**Q : Draw the Petri-Net for Banker's algorithm of operating system. Describe the elements of the Petri-net in the context of operating system.**

**Ans :** Petri Nets: Petri-nets are a graphical and mathematical modelling tool applicable to study the dynamics of various systems.

A Petri-net is five tuple system can be defined as: PN= (P, T, F, W, $M_0$) where

P = { P1 , P2, P3, …, Pm } = Finite set of places

T = { t1, t2, t3, …, tn } = Finite set of transitions

F ⊆ (P x T) U (T x P) is a set of arcs (flow relation)

W: F → {1, 2, 3, …} is a weight function

$M_0$: P → {0, 1, 2, 3, …} is the initial marking

Banker's Algorithm: Banker's Algorithm is a resource allocation and deadlock avoidance algorithm. It is developed by Dijkstra. Deadlock avoidance ensures that a system will never enter an unsafe state. A system is in unsafe state if there is a possibility of deadlock.

The Banker's algorithm is actually made up of two separate algorithms:

1. Safety Algorithm

2. Resource Allocation Algorithm

Let's take an example of Banker's Algorithm :

We have 3 resources and 4 processes, and we must get all 4 processes fired so that the system stays in a stable state. To keep the system from a deadlock we need to make sure we fire the correct processes sequence, so that the resources assigned and available meet the resources needs. If you fire a process and the resources are not met, then your system will become deadlock and will most likely become in an unstable state.

4 processes P0 through P3; 3 resource types A (7 instances), B (9 instances), and C (10 instances)

Initial State:

| Processes | Allocation | Maximum | Need | Available |
|---|---|---|---|---|
| | <A,B,C> | <A,B,C> | <A,B,C> | <A,B,C> |
| P0 | <0,3,4> | <6,5,4> | <6,2,0> | <4,3,1> |
| P1 | <2,1,2> | <3,4,2> | <1,3,0> | |
| P2 | <0,0,2> | <1,0,4> | <1,0,2> | |
| P3 | <1,2,1> | <3,2,5> | <2,0,4> | |

Step 1: First complete the process P1 as it satisfies the need and after the processes complete it will release the resource. Safe state = <P1>

| Processes | Allocation | Maximum | Need | Available |
|---|---|---|---|---|
| | <A,B,C> | <A,B,C> | <A,B,C> | <A,B,C> |
| P0 | <0,3,4> | <6,5,4> | <6,2,0> | <6,4,3> |
| P1 | <0,0,0> | <3,4,2> | <0,0,0> | |
| P2 | <0,0,2> | <1,0,4> | <1,0,2> | |
| P3 | <1,2,1> | <3,2,5> | <2,0,4> | |

Step 2: Then we can execute the process P0 as its need can be satisfied by available resources. Safe State = <P1,P0>

| Processes | Allocation | Maximum | Need | Available |
|---|---|---|---|---|
| | <A,B,C> | <A,B,C> | <A,B,C> | <A,B,C> |
| P0 | <0,0,0> | <6,5,4> | <0,0,0> | <6,7,7> |
| P1 | <0,0,0> | <3,4,2> | <0,0,0> | |
| P2 | <0,0,2> | <1,0,4> | <1,0,2> | |
| P3 | <1,2,1> | <3,2,5> | <2,0,4> | |

Step 3 : Then we can execute the process P2 because its need can be clearly satisfied by the available resources. Safe state = <P1,P0,P2>

