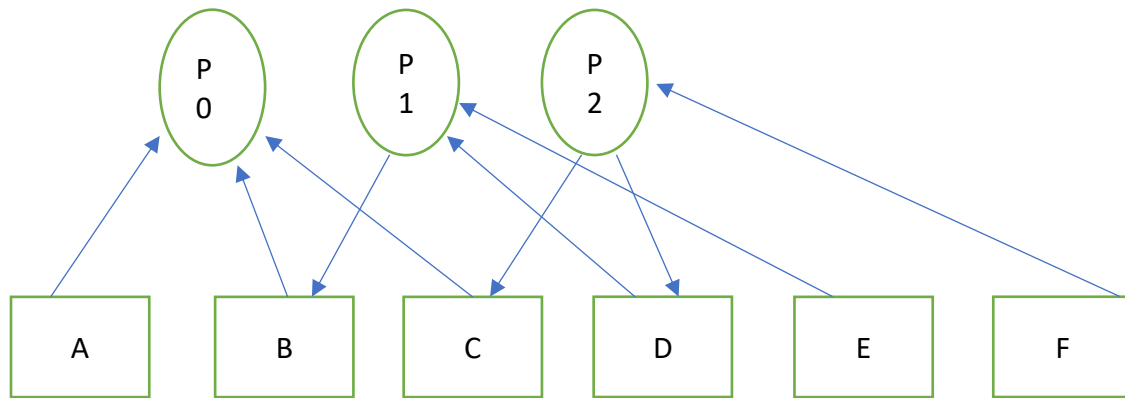
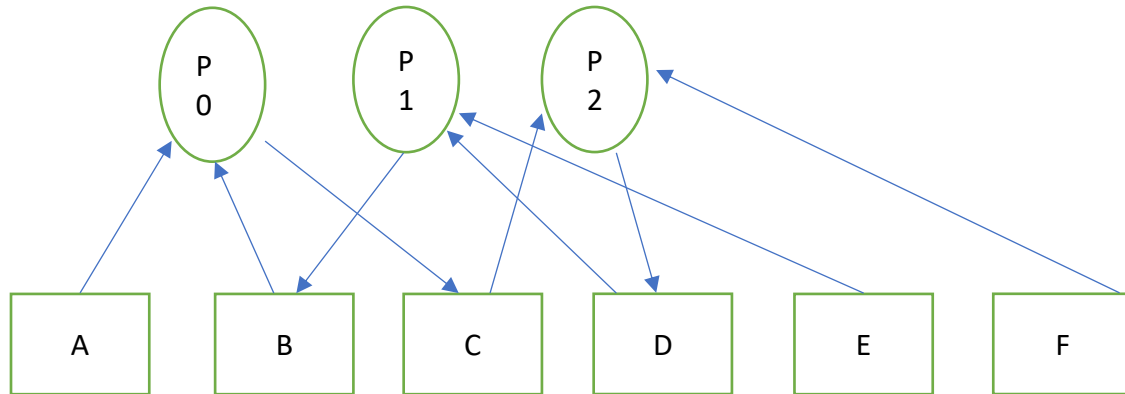
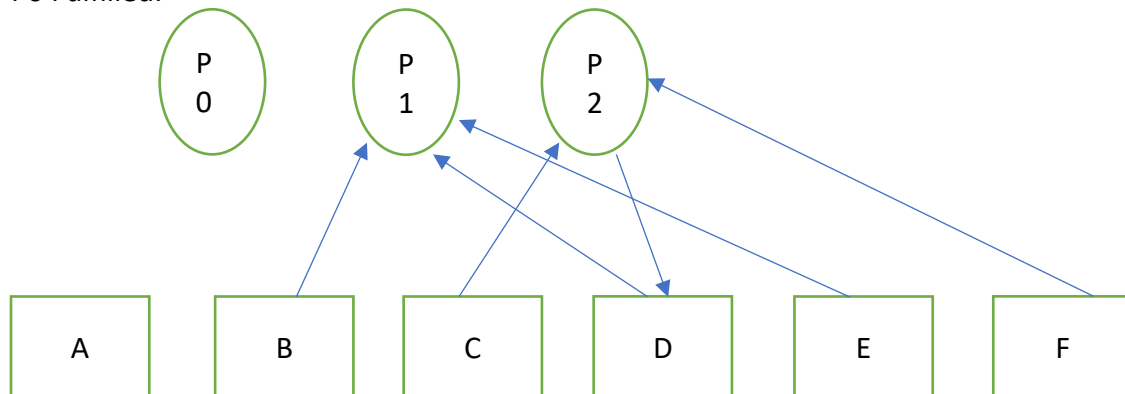


