**Assignment**

**Question-1: What is the difference between RSS and VSZ ?**

**Answer: Link-** [**https://softwareperformancenotes.github.io/rssvsz/**](https://softwareperformancenotes.github.io/rssvsz/)

**RSS:** RSS is the Resident Set Size and is used to show how much memory is allocated to that process and is in RAM. It does not include memory that is swapped out. It does include memory from shared libraries as long as the pages from those libraries are actually in memory. It does include all stack and heap memory.

**VSZ:** VSZ is the Virtual Memory Size. It includes all memory that the process can access, including memory that is swapped out, memory that is allocated, but not used, and memory that is from shared libraries.

**======================================================================**

**Question-2: What is a Multitasking Operating system?**

**Answer: Link-** [**https://digitalthinkerhelp.com/what-is-multitasking-operating-system-with-their-examples-types/**](https://digitalthinkerhelp.com/what-is-multitasking-operating-system-with-their-examples-types/)Multitasking operating system provides the interface for executing the multiple program tasks by a single user at the same time on the one computer system. For example, users can open Gmail and PowerPoint at the same time.

There are **different types of multitasking OS** and below explained each one in detail:

### **True Multitasking**

True multitasking is capable of executing and processing multiple tasks concurrently without taking delay instead of switching tasks from one processor to another processor. It can perform a couple of tasks in parallel with underlying H/W or S/W.

### **Preemptive Multitasking**

Preemptive multitasking is a special task that is assigned to a computer **operating system**, in which it makes a decision on how much time is spent by one task before assigning another task for using the operating system. Operating system has control for completing this entire process, so it is known as “Preemptive”.

### **Cooperative Multitasking**

Cooperative multitasking is known as “Non-Preemptive Multitasking”. Main goal of Cooperative multitasking is to run the current task, and to release the CPU to allow another task run. This task is performed by calling taskYIELD().Context-switch is executed when this function is called.

There are some **examples of multi tasking OS** like as –

* Windows XP
* Windows Vista
* Windows 7
* Windows 8
* Windows 10
* IBM’s OS/390
* Linux
* UNIX

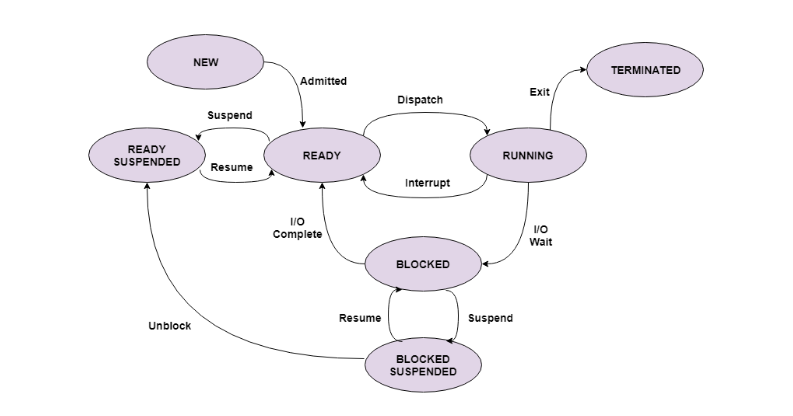
**======================================================================**

**Question-3: Difference states of process? Explain every state?**

**Answer: Link-** [**https://www.tutorialspoint.com/what-are-the-different-states-of-a-process**](https://www.tutorialspoint.com/what-are-the-different-states-of-a-process)

A process is an active program. It can also be said as a program that is under execution. It is more than the program code as it includes the program counter, process stack, registers, program code etc. Compared to this, the program code is only the text section.

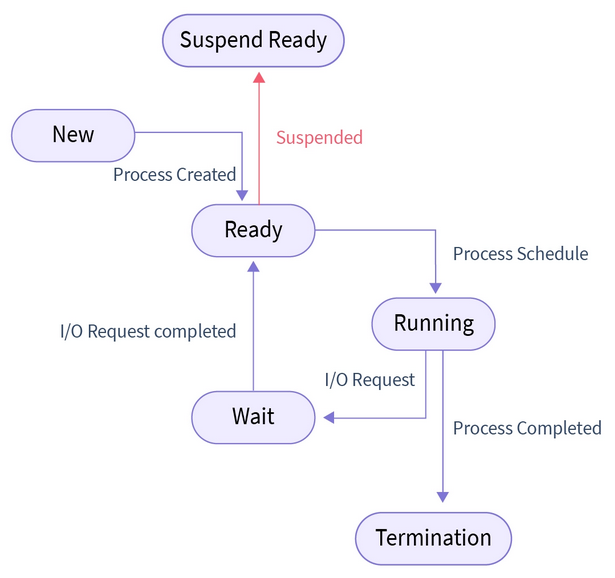
A process passes through different states as it executes. These states may be different in different operating systems. However, the common process states are explained below with the help of a diagram −



**New:** This is the state when the process has just been created. It is the initial state in the process life cycle.

## **Ready:** In the ready state, the process is waiting to be assigned the processor by the short term scheduler, so it can run. This state is immediately after the new state for the process.

