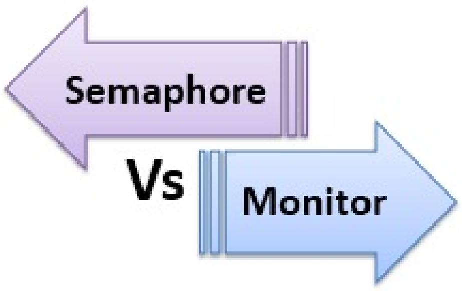
Difference Between Semaphore and Monitor in OS

January 6, 2017 — Leave a Comment



Semaphore and Monitor both allow processes to access the shared resources in mutual exclusion. Both are the process synchronization tool. Instead, they are very different from each other. Where **Synchronization** is an integer variable which can be operated only by wait() and signal() operation apart from the initialization. On the other hand, the **Monitor** type is an abstract data type whose construct allow one process to get activate at one time. In this article, we will discuss the differences between semaphore and monitor with the help of comparison chart shown below.

Content: Semaphore Vs Monitor

Comparison Chart

BASIS FOR COMPARISON	SEMAPHORE	MONITOR
Basic	Semaphores is an integer variable S.	Monitor is an abstract data type.
Action	The value of Semaphore S indicates the number of shared resources availabe in the system	The Monitor type contains shared variables and the set of procedures that operate on the shared variable.
Access	When any process access the shared resources it perform wait() operation on S and when it releases the shared resources it performs signal() operation on S.	When any process wants to access the shared variables in the monitor, it needs to access it through the procedures.
Condition variable	Semaphore does not have condition variables.	Monitor has condition variables.

Definition of Semaphore

Being a process synchronization tool, **Semaphore** is an **integer variable S.** This integer variable S is initialized to the **number of resources** present in the system. The value of semaphore S can be modified only by two functions **wait**() and **signal**() apart from initialization.

The wait() and signal() operation modifies the value of the semaphore S indivisibly. Which means when a process is modifying the value of the semaphore, no other process can simultaneously modify the value of the semaphore. Further, the operating system distinguishes the semaphore in two categories Counting semaphores and Binary semaphore.

In **Counting Semaphore**, the value of semaphore S is initialized to the number of resources present in the system. Whenever a process wants to access the shared resources, it performs **wait**() operation on the semaphore which **decrements** the value of semaphore by one. When it releases the shared resource, it performs a **signal**() operation on the semaphore which **increments** the value of semaphore by one. When the semaphore count goes to **0**, it means **all resources are occupied** by the processes. If a process need to use a resource when semaphore count is 0, it executes wait() and get **blocked** until a process utilizing the shared resources releases it and the value of semaphore becomes greater than 0.

In **Binary semaphore**, the value of semaphore ranges between 0 and 1. It is similar to mutex lock, but mutex is a locking mechanism whereas, the semaphore is a signalling mechanism. In binary semaphore, if a process wants to access the resource it performs wait() operation on the semaphore and **decrements** the value of semaphore from 1 to 0. When process releases the resource, it performs a **signal**() operation on the semaphore and increments its value to 1. If the value of the semaphore is 0 and a process want to access the resource it performs wait() operation and block itself till the current process utilizing the resources releases the resource.

Definition of Monitor

To overcome the timing errors that occurs while using semaphore for process synchronization, the researchers have introduced a high-level synchronization construct i.e. the **monitor type**. A monitor type is **an abstract data type** that is used for process synchronization.

Being an abstract data type monitor type contains the **shared data variables** that are to be shared by all the processes and some programmer-defined **operations** that allow processes to execute in mutual exclusion within the monitor. A process can **not directly access** the shared data variable in the monitor; the process has to access it **through procedures** defined in the monitor which allow only one process to access the shared variables in a monitor at a time.

The syntax of monitor is as follow:

```
monitor monitor_name
{
//shared variable declarations
procedure P1 ( . . . ) {
}
procedure P2 ( . . . ) {
}
procedure Pn ( . . . ) {
}
initialization code ( . . . ) {
}
}
```

A monitor is a construct such as only one process is active at a time within the monitor. If other process tries to access the shared variable in monitor, it gets blocked and is lined up in the queue to get the access to shared data when previously accessing process releases it.

Conditional variables were introduced for additional synchronization mechanism. The conditional variable **allows a process to wait inside the monitor** and allows a waiting process to resume immediately when the other process releases the resources.

The **conditional variable** can invoke only two operation **wait**() and **signal**(). Where if a process **P invokes a wait()** operation it gets suspended in the monitor till other process **Q invoke signal()** operation i.e. a signal() operation invoked by a process resumes the suspended process.

Key Differences Between Semaphore and Monitor

1. The basic difference between semaphore and monitor is that the **semaphore** is an **integer variable S** which indicate the number of resources available in the system whereas, the **monitor** is the **abstract data type** which allows only one process to execute in critical section at a time.
2. The value of semaphore can be modified by **wait()** and **signal()** operation only. On the other hand, a monitor has the shared variables and the procedures only through which shared variables can be accessed by the processes.
3. In Semaphore when a process wants to access shared resources the process performs **wait**() operation and block the resources and when it release the resources it performs **signal**() operation. In monitors when a process needs to access shared resources, it has to access them through procedures in monitor.
4. Monitor type has **condition variables** which semaphore does not have.

Conclusion:

Monitors are easy to implement than semaphore, and there is little chance of mistake in monitor in comparison to semaphores.

God's Purpose For You

Discover What You Were Meant For. Find Your Purpose. Learn More Now!

journeyanswers.com

Related Differences:

1. Difference Between Semaphore and Mutex
2. Difference Between Interface and Abstract Class in Java & C#
3. Difference Between Analog and Digital Signal
4. Difference Between sleep() and wait() Method in Java
5. Difference between Simplex, Half duplex and Full Duplex Transmission Modes

Filed Under: Operating System

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

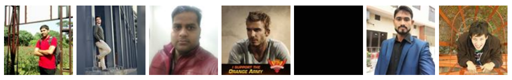
Difference Between Magnetic Tape and Magnetic Disk

CATEGORIES

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 differnces between Magnetic Tape and Magnetic Disk http://techdifferences.com/difference-between-magnetic-tape…