

## 31 # Resource Allocation Graph :-


⇒ Process →  $P$

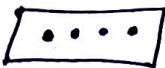
⇒ Resource →  $R$

Assign edge

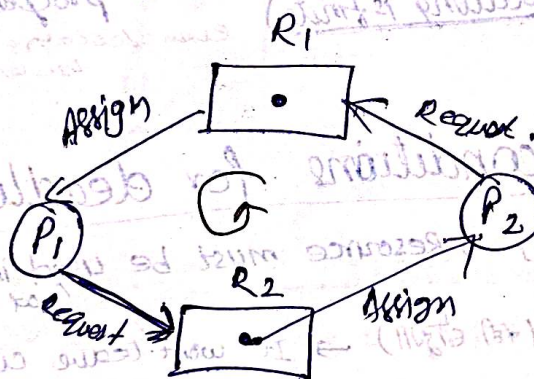
Request edge



⇒  → single instance e.g. CPU.

⇒  → Multi Instance

⇒ Single Instance (to check whether Deadlock exists or not)



$\left. \begin{array}{l} P_1 \text{ to } R_2 \text{ back} \\ \text{so circular} \\ \text{wait in this} \end{array} \right\}$

Here it's circular wait &  $P_1$  requires  $R_2$  which is possessed by  $P_2$  so deadlock.

⇒ we can find deadlock using Banker's Algorithm.

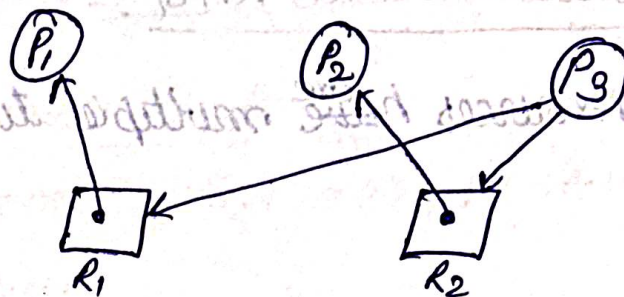
	Allocate		Request	
	$R_1$	$R_2$	$R_1$	$R_2$
$P_1$	1	0	0	1
$P_2$	0	1	1	0

Availability =  $(0, 0)$

It's neither full fill the request of  $R_1$  or  $R_2$  so it's deadlock situation.



## Example-2 :-



	Allocate		Request	
	R <sub>1</sub>	R <sub>2</sub>	R <sub>1</sub>	R <sub>2</sub>
P <sub>1</sub>	1	0	0	0
P <sub>2</sub>	0	1	0	0
P <sub>3</sub>	0	0	1	1

Availability = (0, 0)

Since both R<sub>1</sub> & R<sub>2</sub> occupied by P<sub>1</sub> & P<sub>2</sub> respectively.

∴ But P<sub>1</sub> requires (0, 0) → so P<sub>1</sub> terminates and its resource taken back (freed).

∴ New availability = (1, 0).

Also P<sub>2</sub> requires (0, 0) → so P<sub>2</sub> also terminates and its resources freed.

∴ New availability = (1, 1)

Now this new resource (1, 1) can fulfill P<sub>3</sub> (1, 1)

so P<sub>3</sub> also terminates.

sequence
P <sub>1</sub> → P <sub>2</sub> → P <sub>3</sub>

→ so there is no deadlock

⇒ If RAG has a circular wait (cycle) then always deadlock only when there is a single instance.

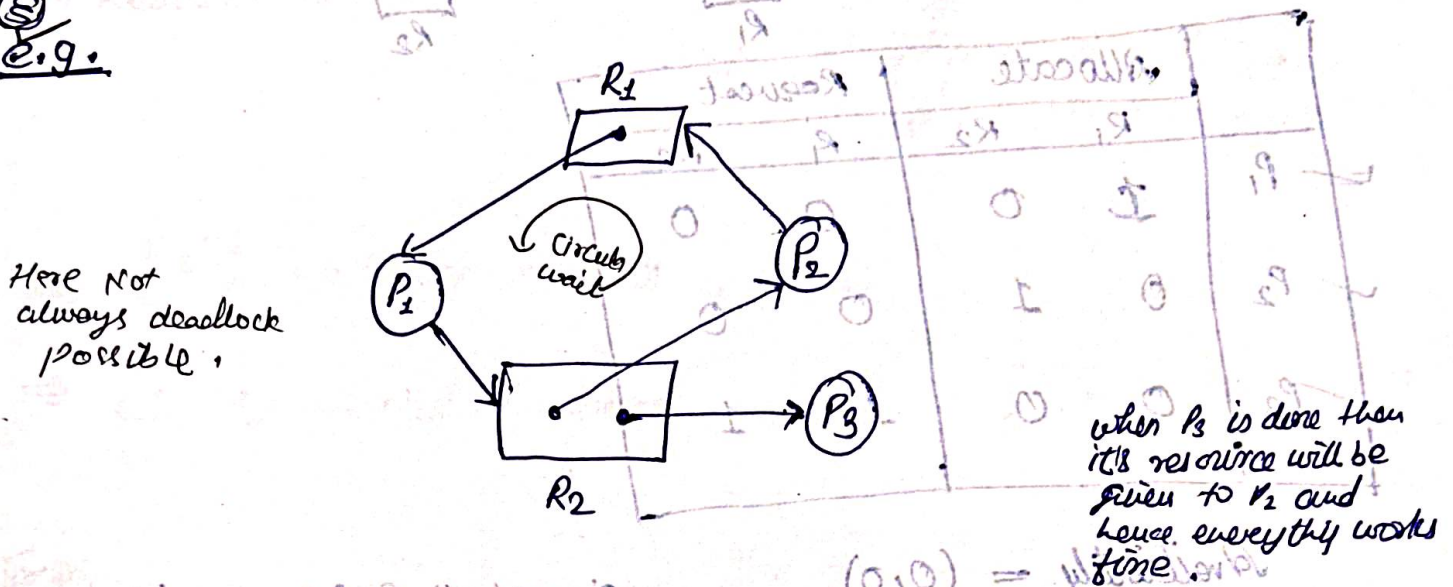
⇒ If RAG has no cycle then no deadlock for single instance condition



## 32) # Multi-Instance RAG :-

⇒ where resources have multiple times present.

E.g.



	Allocate		Request	
	$R_1$	$R_2$	$R_1$	$R_2$
$P_1$	0	0	0	1
$P_2$	0	1	1	0
$P_3$	0	1	0	0

Current Availability =  $(0, 0)$  →  $P_3$  will be terminated free  $(0, 1)$

∴ new Avail =  $(0, 1)$  →  $P_1$  executes free  $(1, 0)$

new Avail =  $(1, 0) + (0, 1) = (1, 1)$  →  $P_2$  executes

∴ New Avail  $(1, 2)$

$P_3 \rightarrow P_1 \rightarrow P_2$

so here cycle and still no deadlock in multi instance resource type.



