1) Consider the following snapshot at time t_0

Snapshot at time T0:

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

(i) Is the s/m in a safe state?

(ii) If a request from P1 arrives for (1, 0 2) can the request be granted immediately

Soln: Total instances of $A = 7+3 = 10$; $B = 2+3 = 5$; $C = 7$

(i) Safety algorithm:

work = Available, Finish[$P_0 - P_4$] = False

work = [3, 3, 2]

(1) $i = P_0$ need[P_0] $\leq$ work

[7, 4, 3] $\not\leq$ [3, 3, 2] P_0 cannot execute, wait

(2) $i = P_1$ need[P_1] $\leq$ work

[1, 2, 2] $\leq$ [3, 3, 2]. P_1 executes

work = work + allocation

= [3, 3, 2] + [2, 0, 0]

work = [5, 3, 2], Finish[P_1] = True

(3) $i = P_2$ need $[P_2] \leq$ work

$[6, 0, 0] \nleq [5, 3, 2]$ P_2 cannot execute, wait

(4) $i = P_3$ need $[P_3] \leq$ work

$[0, 1, 1] \leq [5, 3, 2]$ P_3 executes

work = work + allocation $[P_3]$

$= [5, 3, 2] + [2, 1, 1]$

work $= [7, 4, 3]$, Finish $[P_3]$ = True

(5) $i = P_4$ need $[P_4] \leq$ work

$[4, 3, 1] \leq [7, 4, 3]$ P_4 executes

work = work + allocation $[P_4]$

$= [7, 4, 3] + [0, 0, 2]$

$= [7, 4, 5]$, Finish (P_4) = True

(6) $i = P_0$ $[7, 4, 3] \leq [7, 4, 5]$, P_0 executes

work $= [7, 4, 5] + [0, 1, 0]$

work $= [7, 5, 5]$, Finish $[P_0]$ = True

(7) $i = P_2$ $[6, 0, 0] \leq [7, 5, 5]$ P_2 executes

work $= [7, 5, 5] + [3, 0, 2]$

work $= [10, 5, 7]$ = Total no. of instances of Resource types

Finish $[P_2]$ = True

$\therefore \langle P_1, P_3, P_4, P_0, P_2 \rangle$ safe sequence

(ii) Snapshot at time T0:

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

$P_1 \xrightarrow{\text{request}} [1, 0, 2]$

(1) $Request[P_1] \leq Need[P_1]$

$[1, 0, 2] \leq [1, 2, 2]$

$\therefore$ requested resources are needed

(2) $Request[P_1] \leq Available$

$[1, 0, 2] \leq [3, 3, 2]$

$\therefore$ requested resources are available

(3) Assume P1 is allocated requested resources.

$Allocate[P_1] = Allocate[P_1] + Request[i]$

$= [2, 0, 0] + [1, 0, 2] = [3, 0, 2]$

$Need[P_1] = Need[P_1] - Request[P_1]$

$= [1, 2, 2] - [1, 0, 2] = [0, 2, 0]$

$Available = Available - Request[P_1]$

$= [3, 3, 2] - [1, 0, 2] = [2, 3, 0]$

Snapshot at time T0:

	Allocation	Maximum	Available	Need
	A B C	A B C	A B C	A B C
P0	0 1 0	7 5 3	**2 3 0**	7 4 3
P1	**3 0 2**	3 2 2		0 2 0 changed
P2	3 0 2	9 0 2		6 0 0
P3	2 1 1	2 2 2		0 1 1
P4	0 0 2	4 3 3		4 3 1

With these changes safety algorithm is checked to find if the s/m is in safe state.

work = available = [2, 3, 0]

Finish [P0 – P4] = False

1) $i = P_0$, need [P0] ≤ work
$[7, 4, 3] \nleq [2, 3, 0]$ P0 cannot execute; waits

