

---

# Difference Between Deadlock and Starvation in OS

January 11, 2017 — Leave a Comment



Deadlock and Starvation both are the conditions where the processes requesting for a resource has been delayed for a long. Although deadlock and starvation both are different from each other in many aspects. **Deadlock** is a condition where no process proceeds for execution, and each waits for resources that have been acquired by the other processes. On the other hands, in **Starvation**, process with high priorities continuously uses the resources preventing low priority process to acquire the resources. Let us discuss some more differences between deadlock and starvation with the help of comparison chart shown below.



Hello English: Learn English

FREE  Google Play

★★★★☆ (444,108)

Learn English from Nepali

Dictionary, Lessons, Game Master English Speaking



INSTALL



## Content: Deadlock Vs Starvation

### 1. Comparison Chart

- 2. Definition
- 3. Key Differences
- 4. Conclusion

## Comparison Chart

BASIS FOR COMPARSION	DEADLOCK	STARVATION
Basic	Deadlock is where no process proceeds, and get blocked.	Starvation is where low priority processes get blocked, and high priority process proceeds.
Arising condition	The occurrence of Mutual exclusion, Hold and wait, No preemption and Circular wait simultaneously.	Enforcement of priorities, uncontrolled resource management.
Other name	Circular wait.	Lifelock.
Resources	In deadlocked, requested resources are blocked by the other processes.	In starvation, the requested resources are continuously used by high priority processes.
Prevention	Avoiding mutual exclusion, hold and wait, and circular wait and allowing preemption.	Aging.



Hello English: Learn English

FREE  Google Play

★★★★☆ (444,108)

Learn English from Nepali

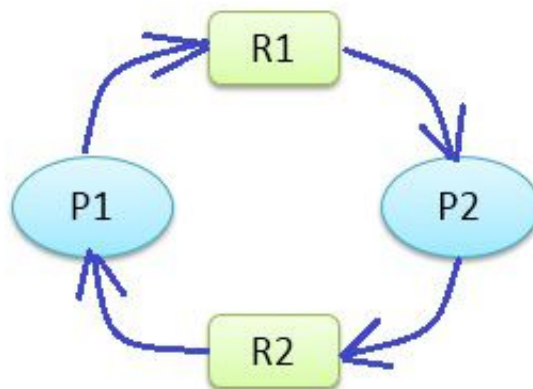
Dictionary, Lessons, Game Master English Speaking

 INSTALL



## Definition of Deadlock

Deadlock is a situation where the several processes in CPU compete for the finite number of resources available within the CPU. Here, each process holds a resource and wait to acquire a resource that is held by some other process. All the processes wait for resources in a circular fashion. In the image below, you can see that Process P1 has acquired resource R2 that is requested by process P2 and Process P1 is requesting for resource R1 which is again held by R2. So process P1 and P2 form a deadlock.



Deadlock is a common problem in multiprocessing operating systems, distributed systems, and also in parallel computing systems. There are four conditions which must occur simultaneously to raise the condition of deadlock, which are Mutual exclusion, Hold and waits, No preemption, and Circular wait.

- **Mutual exclusion:** Only one process at a time can use a resource if other process requests the same resource, it has to wait till the process using resource releases it.
- **Hold and Wait:** A process must be holding a resource and waiting to acquire another resource that is held by some other process.
- **No Preemption:** The process holding the resources can not be preempted. The process

holding the resource must release the resource voluntarily when it has completed its task.

- **Circular wait:** The process must wait for resources in a circular fashion. Suppose we have three processes {P0, P1, P2}. The P0 must wait for the resource held by P1; P1 must wait to acquire the resource held by process P2, and P2 must wait to acquire the process held by P0.

Although there are some applications that can detect the programs that may get deadlocked. But the operating system is never responsible for preventing the deadlocks. It is the responsibility of programmers to design deadlock free programs. It can be done by avoiding the above conditions which are necessary for deadlock occurrence

## Definition of Starvation

Starvation can be defined as when a process request for a resource and that resource has been continuously used by the other processes then the requesting process faces starvation. In starvation, a process ready to execute waits for CPU to allocate the resource. But the process has to wait indefinitely as the other processes continuously block the requested resources.

The problem of starvation generally occurs in **priority scheduling algorithm**. In priority scheduling algorithm, the process with higher priority is always allocated the resource, preventing the lower priority process from getting the requested resource.

**Aging** can resolve the problem of starvation. Aging gradually increases the priority of the process that has been waiting long for the resources. Aging prevents a process with low priority to wait indefinitely for a resource.

