CSE3241: Operating System and System Programming

Lecture-7

(Process Concept)

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 / sangeeta.cse.ru@gmail.com

November 11, 2017

Outline

- Process Management
  - Process Concept
    - What is Process?
    - Process States
    - Process Control Block
    - Process Scheduling

What is Process?

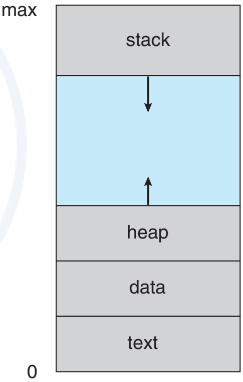
- A process is a program in execution. It is associated with an *address space*.

Address Space of a Process Contains:

- **Stack**: temporary data such as function parameters, return addresses, local variables.
- **Heap**: dynamically allocated memory locations.
- **Data Section**: global variables.
- **Text Section**: executable program.

- Multiple processes associated with the same program:
  - have equivalent text sections.
  - may have different data, heap and stack sections.

Process in Memory [1]



Process States

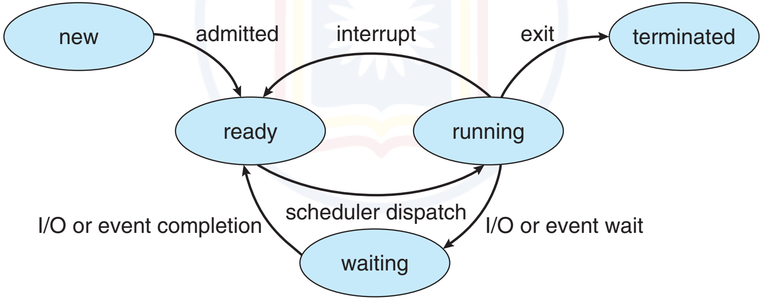
- As a process (say P) executes, it changes state.

- P may be at one of the following 5 states in a system:
  - **New**: P is being created.
  - **Running**: P's instructions are being executed.
  - **Waiting**: P is waiting for some event to occur.
  - **Ready**: P is waiting to be assigned to a processor
  - **Terminated**: P has finished execution.

- On a single processor machine:
  - only one process can be at running state at any instant.
  - many processes may be at ready and waiting states.

Diagram of Process States [1]

- A new process is initially put in ready state.
- Scheduller decides which process will go from ready to running state.
- An interrupted process goes from running to ready state.
- An I/O request sends a running process to the waiting state.
- After fulfilling demands, waited processes are sent back to ready state.



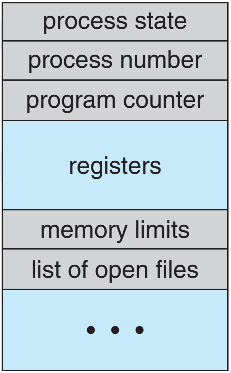
Process Control Block (PCB)

- Each process is represented in the OS by a process control block.

- Process control block (PCB):
  - is known as a task control block.
  - simply serves as the repository for any information that may vary from process to process.
  - is handled by the OS.

- PCB contains many pieces of information associated with a specific process, such as:
  - process state.
  - values of CPU registers.
  - information of CPU schedulig, memory management, I/O status, etc.

Figure: PCB [1]



process state

process number

program counter

registers

memory limits

list of open files

• • •

Some Fields of a PCB I

- Process ID: a unique identification number given by the OS.
- Parent ID: parent's unique ID.
- Process State: new / ready / running / waiting / halted.
- Values of CPU Registers: information stored in program counter, accumulator, index register, stack pointer, etc.
- CPU Scheduling Information: process priority, pointers to scheduling queues and so on.
- Memory-Management Information: values of base and limit registers, page table, segment table, memory limits, etc.

Some Fields of a PCB II

- Process Privileges: allowed/disallowed access to system resources.
- Interprocess Communication Information: various flags, signals and messages associated with the communication among independent processes.
- Process Structuring Information: process's children id's, or the id's of other processes related to the current one.
- Accounting Information: time CPU spent for the process execution, time limits.
- I/O Status Information: lists of allocated I/O devices, lists of opened files, etc.

Context Switch

- In a multiprogramming and single processor system, every user thinks only his/her single program is running in the system.

- In reality, CPU is switched among processes very frequently.

