CSE3241: Operating System and System Programming

Class-4

Sangeeta Biswas, Ph.D.

Assistant Professor
Dept. of Computer Science and Engineering (CSE)
Faculty of Engineering
University of Rajshahi (RU)
Rajshahi-6205, Bangladesh
E-mail: sangeeta.cse@ru.ac.bd

Multitasking OS

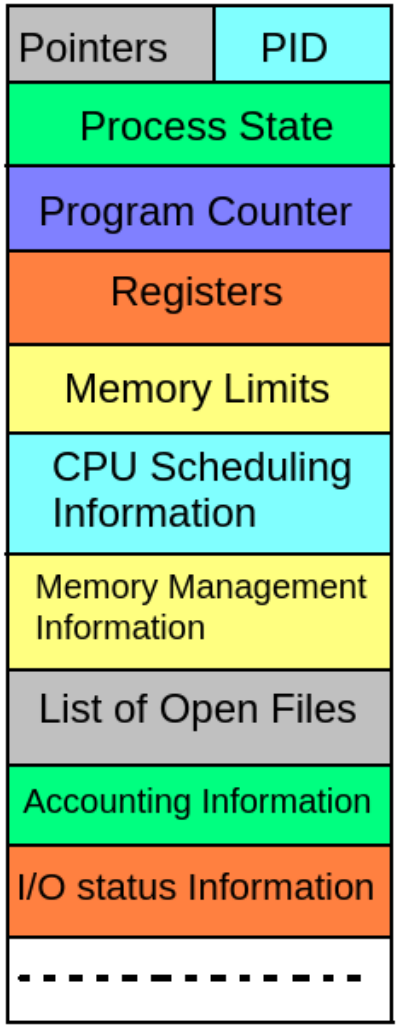
Multi-tasking OS allows a user to perform multiple computer tasks using a single set of resources in such a way that user get pseudo-parallel feeling, i.e., the user feels all of her/his tasks are running in parallel.

- Almost all modern OS, such as Microsoft Windows 2000, IBM's OS/390, and Linux, have multi-tasking capability.
- The main aim of multitasking is to ensure maximum utilization of advanced CPU, which is much more faster than the old time CPU, by keeping it busy at the maximum time.
- In multitasking, OS switches execution power from one process to another frequently so that each process has a progress instead of waiting until a specific process completely finished.
- OS needs to keep track of all processes to do multitasking smoothly, therefore, it uses **Process Control Block**.

Process Control Block

Process Control Block (PCB) is a block of information that represent a process in an OS. It is also known as a **task control block**.

- **Process state**: running, waiting, etc.
- **Program counter**: location of instruction to next execute.
- **CPU registers**: contents of all process-centric registers
- **CPU scheduling information**: priorities, scheduling queue pointers.
- **Memory limits**: memory allocated to the process.
- **Accounting information**: CPU used, clock time elapsed since start, time limits.
- **I/O status information**: I/O devices allocated to process.



PID

PID or Process number is a unique number of a process assigned by the OS so that it can easily keep track all current processes.

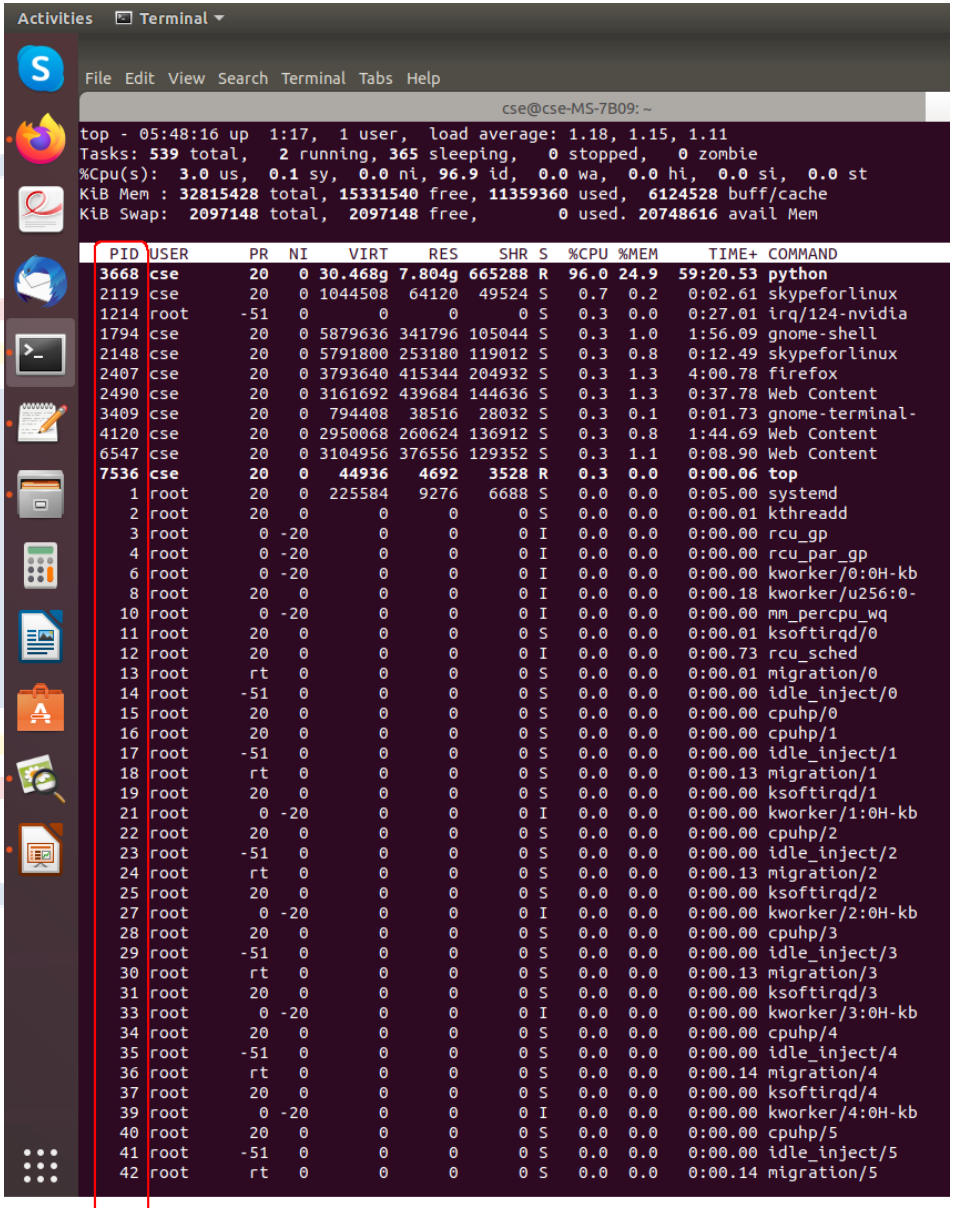
- It is only valid until the process is properly terminated in the current session.
  - If ,for some reasons, a process is killed/ finished execution, but do not get a chance to properly inform its parent process, then it holds the process ID.
- For a new start of a program, it gets a new PID.
  - So, if we run our code at different time/ different sessions, our program will become a new process and get a different PID.
- As long as a process has its PID, it has an entry in the **Process Table**.

