

Process and Process States

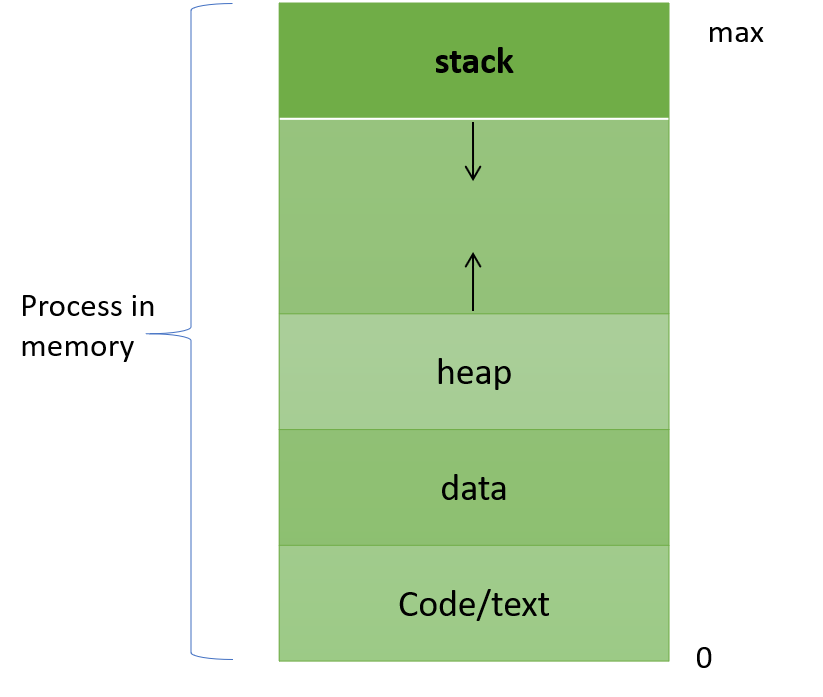
Have you ever executed a process? Have your process changed states? In this post, we going to cover the notion of a process and process states. We will understand: what is a process? How to convert your program into a process? What are the different states of a process? and Where all the information related to a process is stored?

**What is a Process?**

After you write a program in any language, two steps follow:  
1. Compiling  
2. Running/Executing  
The second step is what makes that program a process. You double-click any software in your computer system or you tap on any application your mobile or you write a command like $./a.out, all these convert the application(program) into a process. Every application is a program until you execute it by double click or a tap or a command, after which it becomes a process.  
*So, a process is a program in execution*

Now, once you initiate a process, the operating system loads it into the memory (RAM). Inside the RAM the structure of the process looks like as shown below:

1. Text – program code
2. Data – contains global variables
3. Stack – contains temporary data
   * function parameters
   * return addresses
   * Local variables
4. Heap – memory allocated dynamically during run time



The *text section* contains the program or the code, the *data section*contains the global variables. These two sections have fixed size because neither the code is going to change nor the variables used in the program. Heap is used for dynamic memory allocation. Now, we use dynamic memory allocation when we can not determine the memory required. Hence the heap section can grow in size if required.

Lastly, the *stack section* is used for functions. Again, the size of this section is also variable. Because it is difficult to determine the number of function calls required. Consider a program for factorial calculation which uses recursive functions. If the number is 5 then the function call is 5 times. If the number for factorial is 20, then the function call is 20 times. So the system does not know what exactly will be the stack size. Hence, the stack size is variable.

