Introduction of Process Management

geeksforgeeks.org/introduction-of-process-management

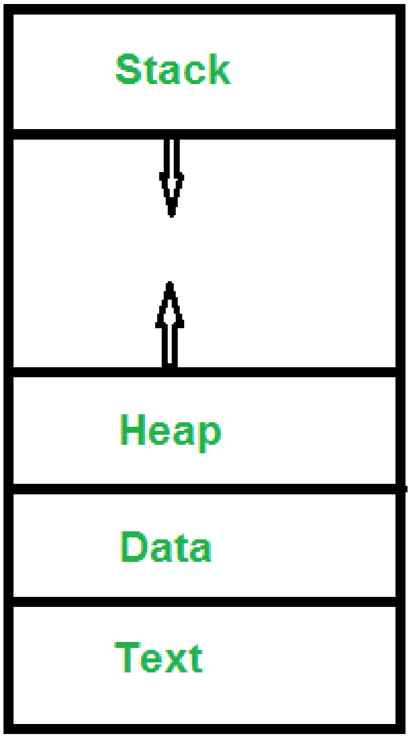
June 14, 2015

Program vs Process
A process is a program in execution. For example, when we write a program in C or C++ and compile it, the compiler creates binary code. The original code and binary code are both programs. When we actually run the binary code, it becomes a process.

A process is an 'active' entity, as opposed to a program, which is considered to be a 'passive' entity. A single program can create many processes when run multiple times; for example, when we open a .exe or binary file multiple times, multiple instances begin (multiple processes are created).

What does a process look like in memory?

Text Section*:*A Process, sometimes known as the Text Section, also includes the current activity represented by the value of the ***Program Counter***.
Stack*:* The Stack contains the temporary data, such as function parameters, returns addresses, and local variables.
Data Section*:* Contains the global variable.
Heap Section*:* Dynamically allocated memory to process during its run time.
Refer this for more details on sections.



Attributes or Characteristics of a Process
A process has following attributes.

1. Process Id: A unique identifier assigned by the operating system
2. Process State: Can be ready, running, etc.
3. CPU registers: Like the Program Counter (CPU registers must be saved and restored when a process is swapped in and out of CPU)
5. Accounts information:
6. I/O status information: For example, devices allocated to the process, open files, etc
8. CPU scheduling information: For example, Priority (Different processes may have different priorities, for example a short process may be assigned a low priority in the shortest job first scheduling)

All of the above attributes of a process are also known as the ***context of the process***. Every process has its own process control block(PCB), i.e each process will have a unique PCB. All of the above attributes are part of the PCB.

States of Process:
A process is in one of the following states:

1. New: Newly Created Process (or) being-created process.

2. Ready: After creation process moves to Ready state, i.e. the process is ready for execution.

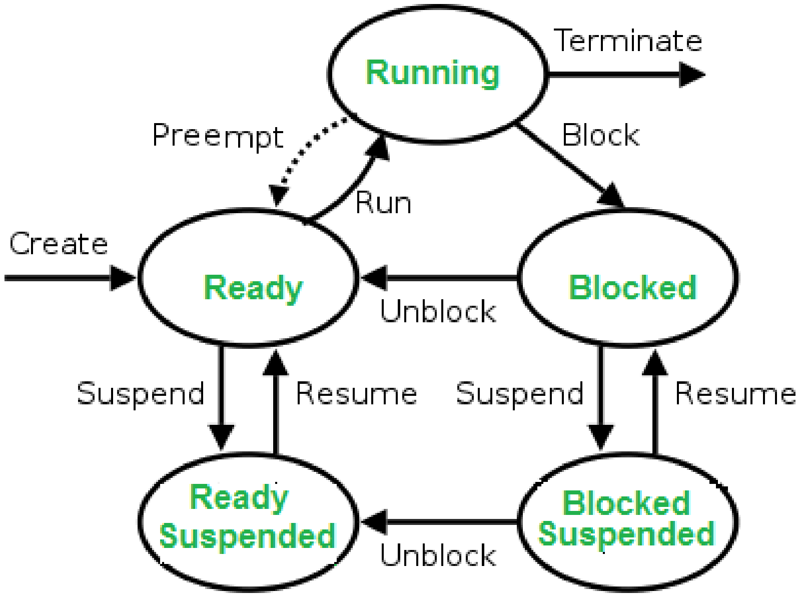
3. Run: Currently running process in CPU (only one process at a time can be under execution in a single processor).

4. Wait (or Block): When a process requests I/O access.

5. Complete (or Terminated): The process completed its execution.

6. Suspended Ready: When the ready queue becomes full, some processes are moved to suspended ready state

7. Suspended Block: When waiting queue becomes full.



Context Switching

The process of saving the context of one process and loading the context of another process is known as Context Switching. In simple terms, it is like loading and unloading the process from running state to ready state.

When does context switching happen?

1. When a high-priority process comes to ready state (i.e. with higher priority than the running process)
2. An Interrupt occurs
3. User and kernel mode switch (It is not necessary though)
4. Preemptive CPU scheduling used.

Context Switch vs Mode Switch

A mode switch occurs when CPU privilege level is changed, for example when a system call is made or a fault occurs. The kernel works in more a privileged mode than a standard user task. If a user process wants to access things which are only accessible to the kernel, a mode switch must occur. The currently executing process need not be changed during a mode switch.
A mode switch typically occurs for a process context switch to occur. Only the kernel can cause a context switch.

CPU-Bound vs I/O-Bound Processes:

A CPU-bound process requires more CPU time or spends more time in the running state.
An I/O-bound process requires more I/O time and less CPU time. An I/O-bound process spends more time in the waiting state.

Exercise:
1. Which of the following need not necessarily be saved on a context switch between processes? (GATE-CS-2000)
(A) General purpose registers
(B) Translation lookaside buffer
(C) Program counter
(D) All of the above

Answer (B)

Explanation:
In a process context switch, the state of the first process must be saved somehow, so that when the scheduler gets back to the execution of the first process, it can restore this state and continue. The state of the process includes all the registers that the process may be using, especially the program counter, plus any other operating system-specific data that may be necessary. A translation look-aside buffer (TLB) is a CPU cache that memory management hardware uses to improve virtual address translation speed. A TLB has a fixed number of slots that contain page table entries, which map virtual addresses to physical addresses. On a context switch, some TLB entries can become invalid, since the virtual-to-physical mapping is different. The simplest strategy to deal with this is to completely flush the TLB.

2. The time taken to switch between user and kernel modes of execution is t1 while the time taken to switch between two processes is t2. Which of the following is TRUE? (GATE-CS-2011)
(A) t1 > t2
(B) t1 = t2
(C) t1 < t2
(D) nothing can be said about the relation between t1 and t2.

Answer: (C)
Explanation: Process switching involves mode switch. Context switching can occur only in kernel mode.

Quiz on Process Management

References:

http://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/3_Processes.html
http://cs.nyu.edu/courses/spring11/G22.2250-001/lectures/lecture-04.html

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

Improved By : Nilesh Singh 2, chrismaher37, SarthakSinghal1