8^{(0)}$	$8^{(0)}$	$8^{(0)}$	$1^{(0)}$	$1^{(1)}$
f_3			$8^{(0)}$	$8^{(0)}$	$8^{(0)}$	$6^{(0)}$	$6^{(0)}$	$6^{(0)}$	$6^{(0)}$	$2^{(0)}$	$2^{(0)}$	$7^{(0)}$	$7^{(0)}$	$7^{(0)}$	$7^{(0)}$
	*	*	*	*	*	*	*	!	*	*	*	*	+	*	!

	0	9	8	4	7	2	6	0
	$2^{(0)}$	$9^{(0)}$	$9^{(0)}$	$4^{(0)}$	$4^{(0)}$	$4^{(0)}$	$6^{(0)}$	$6^{(0)}$
	$1^{(1)}$	$1^{(1)}$	$1^{(0)}$	$1^{(0)}$	$7^{(0)}$	$7^{(0)}$	$7^{(0)}$	$0^{(0)}$
	$7^{(0)}$	$0^{(0)}$	$8^{(0)}$	$8^{(0)}$	$8^{(0)}$	$2^{(0)}$	$2^{(0)}$	$2^{(0)}$
	*	*	*	*	*	*	*	*

Hit ratio $= \frac{2}{23} = 8.6\%$

Second Chance:

	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2
f_1	$1^{(0)}$	$1^{(0)}$	$1^{(0)}$	$4^{(0)}$	$4^{(0)}$	$4^{(0)}$	$5^{(0)}$	$5^{(0)}$	$5^{(0)}$	$1^{(0)}$	$1^{(0)}$	$1^{(0)}$	$7^{(0)}$	$7^{(0)}$	$7^{(0)}$	$2^{(0)}$
f_2		$2^{(0)}$	$2^{(0)}$	$2^{(0)}$	$2^{(1)}$	$2^{(0)}$	$2^{(0)}$	$6^{(0)}$	$6^{(0)}$	$6^{(0)}$	$6^{(0)}$	$3^{(0)}$	$3^{(0)}$	$6^{(0)}$	$6^{(0)}$	$6^{(0)}$
f_3			$3^{(0)}$	$3^{(0)}$	$3^{(0)}$	$1^{(0)}$	$1^{(0)}$	$1^{(0)}$	$2^{(0)}$	$2^{(0)}$	$2^{(1)}$	$2^{(1)}$	$2^{(1)}$	$2^{(0)}$	$3^{(0)}$	$3^{(0)}$
	*	*	*	*	!	*	*	*	*	*	!	*	*	*	*	*

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

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

	1	2	3	6
f_1	$2^{(0)}$	$2^{(1)}$	$2^{(1)}$	$2^{(0)}$
f_2	$1^{(0)}$	$1^{(0)}$	$1^{(0)}$	$6^{(0)}$
f_3	$3^{(0)}$	$3^{(0)}$	$3^{(1)}$	$3^{(0)}$
	*	!	!	*

LRU : ←

	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1
f_1	1	1	1	4	4	4	5	5	5	1	1	1	7	7	7	2	2
f_2		2	2	2	2	2	2	6	6	6	6	3	3	3	3	3	3
f_3			3	3	3	1	1	1	2	2	2	2	2	6	6	6	1
	*	*	*	*	!	*	*	*	*	*	!	*	*	*	!	*	*

2	3	6
2	2	2
3	3	3
1	1	6
~~*~~ !	~~*~~ !	*

$$Hit = \frac{5}{20} \times 100 = 25\%$$

$$Faults = \frac{15}{20} \times 100 = 75\%$$

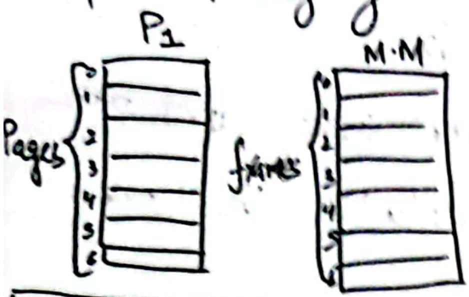
Optimal :

	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
f_1	1	1	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	6
f_2		2	2	2	2	2	2	2	2	2	2	2	7	7	7	2	2	2	2	2
f_3			3	4	4	4	5	6	6	6	6	6	6	6	6	6	1	1	1	1
	*	*	*	*	!	!	*	*	!	!	!	*	*	!	!	*	*	!	!	*

$$Hit = \frac{9}{20} \times 100 = 45\%$$

$$Fault = \frac{11}{20} \times 100 = 55\%$$

Topic 4: Paging



$\boxed{* \text{ Page Size} = \text{Frame Size}}$

$$\text{Number of Frame} = \frac{\text{M·M Size}}{\text{Frame size}}$$

$$\text{Number of Pages} = \frac{\text{Process Size}}{\text{Page Size}}$$

$\boxed{* \text{ Page Entries} = \text{Page Number}}$

MAPPING

CPU

Logical Address
LAS

Page (Binary) Number	Size of Page (byte)
p	d

Main Memory
Physical Address
PAS

Frame (Binary) Number	Size of (byte) Frame

$2^1=2$	$2^6=64$	$2^{20}=1M$
$2^2=4$	$2^7=128$	$2^{30}=1G$
$2^3=8$	$2^8=256$	$2^{40}=1T$
$2^4=16$	$2^9=512$	
$2^5=32$	$2^{10}=1K$	

