SNS COLLEGE OF TECHNOLOGY

Coimbatore-35.
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

COURSE NAME : OPERATING SYSTEMS

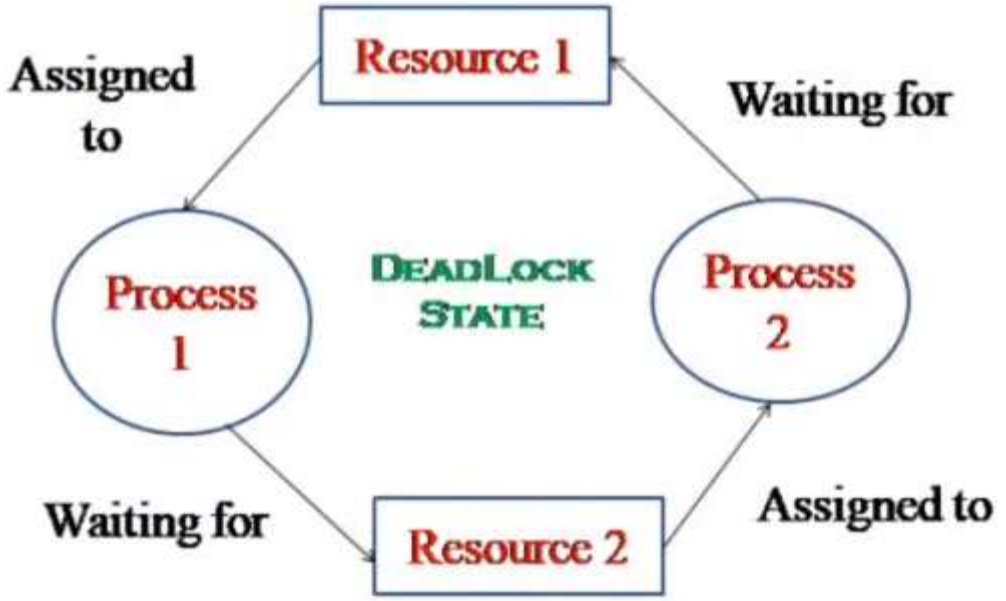
II YEAR/ IV SEMESTER

UNIT – II PROCESS SCHEDULING AND SYNCHRONIZATION

Topic: Methods for handling deadlocks – Deadlock prevention

Dr.B.Vinodhini
Associate Professor
Department of Computer Science and Engineering

Dead Lock



Dead lock is situation where a set of processes are blocked because each process is holding a resource and waiting for another resource and waiting for another resource acquired by some other process

Dead Lock Handling Methods

- **Dead Lock Ignorance-** Ignore the problem and pretend that deadlocks never occur in the system; used by most operating systems, including UNIX
- Ensure that the system will never enter a deadlock state:
  - **Dead Lock Prevention**
  - **Dead Lock Avoidance**
- **Dead Lock Detection & Recovery** - Allow the system to enter a deadlock state and then recover

Dead Lock Handling Methods
Dead lock Prevention

Mutual Exclusion – not required for sharable resources (e.g., read-only files); must hold for non-sharable resources

Hold and Wait – must guarantee that whenever a process requests a resource, it does not hold any other resources

- **Require process to request and be allocated all its resources before it begins execution**, or allow process to request resources only when the process has none allocated to it.
- **Low resource utilization; starvation possible**

No Preemption –

- If a process that is holding some resources requests another resource that cannot be immediately allocated to it, then all resources currently being held are released
- Preempted resources are added to the list of resources for which the process is waiting
- Process will be restarted only when it can regain its old resources, as well as the new ones that it is requesting

Circular Wait – impose a total ordering of all resource types, and require that each process requests resources in an increasing order of enumeration

References

1. Silberschatz, Galvin, and Gagne, “Operating System Concepts”, Ninth Edition, Wiley India Pvt Ltd, 2009.
2 . Andrew S. Tanenbaum, “Modern Operating Systems”, Fourth Edition, Pearson Education, 2010.


Thank
you