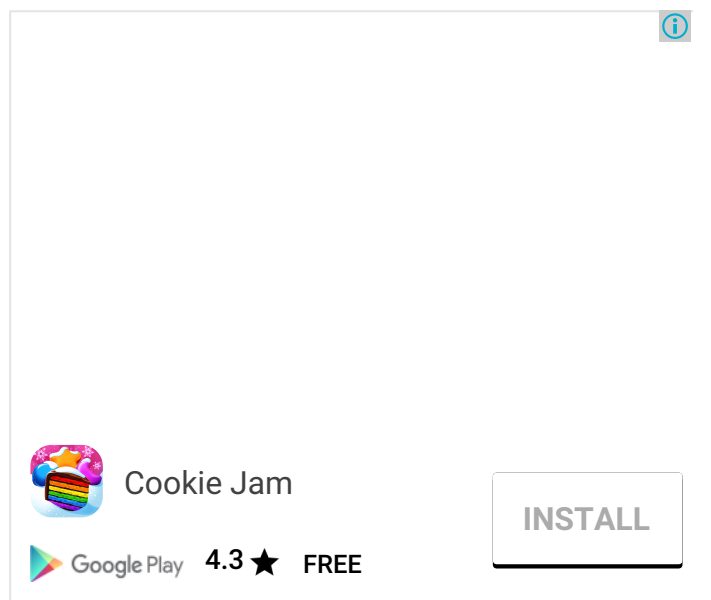
## Key Differences Between Deadlock and Starvation in OS

1. In a deadlock, none of the processes proceeds for execution, each process get blocked waiting for the resources acquired by the another process. On the other hand, starvation is a condition where the processes that possess higher priority is allowed to acquire the resources continuously by preventing the low priority processes to acquire resources resulting in indefinite blocking of low priority processes.
2. Deadlock arises when four conditions **Mutual exclusion, Hold and wait, No preemption, and Circular wait** occurs simultaneously. However, starvation occurs when process **priorities have been enforced** while allocating resources, or there is uncontrolled resource management in the system.

3. Deadlock is often called by the name **circular wait** whereas, the starvation is called **Lived lock**.
4. In Deadlock the resources are blocked by the process whereas, in starvation, the processes are continuously being used by the processes with high priorities.
5. Deadlock can be prevented by the avoiding the conditions like mutual exclusion, Hold and wait, and circular wait and by allowing the preemption of the processes that are holding resources for a long time. On the other hand, Starvation can be prevented by **aging**.

## Conclusion:

Both Deadlock and Starvation delays the process execution by blocking it. On one hand where deadlock can cause processes to starve, and on the other hands starvation can get the processes out of the deadlock.



## Related Differences:

1. [Difference Between Semaphore and Mutex](#)
2. [Difference Between Preemptive and Non-Preemptive Scheduling in OS](#)
3. [Difference Between Semaphore and Monitor in OS](#)
4. [Difference Between sleep\(\) and wait\(\) Method in Java](#)
5. [Difference Between dispose\(\) and finalize\(\) in C#](#)

# Leave a Reply

Your email address will not be published. Required fields are marked \*

Comment

Name \*

Email \*

Website

I'm not a robot

reCAPTCHA  
Privacy - Terms

POST COMMENT

Search the site ...

## TOP 10 DIFFERENCES

Difference between Synchronous and Asynchronous Transmission

Difference Between Preemptive and Non-Preemptive Scheduling in OS

---

Difference Between Logical and Physical Address in Operating System

---

Difference Between while and do-while Loop

---

Difference between Simplex, Half duplex and Full Duplex Transmission Modes

---

Difference Between One-Dimensional (1D) and Two-Dimensional (2D) Array

---

Difference Between LAN, MAN and WAN

---

Difference Between Go-Back-N and Selective Repeat Protocol

---

Difference Between Circuit Switching and Packet Switching

---

Difference Between Pure ALOHA and Slotted ALOHA

---

## RECENT ADDITION

Difference Between RAM and ROM Memory

---

Difference Between Supercomputer and Mainframe Computer

---

Difference Between Primary and Secondary Memory

---

Difference Between Loosely Coupled and Tightly Coupled Multiprocessor System

---

Difference Between Magnetic Tape and Magnetic Disk

---

## CATEGORIES


DBMS

---

Networking

Operating System

Programming



Tech Differences

96 likes

Like Page

Share

1 friend likes this



Tech Differences

19 January at 17:20

Have you ever tried to find the differences between Magnetic Tape and Magnetic Disk  
<http://techdifferences.com/difference-between-magnetic-tape...>