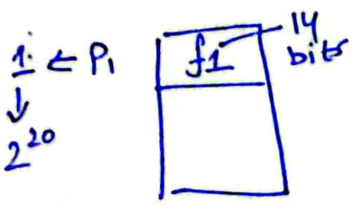
Q1.// LAS = 4GB
PAS = 64MB
Page Size = 4 Kb

Find:
Number of Pages = ?
Number of Frame = ?
Number of Entries = ?
Size of Page Table

$LA = 4GB = 2^2 \times 2^{30} = 2^{32}$

$\text{Page size} = 4Kb = 2^2 \times 2^{10} = 2^{12}$

$PA = 64MB = 2^6 \times 2^{20} = 2^{26}$



i:) LA

Page No | Page Size

20	12

$\xleftarrow{\quad 32 \quad}$

Page Number $= 2^{20}$

ii:) PA

Frame Number | frame size

14	12

$\xleftarrow{\quad 26 \quad}$

$\because$ Page Size = Frame Size

Frame Number $= 2^{14}$

iii:) $\because$ No of entries = No Pages $= 2^{20}$

iv.) Page Table Size = Page Number $\times$ Frame Number (bits)

$= 2^{20} \times 14$

Q6.) Consider a system with byte-addressable memory, 32 bit logical addresses, 4 Kilobyte page size, and page Table entries of 4 bytes each. Calculate the page Table size in the system.

LA = 32 bit

Page Size = 4 Kb $= 2^2 \times 2^{10} = 2^{12}$

LA

20	12

$\xleftrightarrow{32}$

Number of Pages $= 2^{20}$

∴ Page Table ki Entries = frames = 4 byte $= 2^2$ ~~byt~~

0	f_1
1	f_2
	⋮

4 byte

Page Table Size $= 2^{20} \times 2^2$ or $2^{20} \times 4$

$= 2^{22}$

= 4 mb

Q8.) Consider a Virtual address space of 32 bit and Page Size of 4KB. System Ram of 128 Kb. Then what will ratio of Page Table and Inverted Page Table Size if each entry in both is 4B.

LA PS = 4Kb $= 2^{12}$

20	12

$\xleftrightarrow{32}$

PA PS $= 2^{10} \times 2^7 = 128$ kb

2^{17}

5	12

$\xleftrightarrow{17}$

PTS $= 2^{20} \times 4 =$

ITS $= 2^5 \times 4$

Ratio: $\frac{\text{Page Table Size}}{\text{Inverted Table size}}$

$\frac{2^{20} \cancel{\times 4}}{2^5 \cancel{\times 4}}$

$\boxed{2^{15} : 1}$

Q2.// LA = 7 ~~byte~~ bits

PA = 6 bits

Page Size = 8 Words = 2^3

Find

i) Number of Pages = ?

ii) Number of Frames = ?

i)

Page No LA

4	3

Page Size..

$\longleftarrow 7 \longrightarrow$

Number of Pages = 2^4

ii)

Frame Number PA

3	3

frame size

$\longleftarrow 6 \longrightarrow$

Number of frame = 2^3

Q3.// Calculate the size of memory if its address consists of 22 bits and the memory is 2-byte addressable

Size of Memory = No of location × Size of location

$= 2^{22} \times 2^1$

$= 2^{23}$

$= 2^{20} \cdot 2^3$

$= 8MB$

Q4.// Calculate the number of bits required in the address for a memory having a size of 16GB. Assume the memory is 4 byte addressable

Size of memory = $2^4 \times 2^{30} = 2^{34}$

Size of location = 2^2

Size of Memory = No of location × Size of location

2^{34} = Num of location $\times 2^2$

Num of location $= \frac{2^{34}}{2^2} = 2^{32}$

$\boxed{\text{Num of location} = 32 \text{ bits}}$

Q7.//

$PAS = 256\,MB = 2^8 \times 2^{20} = 2^{28}$

$LAS = 4GB$

Frame Size $= 4kb = 2^2 \times 2^{10} = 2^{12}$

Page table Entry $= 2B$

PA

Frame	size
16	12

$\longleftarrow 28 \longrightarrow$

Number of Frame $= 2^{16} = 65536$

LA

Pages	size
20	12

$\longleftarrow 32 \longrightarrow$

Number of Pages $= 2^{20}$

PT Size $= 2^{20} \times$ Entries

$= 2^{20} \times 2$

$= 2MB.$; 2MB > 4kb

32

Inner	20	12
Outer	9 | 11	

For Outer Page Table

$O.T = \dfrac{2MB}{4kb} = \dfrac{2^{21}}{2^{12}} = 2^9$

OPT Size $= 2^9 \times 2 = 1Kb$; $1Kb < 4kb$

Q9.11 4Kb Page Size, what are the Page Numbers and offsets for the following address references (provided as Decimal Numbers)

