

Process Synchronization | Monitors

Monitor is one of the ways to achieve Process synchronization. Monitor is supported by programming languages to achieve mutual exclusion between processes. For example Java Synchronized methods. Java provides wait() and notify() constructs.

1. It is the collection of condition variables and procedures combined together in a special kind of module or a package.
2. The processes running outside the monitor can't access the internal variable of monitor but can call procedures of the monitor.
3. Only one process at a time can execute code inside monitors.

Syntax of Monitor

```
Monitor Demo //Name of Monitor
{
  variables;
  condition variables;

  procedure p1 {...}
  procedure p2 {...}

}
```

Syntax of Monitor

Condition Variables

Two different operations are performed on the condition variables of the monitor.

Wait.
signal.

let say we have 2 condition variables
condition x, y; //Declaring variable

Wait **operation**

x.wait() : Process performing wait operation on any condition variable are suspended. The suspended processes are placed in block queue of that condition variable.

Note: Each condition variable has its unique block queue.

Signal **operation**

x.signal(): When a process performs signal operation on condition variable, one of the blocked processes is given chance.

```
If (x block queue empty)
  // Ignore signal
else
  // Resume a process from block queue.
```

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above