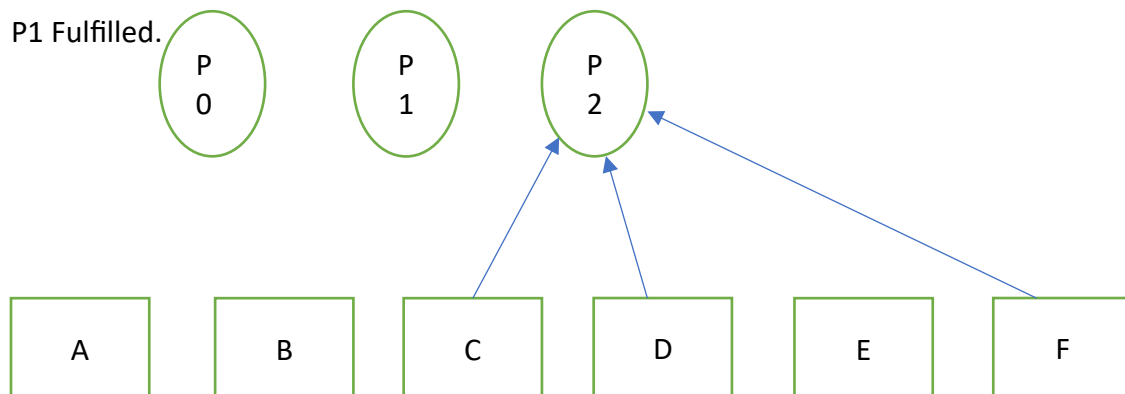
1.



P0 Fulfilled.



P1 Fulfilled.



2.

yes, both foo and bar can be blocked forever, because both needs a certain permission (S and R) to continue but they end up waiting for each other in the loop.

First foo will start and takes some permissions (S and R), increasing a value (x) by 1, then bar will also start but needs the same permissions that foo has, and since foo has them, bar gets stuck and can't continue, foo will keep doing its job, repeatedly taking and releasing permissions, and bar remains stuck because it can't get the permissions that foo keeps taking and releasing, and this cycle can continue forever, with neither foo nor bar making any progress. they are both blocked.

3.

Deadlocks Avoidance Constrains resource requests to prevent at least one of the four conditions of deadlock; this is either done indirectly, by preventing one of the three necessary policy conditions (mutual exclusion, hold and wait, no preemption), or directly, by preventing circular wait.

- is often impossible to implement.
- The system requires additional a priori information regarding the overall potential use of each resource for each process.
- Needs to be manipulated until at least one safe path is found.
- There is no preemption.

Deadlock detection Allows the three necessary conditions but makes judicious choices to assure that the deadlock point is never reached.

- The system does not require additional a priori information regarding the overall potential use of each resource for each process in all cases.
- For the system to detect the deadlock condition it does not need to know all the details of all resources in existence, available and requested.
- Needs to be invoked periodically to test for deadlock.
- Preemption is seen.

Deadlock Prevention, the goal is to ensure that at least one of the necessary conditions for deadlock can never hold.

- is often impossible to implement.
- The system does not require additional a priori information regarding the overall potential use of each resource for each process.
- All resources are requested at once.
- In some cases, preempts more than often necessary.

4.

by contradiction, suppose the system is deadlocked.

This implies that each process is holding one resource and is waiting for one more. Since there are three processes and four resources; one process must be able to obtain two resources. This process requires no more resources and therefore it will return its resources when done.