**Points to Remember**

Program: is a passive entity  
Process: is a active entity

**Question?**

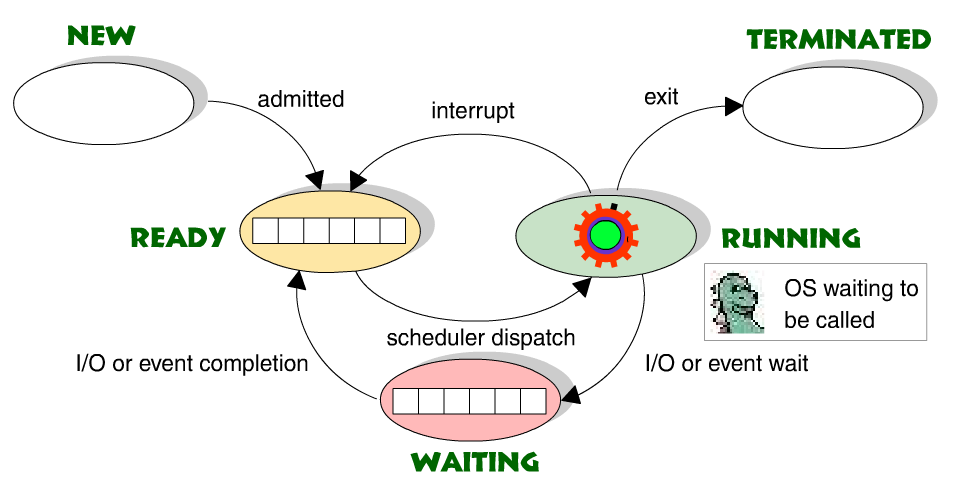
Q1.Can two processes be associated with the same program?

**Answer:**

**Process States**

A process can change its state during its lifetime. The various states of a process are:

1. New – when a process is created
2. Ready – when the process is in the RAM and is waiting for CPU allocation.
3. Running – the process gets the CPU and is executing in this state
4. Waiting – the process is waiting for some I/O device
5. Terminated – the process finishes its execution either normally or forcefully

[](http://williamstallings.com/OS-Animation/Queensland/PROCESS.SWF)

**Process Control Box(PCB)**

Sometimes a running process changes its state to waiting or ready and then back to running. In this case all the related information of a process needs to saved so that it can be later loaded when the process starts to execute again. This information is saved in *Process Control Box(PCB)*. Another name pf PCB is *Task Control Box*. The most common information stored is:

* Process state
* Program counter
* CPU registers
* CPU scheduling information
* Memory-management information
* Accounting information
* I/O status information

### State Diagram

### OS Process State Diagram

The process, from its creation to completion, passes through various states. The minimum number of states is five.

The names of the states are not standardized although the process may be in one of the following states during execution.

### 1. New

A program which is going to be picked up by the OS into the main memory is called a new process.

2. Ready

Whenever a process is created, it directly enters in the ready state, in which, it waits for the CPU to be assigned. The OS picks the new processes from the secondary memory and put all of them in the main memory.

The processes which are ready for the execution and reside in the main memory are called ready state processes. There can be many processes present in the ready state.

3. Running

One of the processes from the ready state will be chosen by the OS depending upon the scheduling algorithm. Hence, if we have only one CPU in our system, the number of running processes for a particular time will always be one. If we have n processors in the system then we can have n processes running simultaneously.

4. Block or wait

From the Running state, a process can make the transition to the block or wait state depending upon the scheduling algorithm or the intrinsic behavior of the process.

When a process waits for a certain resource to be assigned or for the input from the user then the OS move this process to the block or wait state and assigns the CPU to the other processes.

5. Completion or termination

When a process finishes its execution, it comes in the termination state. All the context of the process (Process Control Block) will also be deleted the process will be terminated by the Operating system.

### Suspend ready

A process in the ready state, which is moved to secondary memory from the main memory due to lack of the resources (mainly primary memory) is called in the suspend ready state.

### If the main memory is full and a higher priority process comes for the execution then the OS have to make the room for the process in the main memory by throwing the lower priority process out into the secondary memory. The suspend ready processes remain in the secondary memory until the main memory gets available.

### Suspend wait

Instead of removing the process from the ready queue, it's better to remove the blocked process which is waiting for some resources in the main memory. Since it is already waiting for some resource to get available hence it is better if it waits in the secondary memory and make room for the higher priority process. These processes complete their execution once the main memory gets available and their wait is finished.