Page Size: $4Kb = 2^2 \times 2^{10} = 2^{12}$

m-n	n
Pages	12

← Memory Size →

a) 3085_{10}

$3085_{10} = 110000001101_2 \rightarrow m = 12$

$\text{Pages} = 2^{12-12} = 2^0 = 1$

$\text{offset} = 110000001101_2 = 3085_{10}$

b.) 42095 30784770|

$42095_{10} = 1010010001101111_2 \rightarrow m = 16$

$\text{Pages} = 2^{16-12} = 2^4 = 16$

$x = \dfrac{42095}{2^{12}} = \underbrace{10}_{m}.\underbrace{27709}_{\text{offset}}$

~~Pages = M-N~~

$\text{offset} = 0.27709 \times 2^{12} = 1135$

If hexa decimal

$12596215h_{16}$

$\dfrac{307847701}{2^{12}} = \underbrace{75158}_{m}.\underbrace{13013}_{\text{offset kahliye}}$

$\text{offset} = 0.13013 \times 2^{12}$

$\text{offset} = 533$

CamScanner

c.) 215201

For Page Numbers

Convert into binary: ~~10001~~ ~~000001~~

1101 0010 0010100001$_2$ → m = 18

$$\text{Pages} = 2^{18-12} = 2^6 = \boxed{64}$$

2	215201	
	107600	1

$$\text{offset} = \frac{215201}{2^{12}} = 52.53930$$

$$\text{offset} = 0.5393 \times 2^{12} = \boxed{2209}$$

Q10.) LA = 32 bit

Page size = 4 Kb

$PA = 512MB = 2^9 \times 2^{20} = 2^{29}$

Entries = ?

Pages = ?

Frames = ?

LA

20	12

← 32 →

PA

17	12

← 29 →

$PS = 4Kb = 2^2 \times 2^{10} = 2^{12}$

$\text{Frames} = 2^{17} = 131072$

$\text{Pages} = 2^{20} = 1048576$

Level Paging: (2/3 Level)

$EAT = hit(TLB + MM) + miss(TLB + (Pages+1) \times MM)$

i) Consider a three level paging scheme with TLB. Assume no page fault occurs. It takes 30ns to search the TLB and 100ns to access ~~access~~ the physical memory. If the TLB hit ratio is 70% What will be its EAT:-

$$EAT = hit(TLB + MM) + miss(TLB + (Pages+1) \times MM)$$
$$= 0.7 \times (30 + 100) + 0.3(30 + (3+1) \times 100)$$
$$= 220\ ns$$

ii) Two level paging. No Page fault occurs. It 25ns for TLB, 100ns for MM. TLB hit ratio is 75%?

$$EAT = hit(TLB + MM) + miss(TLB + (Pages+1) \times MM)$$
$$= 0.75(25 + 100) + 0.25\{25 + (2+1) \times 100\}$$
$$= 175\ ns$$

iii) One-Level

Assue MA = 40ns, TLB hit rate 90% of 10 ns. EAT = ?

$\because MA = PT$

$$EAT = hit(TLB + MA) + miss(TLB + PT + MA)$$
$$= 0.9(10 + 40) + 0.1(10 + 40 + 40)$$
$$= 45 + 9$$
$$= 52\ n.s$$

Main Memory

TLB:

$EMAT = Hit(TLB + M.M) + Miss(TLB + PT + M.M)$

$PT = M.M$

One-level

Q1// A Paging scheme using TLB. TLB Access time 10ns and main memory access time take 50ns What is EMAT (in ns) if TLB hit ratio is 90% and there is no Page fault

$EMAT = 90\%(10+50) + 10\%(10+50+50)$ | + PFAT Page fault Access Time

$EMAT = 65\ ns$

Virtual Memory

V.M:

$EAT = (1-P) \times ma + P(\text{page Fault time})$

P = Page fault rate

Ma = Memory Access

Average Access Time $= (1-P) \times ma + P(ma + \text{Fault } \cancel{\text{time}} \text{ overhead})$

swap in swap out overhead

Q//

MA = 100ns

Page Fault ~~time~~ = 10ms

$P = 0.1\% = \frac{0.1}{100} = 0.001$

$EAT = (1 - 0.00\cancel{0}1)100 + 0.001\underbrace{(\cancel{100}^{0.1} + 1\cancel{0}^{000})}_{\text{Page Fault Time}}$

$= (0.999) \times \cancel{100}^{0.1} + 0.11(\cancel{110})$

$= 99.9 / + \cancel{0}.11 = 0.0999 + 10.0001$

$= 100.01ns = \boxed{10.1\mu s}$

Milli to Micro $\times 1000$

Nano to Micro $\div 1000$

CamScanner

3)

Access Time MM = 1μs

PT = 2μs

Hit = 80%

Miss = 18%

Page Fault = 2%

Average Fault Time

= 25ms

= 25 x 1000

= 25000μs

$$EAT = Hit(\overset{TLB+}{MM}) + miss(PT) + P(\text{Fault time})$$

$$= 0.8(1) + (0.18)(2) + (0.02)(25000)$$

$$= 501.16 \mu s \quad [\div 1000]$$

$$= 0.5 ms$$

Buddy System Allocator

Resource Present $\boxed{256 kb}$:

