A	B	C
10	5	7

	Allocation			Max			Need		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	5	3	7	4	3
P_1	2	0	0	3	2	2	1	2	2
P_2	3	0	2	9	0	2	6	0	0
P_3	2	1	1	2	2	2	0	1	1
P_4	0	0	2	4	3	3	4	3	1

n x m

Available

A	B	C
3	3	2
(10-7)	(5-2)	(7-5)

Asume P_1 can finish

P_1^* available

A	B	C				
3	3	2	→	P_1	2	0	0
+ 2	0	0	→	P_3	2	1	1
5	3	2					
+ 2	1	1					
7	4	3					

P_3^*

=> All can finish system is safe

P_1 Request 1 A unit ?

	Allocation A	B	C	Max A	B	C	Need A	B	C
P_0	0	1	0	7	5	3	7	4	3
P_1	3	0	0	3	2	2	0	2	2
P_2	3	0	2	9	0	2	6	0	0
P_3	2	1	1	2	2	2	0	1	1
P_4	0	0	2	4	3	3	4	3	1

nxm

Avelible

A B C => safe grant

2 3 2

P_1 Request 2 additional units of type C

Allocation	Max	Need
3 0 2	3 2 2	0 2 0

Avelible still safe

2 3 0

A B C

P_4 request (3, 3, 0)

Allocation Max Need

P_4 3 3 2 4 33 1 0 1

Avelible

2 3 0 => Deny request

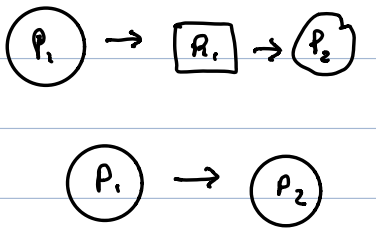
3 Detection and Recovery

a Single instant of resource

- wait for graph

If a cycle exists

=> System is deadlocked



b Multiple instants

1. Work = Avelible , Finish[i] = False
2. Find process i s.t.

   finish[i] == False ;

   request i ≤ Work

   If no such i, go to 4

3. Work = work + allocation i ;

   Finish[i] = True ;

go to 2

4 If finish == false for some (i)
these are deadlocked

Project 2

1 Game

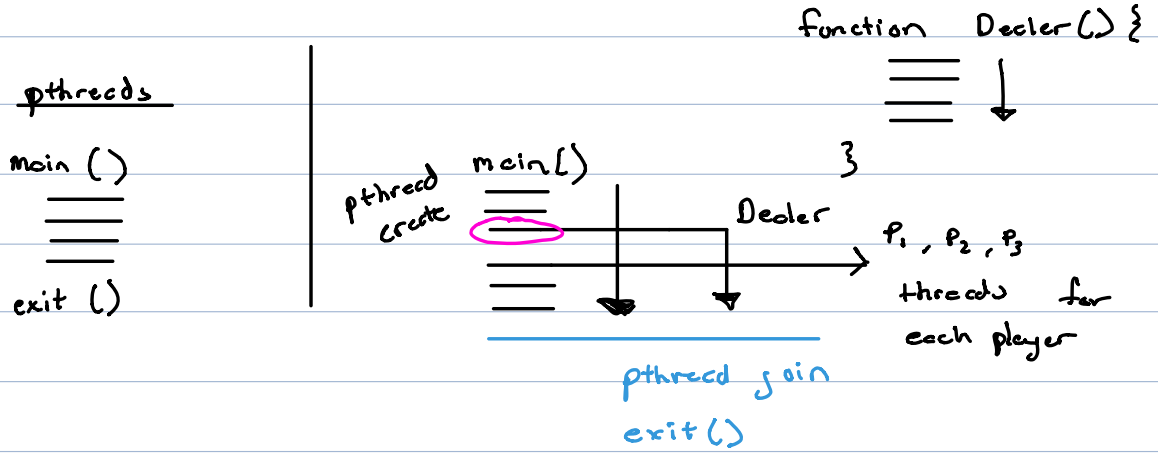
3 rounds

D P_1 P_2 P_3

Round 1 — 1 card each

Shuffle cards

- Checking to see if highest card is present



- Dealers run first
  -> wait for P_1
- Must lock & unlock resources.

→ wait for P_2 • Requires Pthread join

....

Mutex

```
pthread_mutex_t handler;
pthread_mutex_init();
```

⟶ handler.lock();

⟶ handler.unlock();

Condition Variables

```
pthread_condition_t y;
pthread_condition_wait ( condition, mutex)
pthread_condition_signal ( condition);
```

```
pthread_mutex_lock (&x);
pthread_cond_wait (&y, &x)
pthread_mutex_unlock (&x)
        :
pthread_mutex_lock (&x)
pthread_cond_signal (&y)
pthread_mutex_unlock (&x)
```

lock i.e. P_2

wait → condition is unlock but the signal locks it.

2 layers unlocking

Player function (pass number of players)
2nd argument would tell you if it finished properly