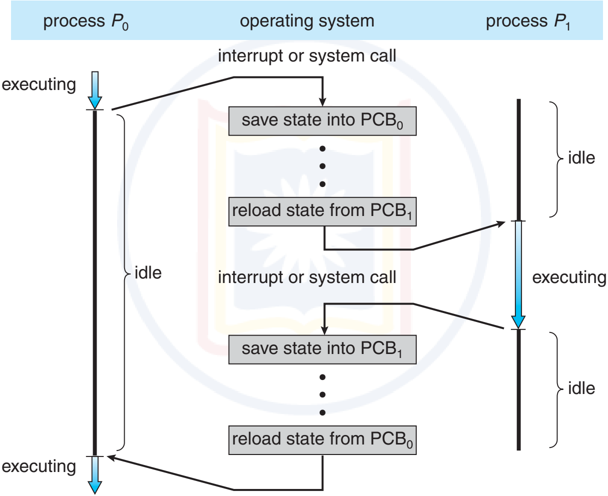
- The task, performed in order to switch the CPU from one process to another process, is known as context switch.

- When context switch occurs, the kernel instructs to:
  - save the state or context of the old process in its PCB, so that it's execution can be resumed from the same point at the later time.
  - load the saved context of the scheduled process from its PCB into CPU registers.

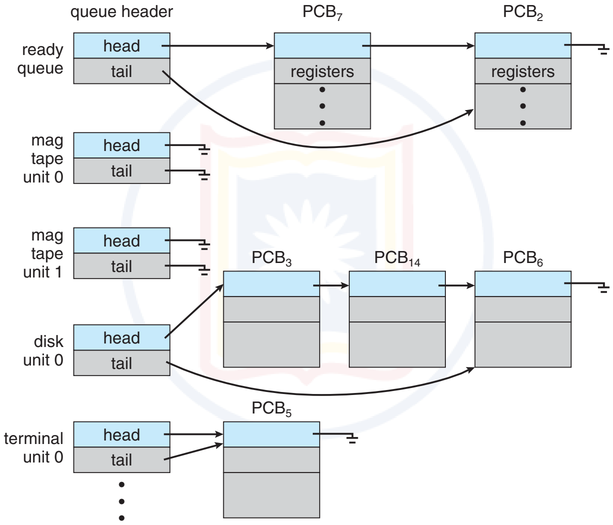
- Context-switch times are highly dependent on hardware support.

- The more complex the OS, the more work must be done during a context swicth.

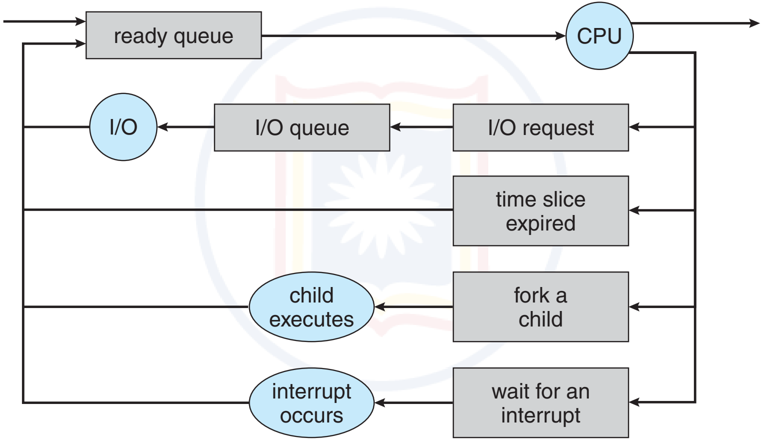
CPU Switches from Process to Process [1]



Various Queues [1]



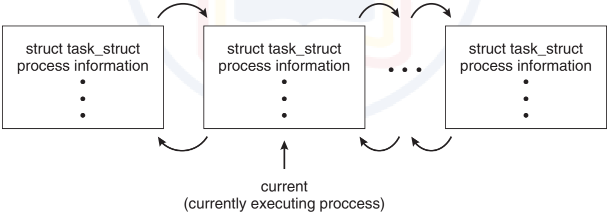
Queueing Diagram of Process Scheduling [1]



Process Reprsentation in Linux

- Linux uses C structure task_struct to hold PCB. Some fields are:
  - pid_t pid; [process identifier]
  - long state; [state of the process]
  - unsigned int time_slice; [scheduling information]
  - struct task_struct *parent; [this process's parent]
  - struct list_head children; [this process's children]

Figure: Doubly linked list of task_struct holding active processes in Linux [1]



References

A. Silberschatz, P. B. Galvin, and G. Gagne.
Operating System Concepts.
John Wiley & Sons, 9 edition, 2012.