Deadlock detection

Multiple processes
Multiple resources
each have single instance

$$V_1 = \{P_1, P_2, \ldots \}$$

Resource - allocation graph

wait-for graph contains only processes as vertices.
$P_i \rightarrow P_j$ only if $P_i \rightarrow R_k$ and $R_k \rightarrow P_j$
P_i waits for resources allocated to P_j.

wait-for graph

If there is a cycle in the wait for graph, then there is a
deadlock.

	Allocation A B C	Request A B C	Available A B C
① P₀ ✓	0 1 0	0 0 0	0 0 0
② P₁ ✓	2 0 0	2 0 2	+ 0 1 0
③ P₂ ✓	3 0 3	0 0 0	0 1 0
④ P₃ ✓	2 1 1	1 0 0	+ 8 0 3
⑤ P₄ ✓	0 0 2	0 0 2	8 1 3

$$\text{Finish}_i = \text{false}$$
$$\text{If Request}_i \leq \text{Available}$$
$$\text{Available} += \text{Allocation}$$
$$\text{Finish}_i = \text{True}$$

Available column additions:
$$+ 2\ 1\ 1$$
$$5\ 24$$
$$+ 0\ 0\ 2$$
$$5\ 2\ 6$$
$$+ 2\ 0\ 0$$
$$7\ 2\ 6$$

Safe sequence
$$< P_0, P_2, P_3, P_4, P_1 >$$

* If any request from any process is not fulfilled and it is not included in the safe sequence, then it is a deadlock sequence situation.

Deadlock recovery

* Process termination ⟵ Abort all deadlock processes
* Resource preemption ⟵ Abort one process at a time until the deadlock cycle is broken.

Resource preemption - we successively preempt some resources from processes and give these resources to other processes until the deadlock cycle is broken

Three issues with resource preemption:

1) Select a victim - which resources and which processes are to be preempted? We also need to determine order of preemption to minimize cost.

2) Rollback - If we preempt a resource from a process, what should be done with that process? Simplest solution is a total rollback: abort the process and restart it

3) Starvation - How do we ensure starvation will not occur? That is, how can we guarantee that resources will not always be preempted from the same process? The most common sol² is to include the no. of rollbacks in the cost factor.