2) $i = P_1$, $[0, 2, 0] \leq [2, 3, 0]$ P_1 executes
work = work + allocation[P1]
$= [2, 3, 0] + [3, 0, 2]$
$= [5, 3, 2]$, Finish [P1] = True

3) $i = P_2$, $[6, 0, 0] \nleq [5, 3, 2]$, P_2 cannot execute

4) $i = P_3$, $[0, 1, 1] \leq [5, 3, 2]$, P_3 executes
work $= [5, 3, 2] + [2, 1, 1]$
$= [7, 4, 3]$, Finish [P3] = True

5) $i = P_4$ $[4, 3, 1] \leq [7, 4, 3]$, P_4 executes

$$work = [7, 4, 3] + [0, 0, 2]$$
$$= [7, 4, 5]$$

6) $i = P_0$, $[7, 4, 3] \leq [7, 4, 5]$ P_0 executes

$$work = [7, 4, 5] + [0, 1, 0]$$
$$= [7, 5, 5]$$, Finish $[P_0]$ = True

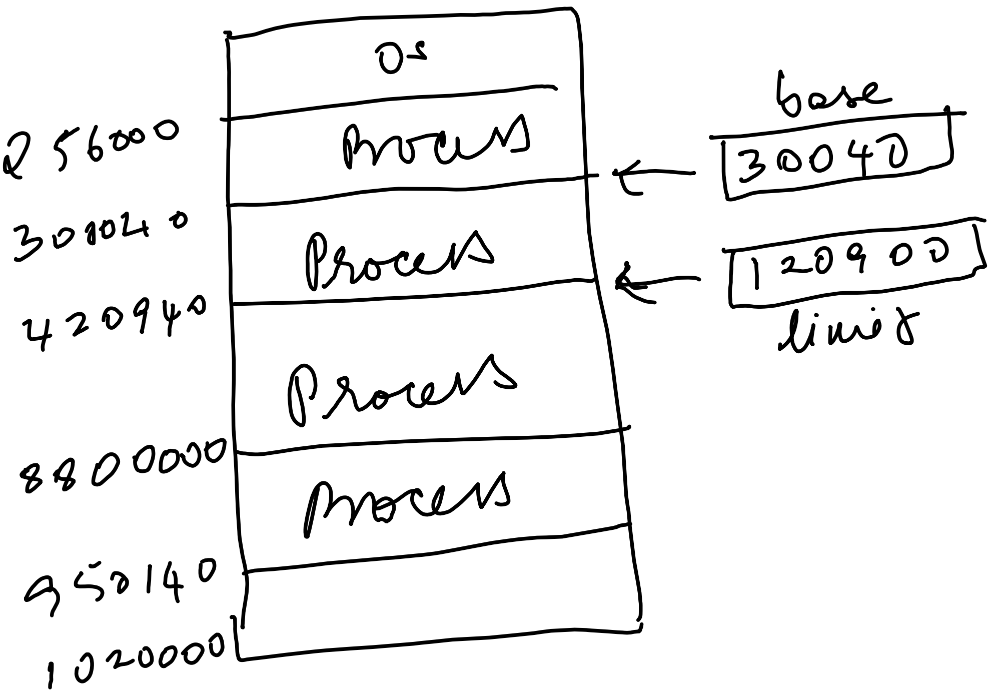
7) $i = P_2$, need $[P_2] \leq$ work

$[6, 0, 0] \leq [7, 5, 5]$ P_2 executes

$$work = [7, 5, 5] + [3, 0, 2]$$
$$= [10, 5, 7]$$

∴ safe sequence is $\langle P1, P3, P4, P0, P2 \rangle$

∴ The request of P1 can be granted



Deadlock detection

Five processes P0 through P4; three resource types
A (7 instances), B (2 instances), and C (6 instances)
Snapshot at time T0:

	Allocation	Request	Available
	A B C	A B C	A B C
P0	0 1 0	0 0 0	0 0 0
P1	2 0 0	2 0 2	
P2	3 0 3	0 0 0	
P3	2 1 1	1 0 0	
P4	0 0 2	0 0 2	

If the requested resources are allocated, check whether s/m will be in safe state.
Also, find the safe sequence.

