

## **Deadlock definition**

Processes or threads are blocked, waiting for resource that others held.

Need access to resources in certain order.

## **Condition**

Mutual exclusion

Only one process at a time can use a resource

Hold and wait

Process holding at least one resource is waiting to acquire additional resources held by other processes

No preemption

Resource can be released only voluntarily by the process holding it, after that process has completed its task

Circular wait

There must exist a set  $\{P_0, P_1, \dots, P_n\}$  of waiting processes such that  $P_0$  is waiting for a resource that is held by  $P_1$ ,  $P_1$  is waiting for a resource held by  $P_2$

## **Prevention**

Mutual exclusion

Not required for sharable resources, but must hold for non sharable resources

Hold and wait

Must guarantee that whenever process requests resource, it does not hold any other resources

-Require process to request and be allocated all its resources before it begins execution

-Allow process to request resources only when the process has none

Problems

-May not know required resources at start of run

-Low resource utilization

-Starvation possible

No preemption

-If process must wait for another resource, all resources currently being held are implicitly preempted

-If requesting resources are allocated to some other process that is waiting for additional resources, preempt the desired resources from waiting process

Circular wait

- Impose total ordering of all resource types

- Require that each process requests resources in an increasing order of enumeration