**13) Explain a semaphore?**

**Ans:-** Semaphores are integer variables that are used to solve the critical section problem by using two atomic operations, wait and signal that are used for process synchronization.

The definitions of wait and signal are as follows −

* **Wait**

The wait operation decrements the value of its argument S, if it is positive. If S is negative or zero, then no operation is performed.

* **Signal**

The signal operation increments the value of its argument S.

## Types of Semaphores

There are two main types of semaphores i.e. counting semaphores and binary semaphores. Details about these are given as follows −

* **Counting Semaphores**

These are integer value semaphores and have an unrestricted value domain. These semaphores are used to coordinate the resource access, where the semaphore count is the number of available resources. If the resources are added, semaphore count automatically incremented and if the resources are removed, the count is decremented.

* **Binary Semaphores**

The binary semaphores are like counting semaphores but their value is restricted to 0 and 1. The wait operation only works when the semaphore is 1 and the signal operation succeeds when semaphore is 0. It is sometimes easier to implement binary semaphores than counting semaphores.

**Advantage of Semaphores**

Some of the advantages of semaphores are as follows −

* Semaphores allow only one process into the critical section. They follow the mutual exclusion principle strictly and are much more efficient than some other methods of synchronization.
* There is no resource wastage because of busy waiting in semaphores as processor time is not wasted unnecessarily to check if a condition is fulfilled to allow a process to access the critical section.
* Semaphores are implemented in the machine independent code of the microkernel. So they are machine independent.

## **Disadvantages of Semaphores**

Some of the disadvantages of semaphores are as follows −

* Semaphores are complicated so the wait and signal operations must be implemented in the correct order to prevent deadlocks.
* Semaphores are impractical for last scale use as their use leads to loss of modularity. This happens because the wait and signal operations prevent the creation of a structured layout for the system.
* Semaphores may lead to a priority inversion where low priority processes may access the critical section first and high priority processes later.

**14) Explain a mutex?**

A **mutex** is a binary variable whose purpose is to provide locking mechanism. It is used to provide mutual exclusion to a section of code, means only one process can work on a particular code section at a time.

There is misconception that binary **semaphore** is same as mutex variable but both are different in the sense that binary semaphore apart from providing locking mechanism also provides two atomic operation signal and wait, means after releasing resource semaphore will provide signaling mechanism for the processes who are waiting for the resource.

**15) Difference among kernel space and user space.**

|  |  |
| --- | --- |
| Kernel Space | User Space |
| Running program by keeping CPU in low privilege mode | Running program by keeping CPU in high privilege mode. |
| Library calls are present | System calls are present |
| Library calls are portable | System calls are fast |
| Indirect way of accessing kernel | Direct way of accessing kernel |
| Library calls are slow | System calls are fast |
| Program starts with main() | There are many entry points |
| Segmentation fault and infinite loop will not give problem | Segmentation fault and infinite loop will corrupt program |
| Returns is not compulsory | Return is compulsory |

**16) Write in brief the ping command.**

**Ans:-** Ping is a command-line utility, available on virtually any operating system with network connectivity, that acts as a test to see if a networked device is reachable. The ping command sends a request over the **network** to a specific device.