Sol^n! Initialization:

work = available
work = (0, 0, 0)

Finish (P_0 - P_4) = False
as all processes are allocated some resources

1) $i = P_0$; Request $(P_0] \leq$ work
$(0, 0, 0) \leq (0, 0, 0)$
work = work + allocation $[P_0]$
$= (0, 0, 0) + (0, 1, 0)$
$= (0, 1, 0)$, Finish $[P_0]$ = True

2) $i = P_1$; $(2, 0, 2) \nleq (0, 1, 0)$

3) $i = P_2$; $(0, 0, 0) \leq (0, 1, 0)$

$work = (0, 1, 0) + (3, 0, 3)$

$= (3, 1, 3)$, $Finish[P_2] = True$

4) $i = P_3$; $(1, 0, 0) \leq (3, 1, 3)$

$work = (3, 1, 3) + (2, 1, 1)$

$= (5, 2, 4)$, $finish[P_3] = True$

5) $i = P_4$; $(0, 0, 2) \leq (5, 2, 4)$

$work = (5, 2, 4) + (0, 0, 2)$

$= (5, 2, 6)$

6) $i = P_1$; $(2, 0, 2) \leq (5, 2, 6)$

$work = (5, 2, 6) + (2, 0, 0)$

$work = (7, 2, 6)$ $Finish[P_1] = True$

Since $Finish[P_0 - P_4] = true$, s/m is in safe state when the requested resources are allocated.

safe sequence $\langle P_0, P_2, P_3, P_4, P_1 \rangle$

Case II P2 requests additional instance of Type C.

Snapshot at time T0:

	Allocation	Request	Available
	A B C	A B C	A B C
P0	0 1 0	0 0 0	0 0 0
P1	2 0 0	2 0 2	
P2	3 0 3	**0 0 1**	
P3	2 1 1	1 0 0	
P4	0 0 2	0 0 2	

check if s/m is safe if the requested resources are allocated.

Sol^n^: 1) $i = P_0$: $request[P_0] \leq \underset{(0,0,0)}{work}$

$(0,0,0) \leq (0,0,0)$

$work = (0,1,0)$, $Finish[P_0] = true$

2) $i = P_1$: $(2,0,2) \nleq (0,1,0)$, false

3) $i = P_2$: $(0,0,1) \nleq (0,1,0)$, false

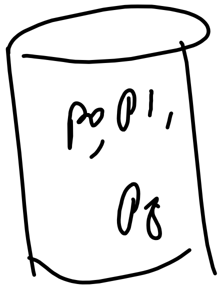
4) $i = P_3$: $(1,0,0) \nleq (0,1,0)$, false