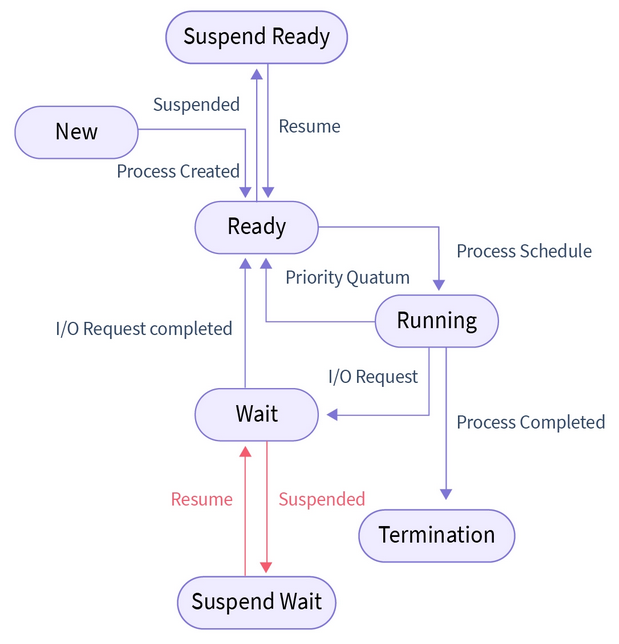
## **Ready Suspended:** The processes in ready suspended state are in secondary memory. They were initially in the ready state in main memory but lack of memory forced them to be suspended and get placed in the secondary memory.



## **Running:** The process is said to be in running state when the process instructions are being executed by the processor. This is done once the process is assigned to the processor using the short-term scheduler.

## **Blocked:** The process is in a blocked state if it is waiting for some event to occur. This event may be I/O as the I/O events are executed in the main memory and don't require the processor. After the event is complete, the process again goes to ready state.

## **Blocked Suspended:** This is similar to ready suspended. The processes in the blocked suspended state are in secondary memory. They were initially in the blocked state in main memory waiting for some event but lack of memory forced them to be suspended and get placed in the secondary memory. A process may go from blocked suspended to ready suspended if its work is done.



## **Terminated:** The process is terminated once it finishes its execution. In the terminated state, the process is removed from main memory and its process control block is also deleted.

## 

**======================================================================**

**Question-4: Explain PS command ?**

**Answer: Link-** [**https://docs.oracle.com/cd/E19455-01/805-7229/spprocess-47/index.html**](https://docs.oracle.com/cd/E19455-01/805-7229/spprocess-47/index.html)

The ps command enables us to check the status of active processes on a system, as well as display technical information about the processes. This data is useful for such administrative tasks as determining how to set process priorities.

Depending on which options we use, ps reports the following information:

* Current status of the process
* Process ID
* Parent process ID
* User ID
* Scheduling class
* Priority
* Address of the process
* Memory used
* CPU time used

The table below describes some of the fields reported by the ps command. The fields displayed depend on which option we choose. Table- Summary of Fields in ps Reports

| Field | Description |
| --- | --- |
| UID | The effective user ID of the process's owner. |
| PID | The process ID. |
| PPID | The parent process's ID. |
| C | The processor utilisation for scheduling. This field is not displayed when the -c option is used. |
| CLS | The scheduling class to which the process belongs: real-time, system, or time sharing. This field is included only with the -c option. |
| PRI | The kernel thread's scheduling priority. Higher numbers mean higher priority. |
| NI | The process's nice number, which contributes to its scheduling priority. Making a process "nicer" means lowering its priority. |
| ADDR | The address of the proc structure. |
| SZ | The virtual address size of the process. |
| WCHAN | The address of an event or lock for which the process is sleeping. |
| STIME | The starting time of the process (in hours, minutes, and seconds). |
| TTY | The terminal from which the process (or its parent) was started. A question mark indicates there is no controlling terminal. |
| TIME | The total amount of CPU time used by the process since it began. |
| CMD | The command that generated the process. |

**======================================================================**