مشہود

علی حسن سندھو محمد مشہود حسنین علی qwerty410

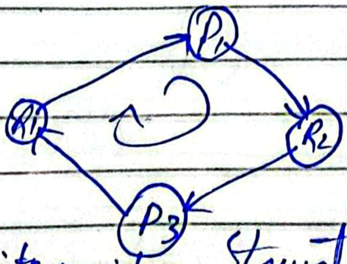
Test and set

```
bool testandSet (bool *target) {
    bool r = *target
    *target = true;
    return r;
}

bool lock=0
while (testandSet (&lock));
/C.S./
lock=0
```

Deadlock

اگر کوئی resources کسی event کا ویٹ کر رہے ہیں
تو کبھی occur نہیں ہو گا۔

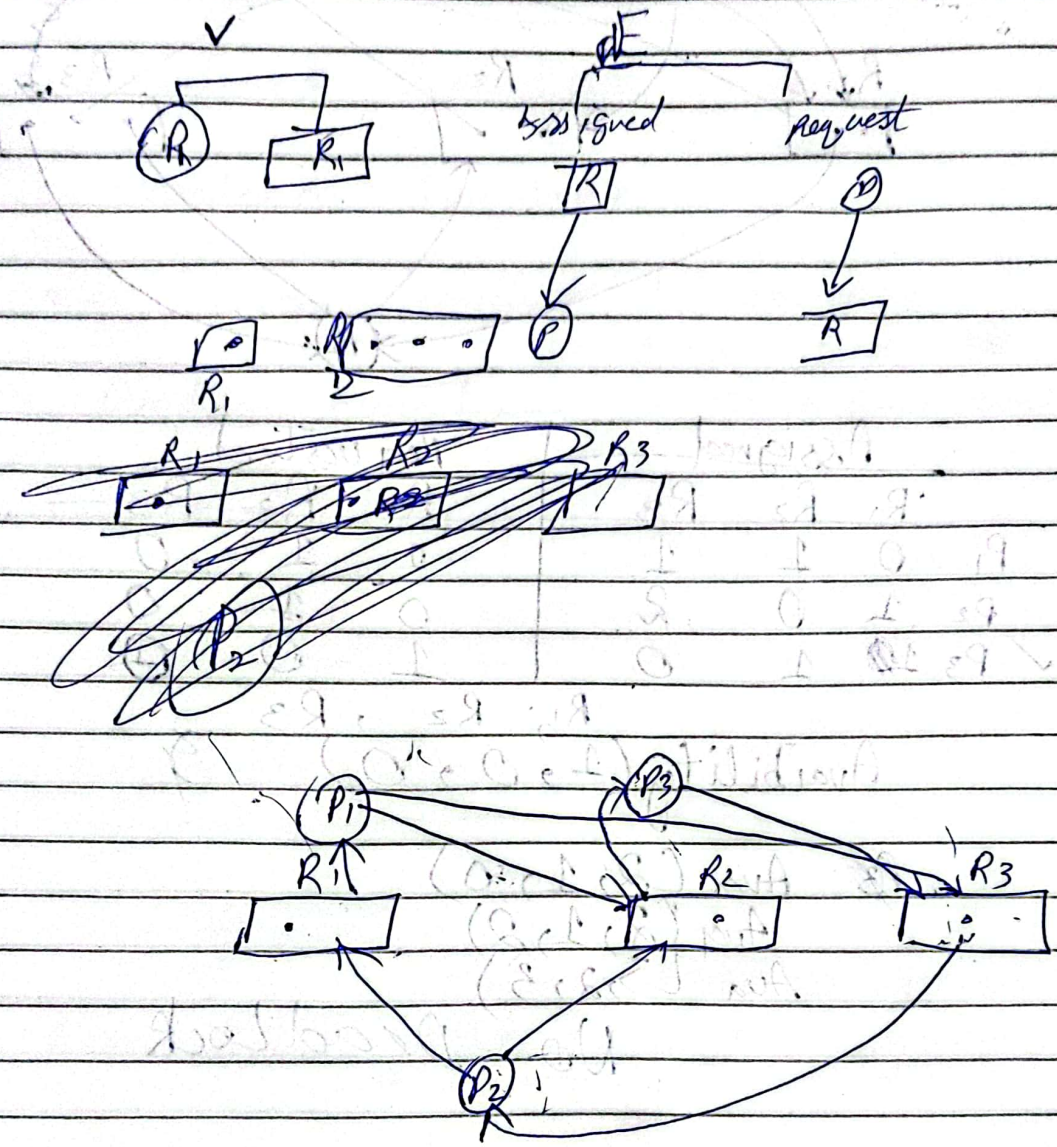


Finite wait = Starvation
infinite = Deadlock

1- Mutual Exclusion
2- No-preemptions
3- Hold and wait
4- Circular Wait

سرکلیٹ لگا ہوا ہے۔

Resource Allocation Graph (RAG)

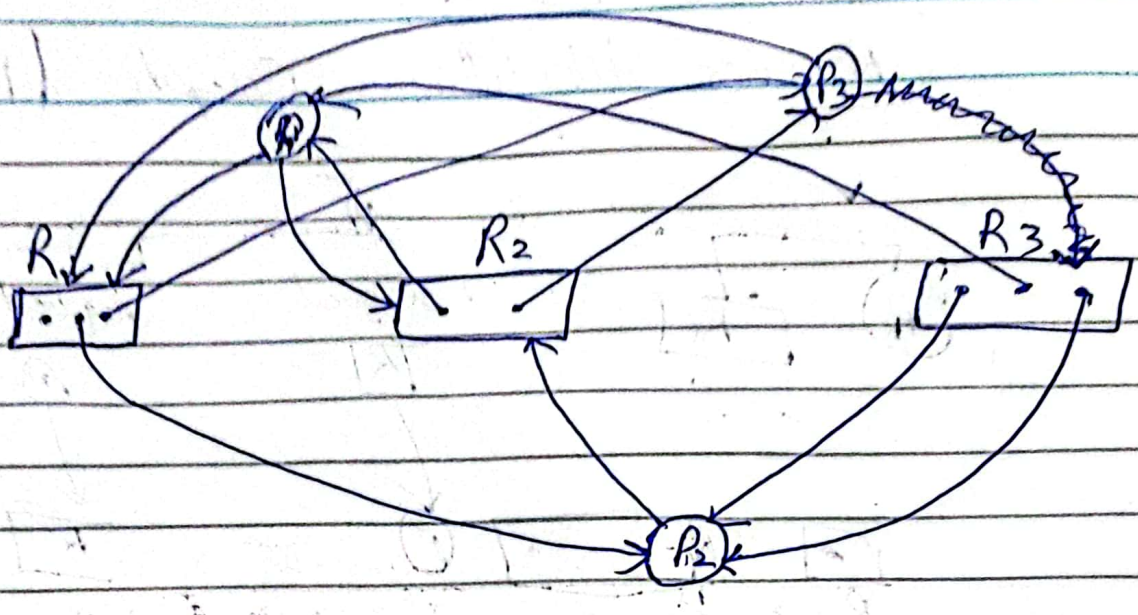


	Allocate R			Requested R		
	R1	R2	R3	R1	R2	R3
P1	1	0	0	0	1	0
P2	0	0	1	1	0	0
P3	0	1	0	0	0	1

R1, R2, R3

Availability (0,0,0)

Dead Lock



	Assigned			Requested		
	R_1	R_2	R_3	R_1	R_2	R_3
P_1	0	1	1	1	1	0
P_2	1	0	2	0	1	0
✓ P_3	1	1	0	1	0	1

R_1, R_2, R_3

Availability $(1, 0, 0)$

Ava $(2, 1, 0)$

Ava $(2, 1, 2)$

Ava $(3, 2, 3)$

No - Deadlock

روحان جاوید عبداللہ اصغر احمد ایاز یاسین منصور راحیل ناظیر
محمد عمر طارق محمد مشہود حسنین طلحہ عمران علی حسن حمزہ خان

Deadlock Handling

1- Deadlock Prevention

2- Deadlock Avoidance

3- Allowing deadlocks and recovering

Deadlock prevention

Four necessary cond for deadlock ★

1- Mutual Exclusion
2- Hold and wait
3- No preemption:-
4- Circular wait

Deadlock Avoidance

Safe State

Such state when escape from that system have a sequence.