Operating System

Week 4 - Long Descriptive Questions

1. Differentiate Long term scheduler from a Short term scheduler.

Ans.

- A long-term scheduler is an operating system scheduler that chooses processes from the job queue and loads them to execution in the main memory. On the other hand, a short-term scheduler is an operating system scheduler that chooses the process from the several processes that the processor runs.
- The long-term scheduler chooses the processes or jobs from the job pool. In contrast, the short-term scheduler chooses the processes from the ready queue.
- The long-term scheduler controls the multiprogramming degree. In contrast, the short-term scheduler has less control over multiprogramming.
- The long-term scheduler assigns the job to the ready queue for further action by the short-term scheduler, which is referred to as a job scheduler. In contrast, the short-term scheduler assigns the task to the CPU for its process; therefore, it is also called a CPU Scheduler.
- The short-term scheduler chooses processes from the ready queue more frequently than the long-term scheduler chooses processes from the job pool.
- The long-term scheduler is slower than the short-term scheduler.

Head-to-head comparison

Long Term Scheduler	Short Term Scheduler
It is an operating system scheduler that chooses processes from the job queue and loads them to execution in the main memory.	It is an operating system scheduler that chooses the process from the several processes that the processor runs.

It is also referred to as the Job Scheduler.	It is also referred to as a CPU scheduler.
It is slower.	It is faster.
It controls the multiprogramming degree.	It provides less control over the multiprogramming degree.
It selects the process less frequently.	It selects the process more frequently.
It is always present in the Batch OS and can or cannot be present at all in the Time-Sharing OS.	It is present in the Batch OS and is only minimally present in the Time-Sharing OS.
It chooses the processes from the job pool.	It chooses the processes from the ready queue.
It chooses a good process that is a mix-up of input/output bound and CPU bound.	It chooses a new process for a processor quite frequently.

2. Write a simple program using fork() and wait() .

Ans.

```
// Program to demonstrate bottom to up execution
// of processes using fork() and wait()
#include <iostream>
#include <sys/wait.h> // for wait()
#include <unistd.h> // for fork()
int main()
{
	// 1 parent : 2 child : 1 grand-child
	pid_t id1 = fork();
	pid_t id2 = fork();


	if (id1 > 0 && id2 > 0) {
		wait(NULL);
		wait(NULL);
		cout << "Parent Terminated" << endl;
	}


	else if (id1 == 0 && id2 > 0) {

		sleep(2);
		wait(NULL);
```

```
        cout << "1st child Terminated" << endl;
    }

    // second child
    else if (id1 > 0 && id2 == 0) {
        sleep(1);
        cout << "2nd Child Terminated" << endl;
    }

    // grand child
    else {
        cout << "Grand Child Terminated" << endl;
    }

    return 0;
}
```

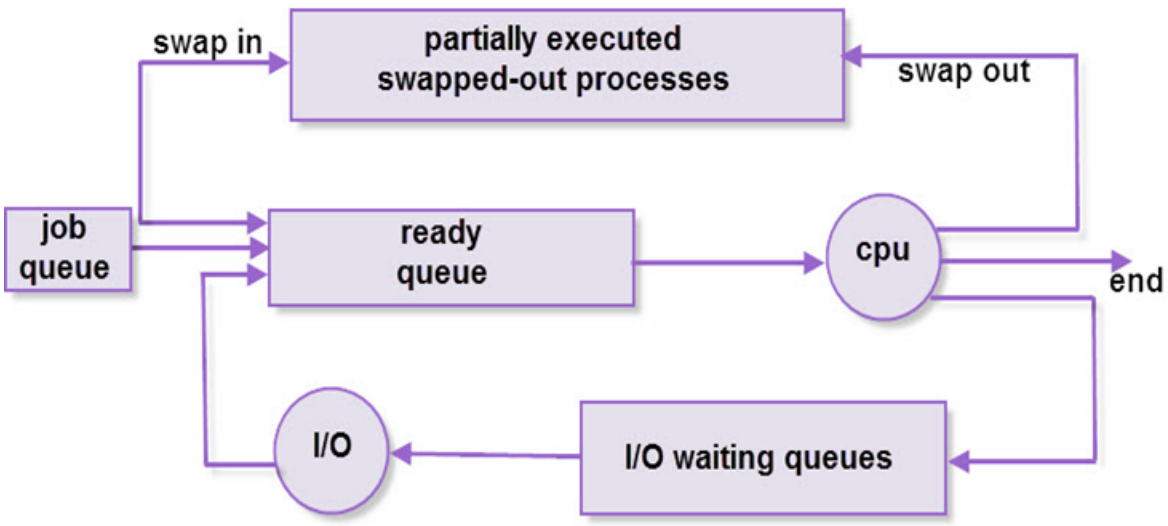
3. What is the need for a Medium-term scheduler ?