Satisfy

$R_1 = 70 kb$

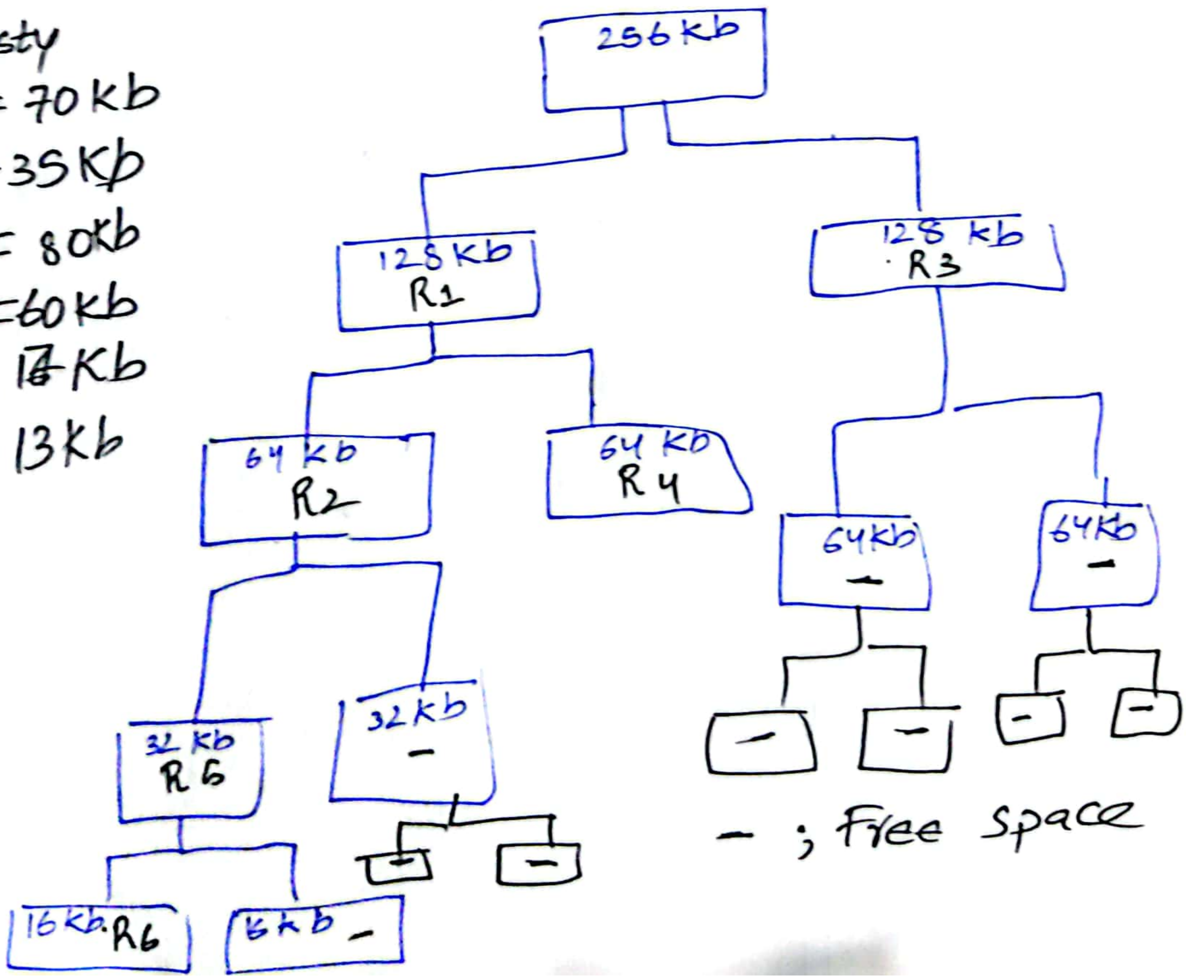
$R_2 = 35 kb$

$R_3 = 80 kb$

$R_4 = 60 kb$

$R_5 = 17 kb$

$R_6 = 13 kb$



RAG

Resource Allocation Graph

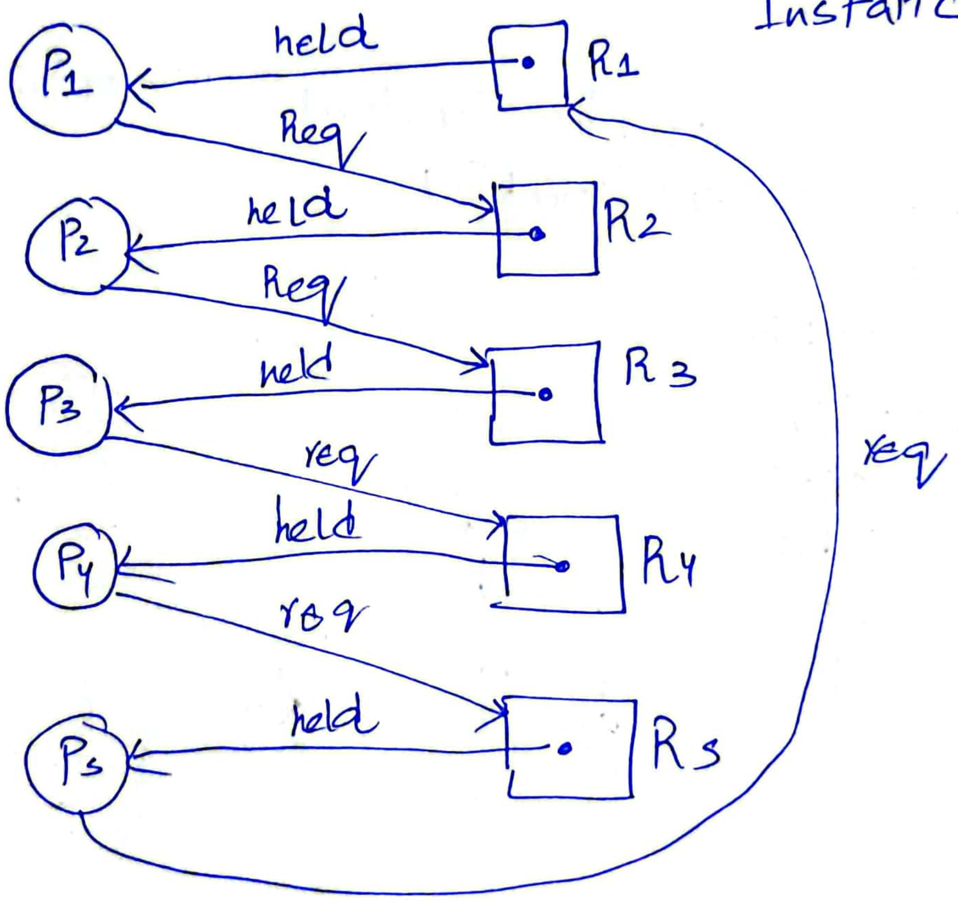
Dining Philosopher Deadlock

5 Philosophers ; Process

5 Chopsticks ; Resources

∵ Each Philosopher need 2