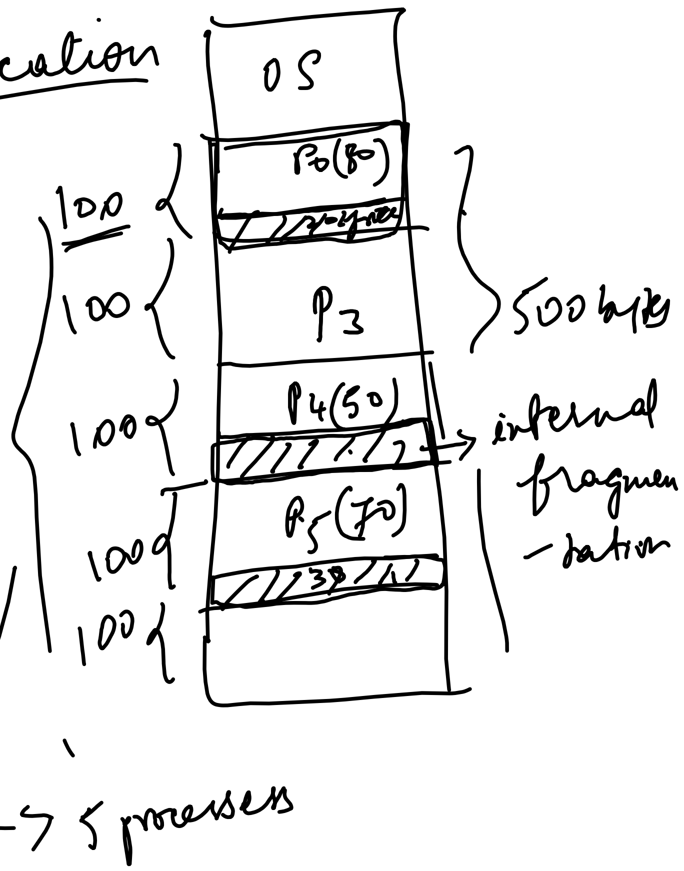
5) $i = P_4$: $(0,0,2) \nleq (0,1,0)$, false

$\langle P_1, P_2, P_3, P_4 \rangle$ deadlocked processes

Contiguous memory allocation

↳ fixed sized





$P_0 = 80$

$P_1 = 150$

$P_3 = 100$

$P_4 = 50$

Variable sized locations

* hole – available / unused / free memory space

P_1 – 100

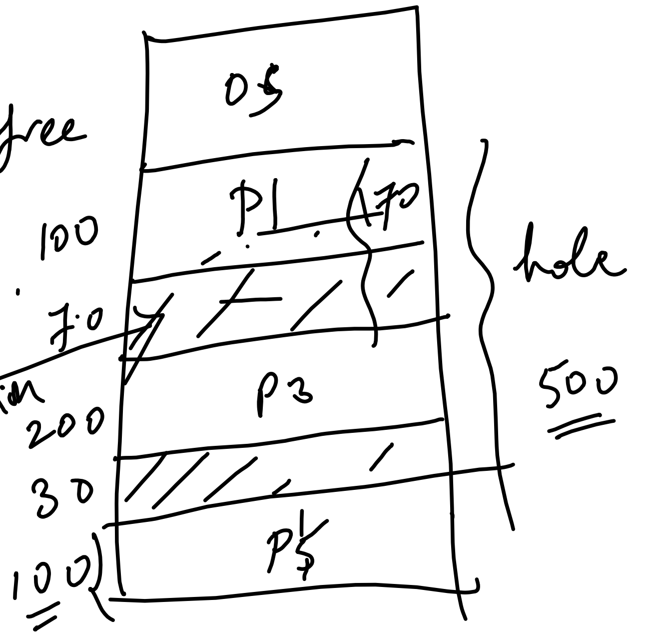
P_2 – 70

P_3 – 200

P_4 – 30

P_7 – 120

external fragmentation



P_6 – 100 byte – waits

Limit Register (301)

Relocation Register (14500)

process - 0 to 300

14500 + 0 ≤ PA process <= 14500 + 300

14800 ≤ ≤ 14800

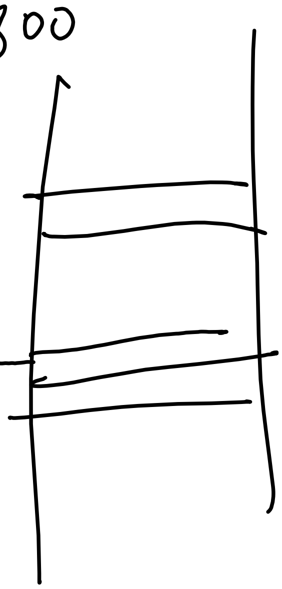
LA = 400 > 301 — trap 14500

LA = 200 < 301

14500 + 200

14700

14800



Paging — VA/LA → PA

Address space

$4k = 4096 = 2^{12}$

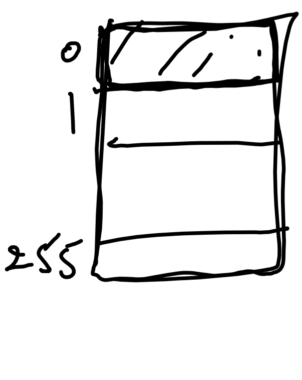
$12 - 4 = 8$ 4

Pg no. (8) | offset (4)

