HARSH CHHIKARA
2K18CSUN01019
B.Tech CSE 4A

24/03/2020

DATE
PAGE

OS-T2

Q1-

(a)

(i) First come first serve
(ii) Multilevel queue
(iii) Shortest Job first
(iv) Shortest remaining time first.
(v) Round Robin

(b)

(i) Aging → As time moves priority of process increases.

(ii) Semaphore → It ensures that only those process which won't interfere with each other enter the critical section.

(iii) Mutual exclusion → It is not required for sharing resources but must be allowed for non-sharing resources.

(iv) The model requires that each process must declare the maximum no. of resources that it is required. This is Banker's Algorithm.

(v) To prevent race condition, all concurrent process need to be synchronised.

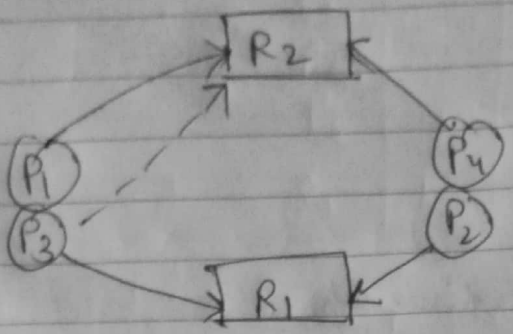
(2)

HARSH CHHIKARA
2K18CSUN01019
BTech CSE-4A

DATE
PAGE

Q2-

(a)



If R_2 is given to P_3, then either P_1 or P_4 may have to be free the allocation, the system is in safe state, since no cycle is formed.

(b)

Process	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P_0	2	0	0	1	4	2	1	2	3	3	2	1
P_1	3	1	2	1	5	2	5	2				
P_2	2	1	0	3	2	3	1	6				
P_3	1	3	1	2	1	4	2	4				
P_4	1	4	3	2	3	6	6	5				

a) Need (Max - allocation)

	A	B	C	D
P_0	2	2	1	1
P_1	2	1	3	1
P_2	0	2	1	3
P_3	0	1	1	2
P_4	2	2	3	3

HARSH CHHIKARA
2K18/CO/NO1019
BTECH CSE 4A

b) P_0 Need = (2, 2, 1, 1)
Available = (3, 3, 2, 1)
Need ≤ Available, New available = (5, 3, 2, 2)

P_1 Need = (2, 1, 3, 1)
Available = (5, 3, 2, 2)
Need < Available, New available = (8, 4, 4, 3)

P_2 Need = (0, 2, 1, 3)
Available = (8, 4, 4, 3)
Need < Available, New Available = (10, 5, 4, 6)

P_3 Need = (0, 1, 1, 2)
Available = (10, 5, 4, 6)
Need < Available, New Available = (11, 8, 5, 8)

P_4 Need = (2, 2, 3, 3)
Available = (11, 8, 5, 8)
Need < Available, New available = (12, 12, 8, 10)

Thereee it is in safe state
Therefore, Safe ~~Require~~ is sequence is P_0, P_1, P_2, P_3, P_4

c) Request$_1$ = (1, 1, 0, 0)

Request$_1$ <= Need
Request$_1$ <= Available

New available = Available - Request
= (2, 2, 2, 1)

HARSH CHHIKARA
2K18CSUN01019
BTech CSE4A

(4)

New allocation =

	A	B	C	D
P_0	2	0	0	1
P_1	4	2	2	1
P_2	2	1	0	3
P_3	1	3	1	2
P_4	1	4	3	2

New need =

	A	B	C	D
P_0	2	2	1	1
P_1	1	0	3	1
P_2	0	2	1	3
P_3	0	1	1	2
P_4	2	2	3	3

P_0 = need = (2, 2, 1, 1)
available = (2, 2, 2, 1)

need < available = new available = (4, 2, 1, 2)

P_1 = need = (1, 0, 3, 1)
available = (4, 2, 1, 2)

need < available, new available = (5, 2, 4, 3)

P_2 = need = (0, 2, 1, 3)
available = (5, 2, 4, 3)
need < available
available = (7, 3, 4, 6)

P_3 = need = (0, 1, 1, 2)
available = (7, 3, 4, 6)
need < available

HARSH CHHIKARA
21218CSUN01019
BTech CSE MA

DATE
PAGE

new available = (8, 6, 5, 8)

P_4 = need = (2, 2, 3, 3)
available = (8, 6, 5, 8)
need < available
new available (9, 10, 8, 10)

Sequence → P_0, P_1, P_2, P_3, P_4

Request will be granted

(d) $Request_4$ = (0, 0, 2, 0)

Request <= need
Request <= Available

new available = available − request
= (3, 3, 0, 1)

New allocation =

	A	B	C	D
P_0	2	0	0	1
P_1	3	1	2	1
P_2	2	1	0	3
P_3	1	3	1	2
P_4	1	4	5	2

new need =

	A	B	C	D
P_0	2	2	1	1
P_1	2	1	3	1
P_2	0	2	1	3
P_3	0	1	1	2
P_4	2	2	1	3

(6)

P_0 need = (2, 2, 1, 1)
available = (3, 3, 0, 1)
need < available
new available = (5, 3, 0, 2)

P_1 need = (2, 1, 3, 1)
available = (5, 3, 0, 2)

need < available
new available = (8, 4, 2, 3)

P_2 need = (0, 2, 1, 3)
available = (8, 4, 2, 3)
need < available
new available = (10, 5, 2, 6)

~~P_3 need = (0, 2, 1, 3)
available = (8, 4, 1, 3)
need < available
new available (10, 5, 2, 6)~~

P_3 need = (0, 1, 1, 2)
available = (10, 5, 2, 6)
need < available
new available = (11, 8, 3, 8)

P_4 need = (2, 2, 1, 3)
available = (11, 8, 3, 8)
need < available
new available = (12, 12, 2, 10)
sequence = P_0, P_1, P_2, P_3, P_4
Request will be granted.

HARSH CHANDRA
21218CJUWD1019
Btech CSE 4A

Q3 —

(a)

Process	AT	BT	CT	TT	WT
P1	0	8	14	14	9
P2	1	3	12	11	8
P3	2	1	5	3	2
P4	3	2	7	4	2
P5	4	3	13	9	6

P1	P2	P3	P4	P5	P1	P2	P5	P1
0–2	2–4	4–5	5–7	7–9	9–11	11–12	12–13	13–14

Average T.T = 8.2
Average W.T = 5.4

(C) S = 12
wait P = 10
Signal, V = 4

Find value of at counting semaphore = 12 − 10 + 4
= 6

(b) Total time spent = 47
idle time = 2 + 3 = 5

Percentage of idle time = (5/47) * 100 = 10.6 %