Single Instance



∵ Deadlock Occurs

Resource Allocation Graph

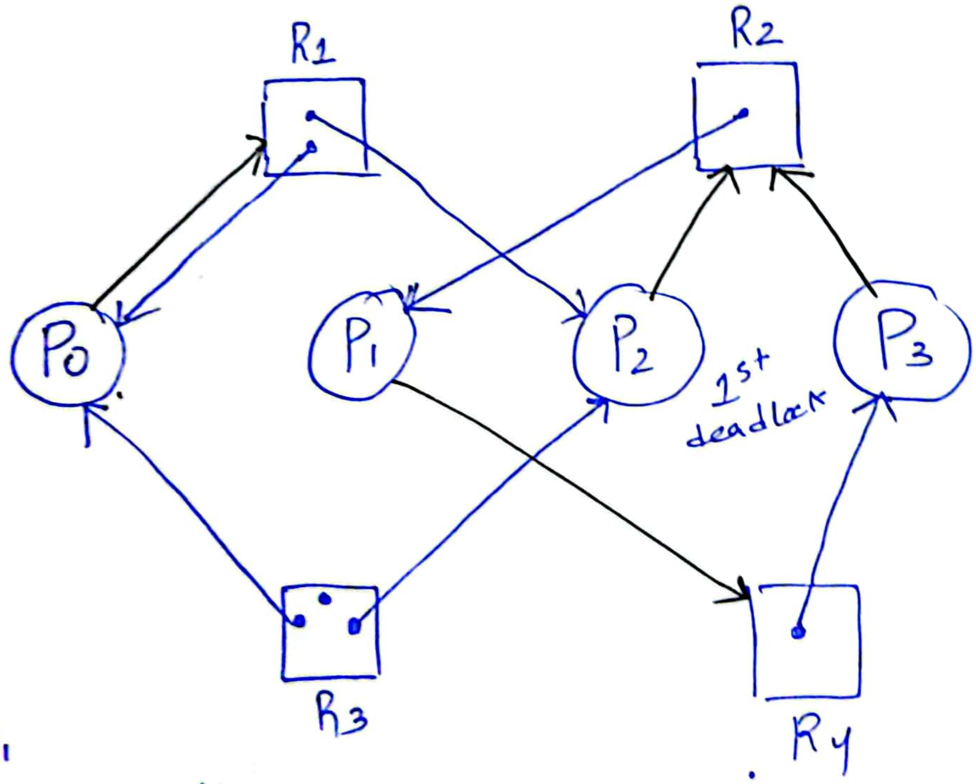
Resource	Instance
R1	2
R2	1
R3	3
R4	1

Request Resources

Process	Resource
P0	R1
P3	R2
P2	R2
P1	R4

Allocated Resources

Process	Resource
P0	R1
P1	R2
P2	R1
P3	R4
P0	R3
P2	R3



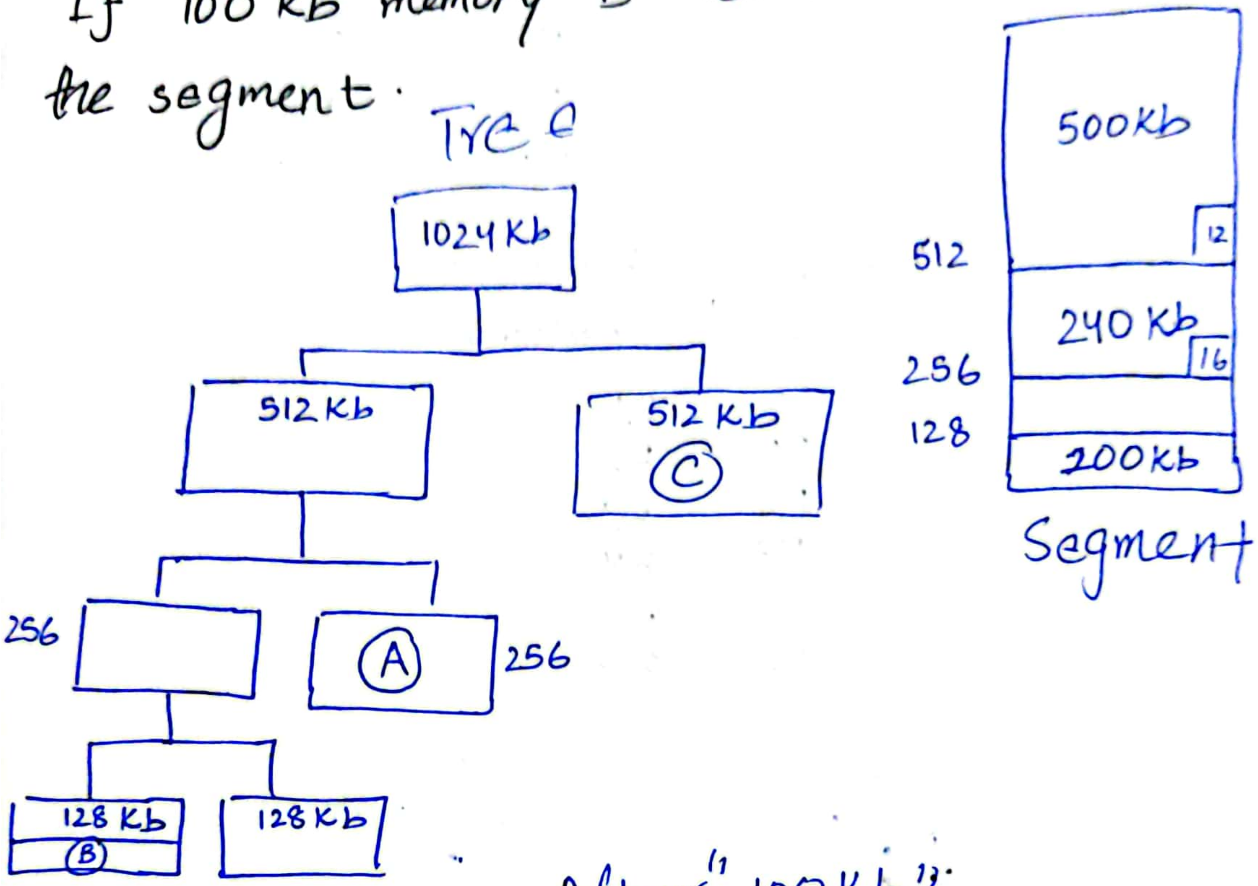