0

1

255



m bits = 12 bits ⇒

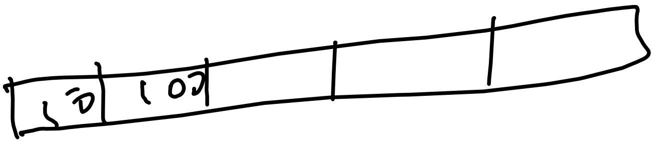
$2^4 = 16$ bytes

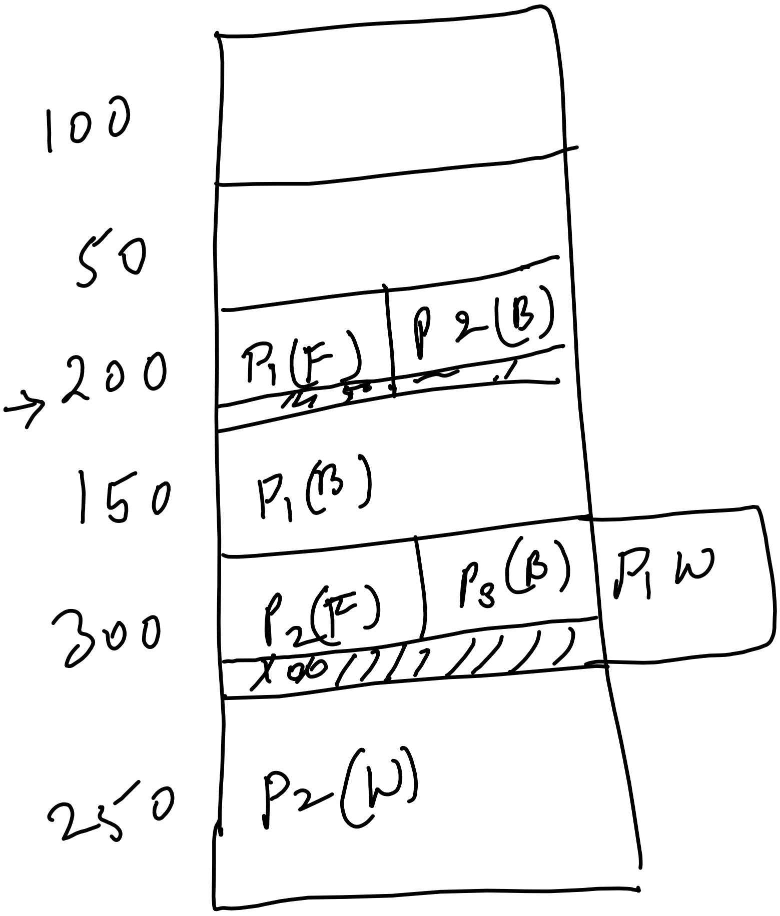
2^8 pages ⇒ 256

000 - - 0 — 0

111 - 1 — 255

1) First fit
Best
Worst fit





$P_1 = 150$

$P_2 = 200$

$P_3 = 300$ (wait)

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

$2^4 = 16$

$2^5 = 32$

$2^6 = 64$

$2^7 = 128$

$2^8 = 256$

$2^9 = 512$

$2^{12} = 4kB$ $\quad 2^{22} = 4MB$ $\quad 2^{32} = GB$

$2^{15} = 32k$ $\quad 2^{25} = 32M$