Investigate PID and PCB in Ubuntu

Commands to see PID and PCBs of all processes, currently managed by the OS including its own processes:

- For PID:
  - $ top
- For PCB:
  - $ ls /proc
  - $ ls /proc/<PID>/

  e.g., $ ls /proc/3668



Process Table

The process table is an array of PCBs.

- It logically contains a PCB for each of the current process in the system.
- OS maintains pointers to each process's PCB in the process table so that it can access the PCB quickly.

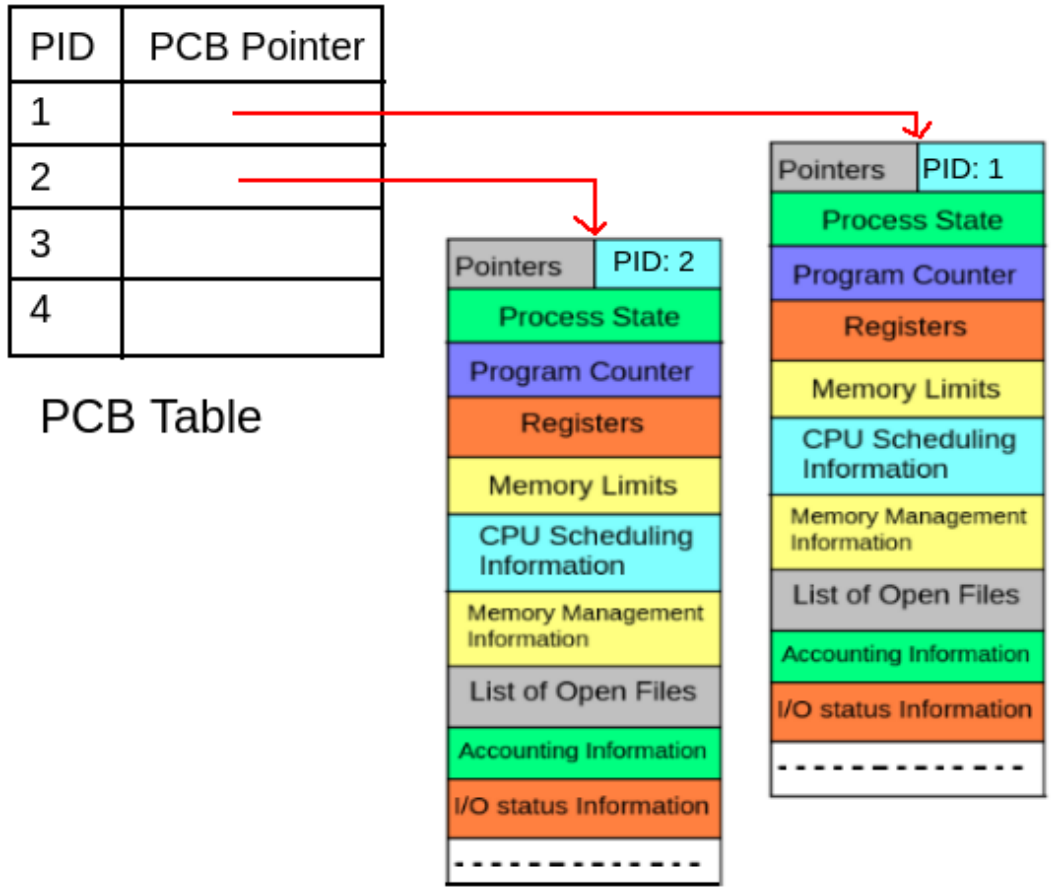


Figure: PCB table

Context Switch

A context switch occurs when the CPU switches from one process to another.

- OS saves the state of the old process and load the saved state for the new process via a context switch.