$R_2 \rightarrow P_1$

$P_1 \rightarrow R_4$ Deadlock

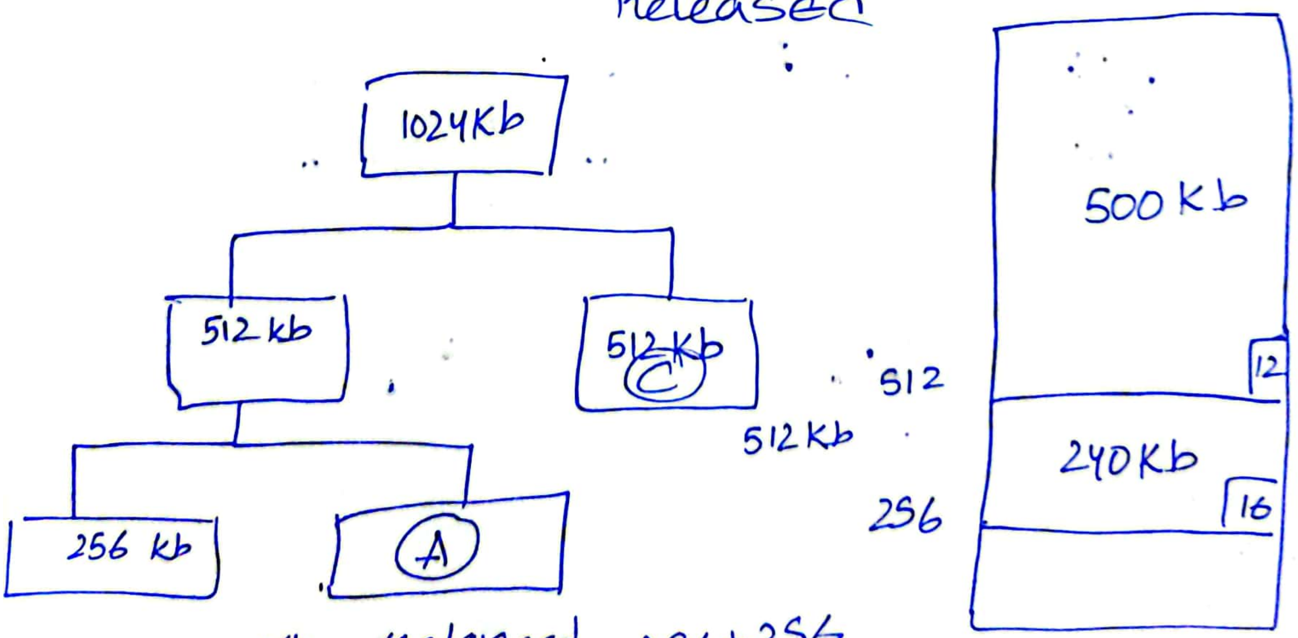
$R_4 \rightarrow P_3$

$P_3 \rightarrow R_2$

In a 1024 kb segment is allocated through Buddy System. Allocate 240Kb (A), 100Kb (B) and 500K (C). If 100 Kb memory is released structure of the segment.