Ans.

Medium Term Scheduler

- The medium term scheduler controls the degree of multiprogramming.
- It admits processes into memory to compete for the CPU. Also called a swapper.
- Goal of medium term scheduler is to keep a good mix of the processes in memory so that the CPU is always kept busy.

Following Fig demonstrates medium-term scheduling :



4. Where do the status of a Process gets stored ? Explain it with a diagram.

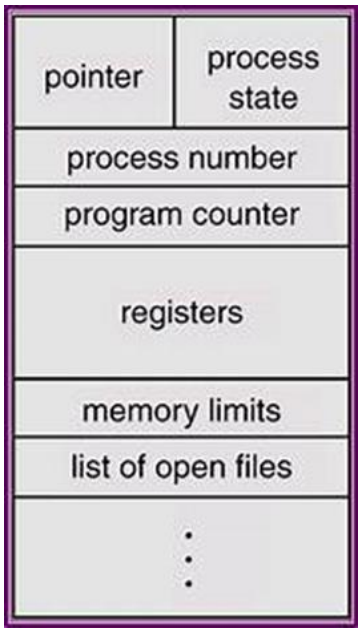
Ans. The states of the process are stored in Process Control Block(PCB). PCB is a special data structure that stores information about the process.

PCB is a data structure used by the operating system to maintain the information of a process.

Following are the process details:

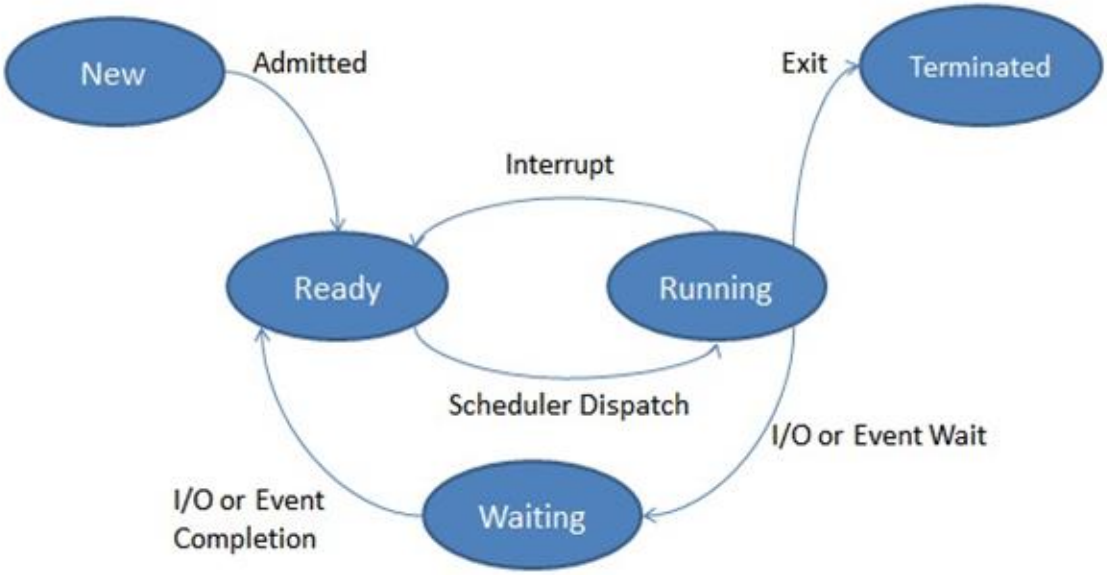
- Process state
- Program counter
- CPU registers
- CPU scheduling information
- Memory-management information
- Accounting information I/O status information

The following figure shows the PCB (Process Control Block)



5. What are the various states a process can take over during its life time with a diagram.

Ans. A process takes different states as it executes. These states may vary in different operating systems. However, Figure shows the common process states.



Various Process states:

· New

In this step , the process is about to be created but not yet created, it is the program which is currently available in secondary memory that will be chosen by OS to create the process.

· Ready

New -> Ready to run.

If a process gets loaded into the main memory it attains a ready state. It becomes ready to run and keeps waiting in a queue, to get the CPU time for its execution.

· Running

Ready -> Running

The process is chosen by CPU for execution and the instructions within the process are executed by any one of the available CPU cores.

· Waiting

Running -> waiting

Whenever the process requests access to I/O or needs input from the user or needs access to a critical region (the lock for which is already acquired) it enters the blocked or wait state. The process continues to wait in the main memory and does not require CPU. Once the I/O operation is completed the process goes to the ready state.

· Terminated

When the process gets finished, It is killed as well as PCB is deleted