Operating System Deadlocks

# What Is In This Chapter?

* What is a deadlock?
* Staying Safe: Preventing and Avoiding Deadlocks
* Living Dangerously: Let the deadlock happen, then detect it and recover from it.

DEADLOCKS

## EXAMPLES:

* "It takes money to make money".
* You can't get a job without experience; you can't get experience without a job.

## BACKGROUND:

The cause of deadlocks: Each process needing what another process has. This results from sharing resources such as memory, devices, and links.

Under normal operation, a resource allocations proceed like this:

1. Request a resource (suspend until available if necessary).
2. Use the resource.
3. Release the resource.

## Bridge Crossing Example:



* Traffic only in one direction.
* Each section of a bridge can be viewed as a resource.
* If a deadlock occurs, it can be resolved if one car backs up (Preempt resources and rollback).
* Several cars may have to be backed up if a deadlock occurs.
* Starvation is possible.

## NECESSARY CONDITIONS

**ALL** of these four **must** happen simultaneously for a deadlock to occur:

* + **Mutual exclusion:**

One or more than one resource must be held by a process in a non-sharable (Exclusive) mode.

* + **Hold and Wait**

A process holds a resource while waiting for another resource.

* + **No Preemption**

There is only voluntary release of a resource - nobody else can make a process give up a resource.

* + **Circular Wait**

Process A waits for Process B waits for Process C waits for Process A.



















## Follow the video link please

## https://ocw.vu.edu.pk/Videos.aspx?cat=Computer+Science%2fInformation+Technology+&course=CS604