After "100 Kb" Released



If 240 Kb realeased 256+256 will merge to 512 Kb

CPU Scheduling Algorithm

FCFS: (First Come First Serve)

$TAT = CT - AT$
$WT = TAT - BT$
$RT = \text{Arrival on chart} - AT$

↳ Criteria: Arrival Time*
↳ Mode: Non-Pre-emptive (Non-stop)

Process No	Arrival Time	Burst Time	Completion Time	TAT	WT	RT
P_1	0	2	2	2	0	0
P_2	1	2	4	3	1	1
P_3	5	3	8	3	0	0
P_4	6	4	12	6	2	8-6 =2

Gantt Chart:

P_1	P_2	▨	P_3	P_4
0 – 2	2 – 4	4 – 5	5 – 8	8 – 12

Time →

$Avg.TAT = \frac{14}{4} = 3.5$

$Avg.WT = \frac{3}{4} = 0.75$

SJF: (Shortest Job First)

↳ Criteria: Burst Time
↳ Mode: Non-Preemptive

Process	AT	BT	CT	TAT	WT	RT
✓ P_1	1	3	6	5	2	2
P_2	2	4	10	8	4	8
✓ P_3	1	2	3	2	0	2
P_4	4	4	14	10	6	10

$Avg.TAT = \frac{25}{4} = 6.25$

$Avg.WT = \frac{12}{4} = 3$

Gantt Chart:

▨	P_3	P_1	P_2	P_4
0 – 1	1 – 3	3 – 6	6 – 10	10 – 14

SJF : SRJF
↳ Crietria : Burst Time
↳ Mode : Pre emptive

Process	AT	BT	CT	TAT	WT	RT
P_1	0	~~5~~ 4	9	9	~~5~~ 4	0-0=0
✓P_2	1	~~3~~ ~~1~~ 0	4	3	~~2~~ 0	1-1=0
P_3	2	4	13	11	7	9-2=7
P_4	4	1	5	1	0	4-4=0

Gantt Chart :

P_1	P_2	P_2	P_4	P_1	P_3

0 1 2 4 5 9 13

$\text{Avg TAT} = \frac{24}{4} = 6$

$\text{Avg WT} = \frac{11}{4} = \cancel{3.75}\ 2.75$

Round- Robbin :
↳ Crietria : Time Quantum
↳ Mode : Pre-emptive

Process	AT	BT	CT	TAT	WT	RT
P_1	0	~~5~~ ~~3~~ 1	12	12	7	0
P_2	1	~~4~~ ~~2~~ 0	11	10	6	1
✓P_3	2	~~2~~ 0	6	4	2	2
P_4	4	~~1~~ 0	9	5	4	4

; TQ=2

Avg TAT = 7.75
Avg WT = 4.75

Gantt Chart

Ready Queue

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

0

Context switch

Running Queue

P_1	P_2	P_3	P_1	P_4	P_2	P_1

0 2 4 6 8 9 11 12

Priority Scheduling:

Criteria: "Priority"

Mode: "Pre-emptive"

Highest the number Higher the Number

Process	Priority	AT	BT	CT	TAT	WT
✓ P_1	10	0	~~5~~ 4	12	12	7
✓ P_2	20	1	~~4~~ ~~3~~ 0	8	7	3
✓ P_3	30	2	~~2~~ 0	4	2	0
✓ P_4	40	4	~~1~~ 0	5	1	0

Gantt Chart:

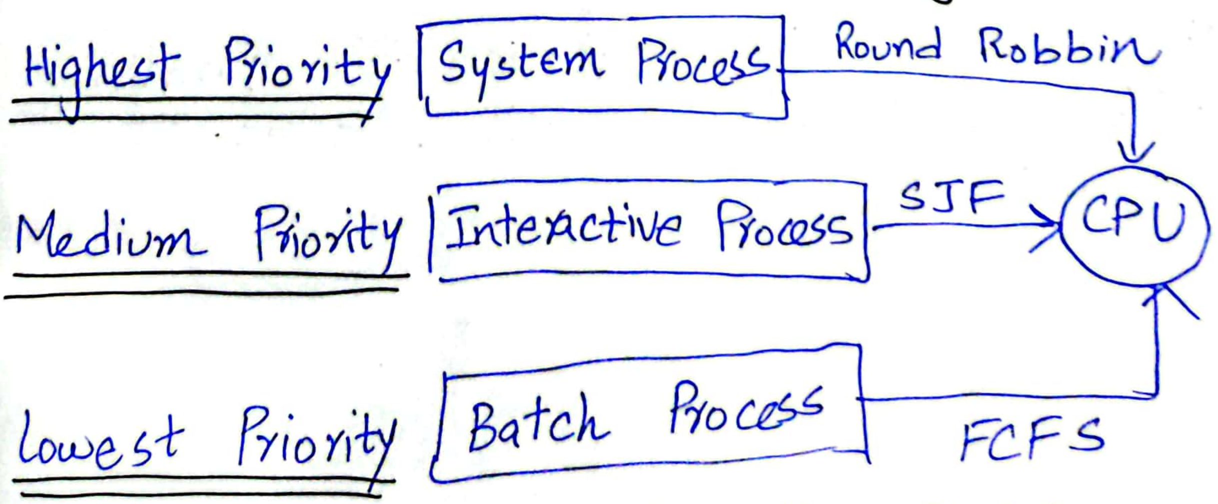
P_1	P_2	P_3	P_4	P_2	P_1

0   1   2   4   5   8   12

$$Avg \cdot TAT = \frac{22}{4} = 5.5$$

$$Avg \cdot WT = \frac{10}{4} = 2.5$$

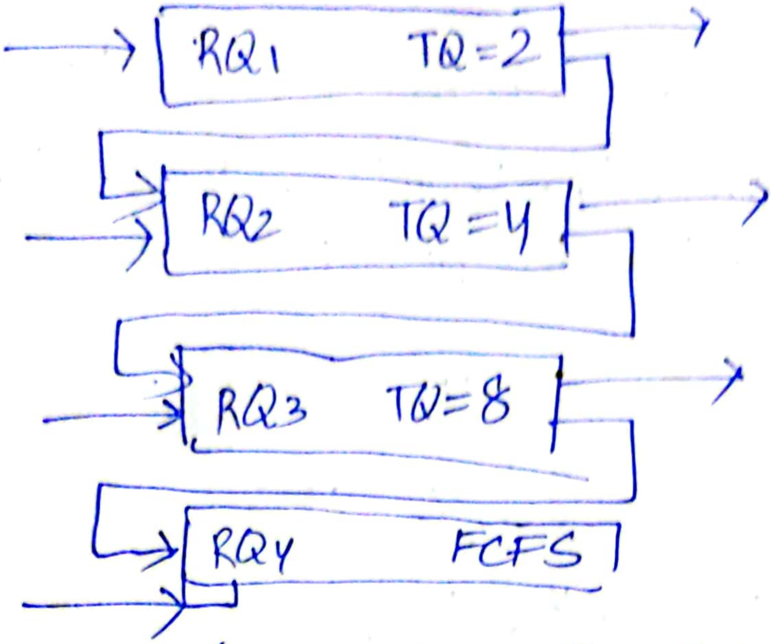
Multi-level Queue Scheduling:



"Starvation" → Problem

Multi-Level Feedback Queue:

Lowest priority



"Starvation" Solved

Amdahl's law:

$$\text{speedup} \leq \frac{1}{S + \frac{(1-S)}{N}}$$

where,

S = Portion of Program executed serially

N = Processing Cores

"How much computation can be speedup by running part of a Program in Parallel."

Q.//

We have an application that is 75% Parallel and 25% serial. If we run with 2 cores. It Speedup will be ?

$S = 25\% \Rightarrow 0.25$

$N = 2$

$$\text{speedup} \leq \frac{1}{0.25 + \frac{(1-0.25)}{2}}$$

$$\text{speedup} \leq 1.6$$

BANKER'S ALGORITHM

Q1.)

Process	Allocation			Maximum			Available			Remaing Need (max-alloc)		
	A	B	C	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	5	3	3	3	2	7	4	3
P_1	2	0	0	3	2	2	5	3	2	1	2	2
P_2	3	0	2	9	0	2	6	4	3	6	0	0
P_3	2	1	1	4	2	2	7	4	5	2	1	1
P_4	0	0	2	5	3	3	7	5	5	5	3	1
	7	2	5				10	5	7			

Total A=10 ⇒ 10-7 = 3
B=5 ⇒ 5-2 = 3
C = 7 ⇒ 7-5 = 2
→ Available Work

Safe segment = {P_1, P_3, P_4, P_0, P_2}

Need ≤ Work

P_0: 743 ≤ 332 ✗

P_1: 122 ≤ 332 ✓

∴ Work = 332+200 = 532

P_2: 600 ≤ 532 ✗

P_3: 211 ≤ 532 ✓

∴ Work = 532+211 = 643

P_4: 531 ≤ 643 ✓

∴ Work = 643+002 = 645

P_0: 743 ≤ 745 ✓

∴ Work = 745+010 = 755

P_2: 600 ≤ 755 ✓

∴ Work = 755+302 = 10 5 7

Process	Allocated (E F G)	Max (E F G)	Available / Work (E F G)	Need (Max - alloc)
P_0	1 0 1	4 3 1	3 3 0	3 3 0
P_1	1 1 2	2 1 4	4 3 1	1 0 2
P_2	1 0 3	1 3 3	5 3 4	0 3 0
P_3	2 0 0	5 4 1	6 4 6	3 4 1
			8 4 6	

Need $\leq$ Work

Safe Segment $= \{P_0, P_2, P_1, P_3\}$

P_0: 330 $\leq$ 330 ✓

Work = 330 + 101 = 431

P_1: 102 $\leq$ 431 ✗

P_2: 030 $\leq$ 431 ✓

Work = 431 + 103 = 534

P_3: 341 $\leq$ 534 ✗

P_1: 102 $\leq$ 534 ✓

Work = 534 + 112 = 646

P_3: 341 $\leq$ 646 ✓