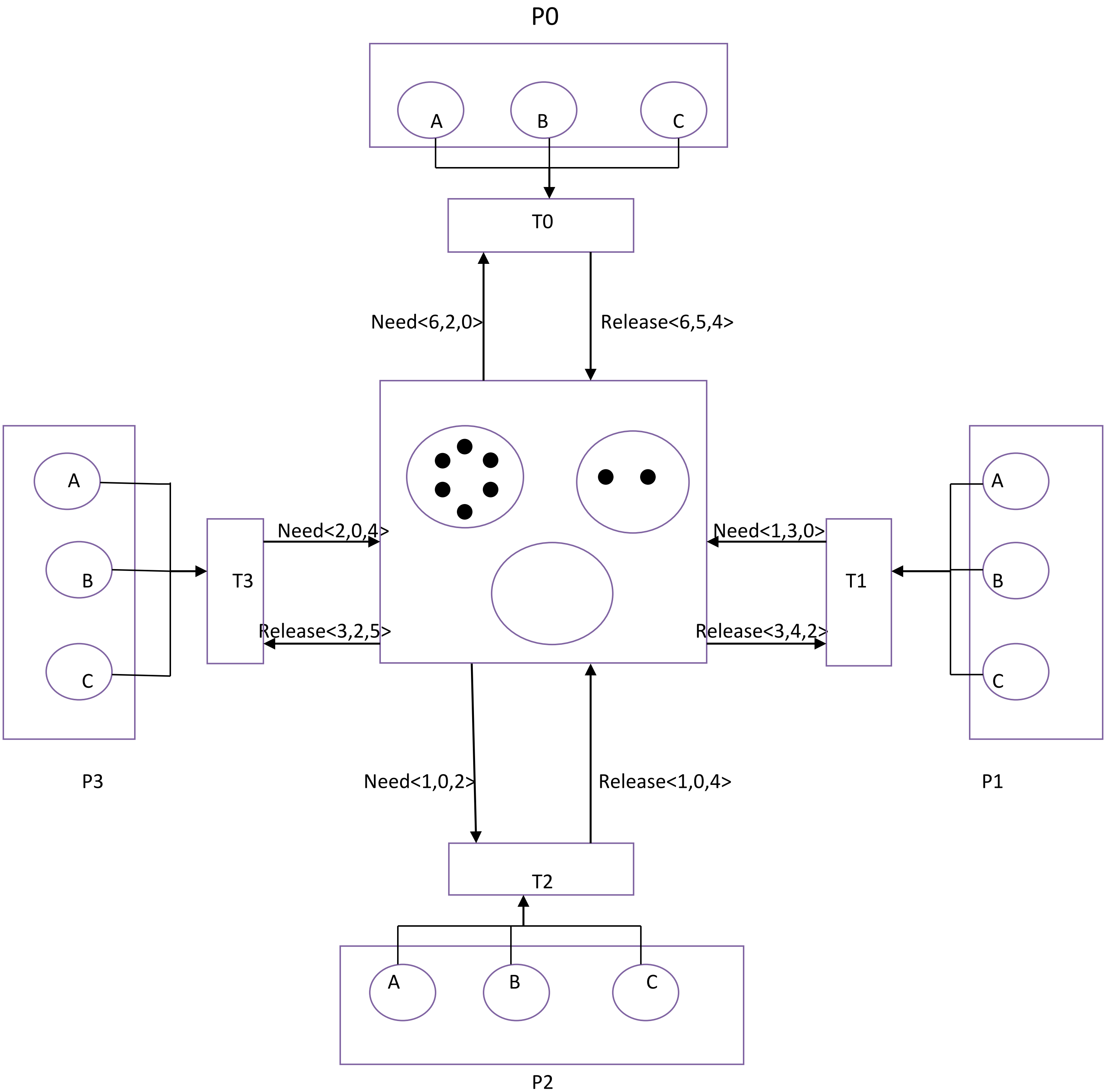
| Processes | Allocation | Maximum | Need | Available |
|---|---|---|---|---|
| | <A,B,C> | <A,B,C> | <A,B,C> | <A,B,C> |
| P0 | <0,0,0> | <6,5,4> | <0,0,0> | <6,7,9> |
| P1 | <0,0,0> | <3,4,2> | <0,0,0> | |
| P2 | <0,0,0> | <1,0,4> | <0,0,0> | |
| P3 | <1,2,1> | <3,2,5> | <2,0,4> | |

Step 4 : Now we can execute the process P3 because its need can be clearly satisfied by the available resources. Safe state = <P1,P0,P2,P3>

| Processes | Allocation | Maximum | Need | Available |
|---|---|---|---|---|
| | <A,B,C> | <A,B,C> | <A,B,C> | <A,B,C> |
| P0 | <0,0,0> | <6,5,4> | <0,0,0> | <7,9,10> |
| P1 | <0,0,0> | <3,4,2> | <0,0,0> | |
| P2 | <0,0,0> | <1,0,4> | <0,0,0> | |
| P3 | <0,0,0> | <3,2,5> | <0,0,0> | |

Conclusion: Here we can see that with the given initial state there will be a safe state possible i.e. Now we will implement the above Banker's Algorithm example using Petri-net.

Here, the Petri-net elements are: -
P = {P0, P1, P2, P3} is Allocated Resources to process.
T = {T0,T1, T2, T3} is Max Condition needs to be satisfied to complete execution
F = All the needed and released arcs (Flow Relation)
W = Resource needed and released mentioned in the flow
$M_0$ = R (Resource Available)



P0
A
B
C
T0
Need<6,2,0>
Release<6,5,4>
A
B
C
P3
T3
Need<2,0,4>
Release<3,2,5>
Need<1,3,0>
Release<3,4,2>
T1
A
B
C
P1
Need<1,0,2>
Release<1,0,4>
T2
A
B
C
P2