Module - I OS, PROCESS AND THREADS

Part - A

1. What is an Operating system?

An operating system is a program that manages the computer hardware. It also provides a basis for application programs and act as an intermediary between a user of a computer and the computer hardware. It controls and coordinates the use of the hardware among the various application programs for the various users.

Operating System Definitions

- Resource allocator - manages and allocates resources.
- Control program - controls the execution of user programs and operation of I/O devices.
- Kernel - the one program running at all times (all else being application programs).

2. What are operating system goals?

Operating system goals:

- Execute user programs and make solving user problems easier.
- Make the computer system *convenient* to use.
- Use the computer hardware in an *efficient* manner.

3. Why is the Operating System viewed as a resource allocator & control program?

A computer system has many resources – hardware & software that may be required to solve a problem, like CPU time, memory space, file-storage space, I/O devices & so on. The OS acts as a manager for these resources so it is viewed as a resource allocator.

The OS is viewed as a control program because it manages the execution of user programs to prevent errors & improper use of the computer.

4. What is the Kernel?

A more common definition is that the OS is the one program running at all times on the computer, usually called the kernel, with all else being application programs.

5. What are Batch systems?

Batch systems are quite appropriate for executing large jobs that need little interaction. The user can submit jobs and return later for the results. It is not necessary to wait while the job is processed. Operators batched together jobs with similar needs and ran them through the computer as a group.

6. What is the advantage of Multiprogramming?

Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute. Several jobs are placed in the main memory and the processor is switched from job to job as needed to keep several jobs advancing while keeping the peripheral devices in use. Multiprogramming is the first instance where the Operating system must make decisions for the users. Therefore they are fairly sophisticated.

7. What is an Interactive computer system?

Interactive computer system provides direct communication between the user and the system. The user gives instructions to the operating system or to a program directly, using a keyboard or mouse, and waits for immediate results.

8. What do you mean by Time-sharing systems?

Time-sharing or multitasking is a logical extension of multiprogramming. It allows many users to share the computer simultaneously. The CPU executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while it is running.

9. What are multiprocessor systems & give their advantages?

Multiprocessor systems also known as parallel systems or tightly coupled systems are systems that have more than one processor in close communication, sharing the computer bus, the clock and sometimes memory & peripheral devices. Their main advantages are

- Increased throughput – by increasing number of processors, we hope to get more work done in less time.
- Economy of scale – save more money than multiple single processor systems, because they can share peripherals, mass storage and power supplies.
- Increased reliability – failure of one processor will not halt the system.

10. What are the different types of multiprocessing?

Symmetric multiprocessing (SMP): In SMP each processor runs an identical copy of the Os & these copies communicate with one another as needed. All processors are peers. Examples are Windows NT, Solaris, Digital UNIX, OS/2 & Linux.

Asymmetric multiprocessing: Each processor is assigned a specific task. A master processor controls the system; the other processors look to the master for instructions or predefined tasks. It defines a master-slave relationship. Example: SunOS Version 4.

11. What is graceful degradation?

In multiprocessor systems, failure of one processor will not halt the system, but only slow it down. If there are ten processors & if one fails the remaining nine processors pick up the work of the failed processor. This ability to continue providing service is proportional to the surviving hardware is called graceful degradation.

12. Define distributed systems.

Distributed Systems - distribute the computation among several physical processors.

Loosely coupled system - each processor has its own local memory; processors communicate with one another through various communication lines, such as high-speed buses or telephone lines.

Advantages of distributed systems:

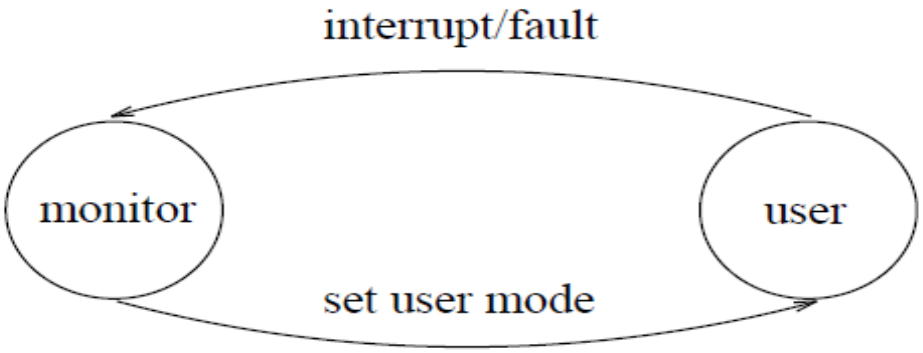
- Resource sharing
- Computation speed up - load sharing
- Reliability
- Communication

13. What is Dual- Mode Operation?

The dual mode operation provides us with the means for protecting the operating system from wrong users and wrong users from one another. Provide hardware support to differentiate between at least two modes of operations.

1. *User mode* - execution done on behalf of a user.
2. *Monitor mode* (also *supervisor mode* or *system mode*) - execution done on behalf of operating system.

Mode bit added to computer hardware to indicate the current mode: monitor (0) or user (1).
When an interrupt or fault occurs hardware switches to monitor mode



14. What are privileged instructions?

Some of the machine instructions that may cause harm to a system are designated as privileged instructions. The hardware allows the privileged instructions to be executed only in monitor mode.

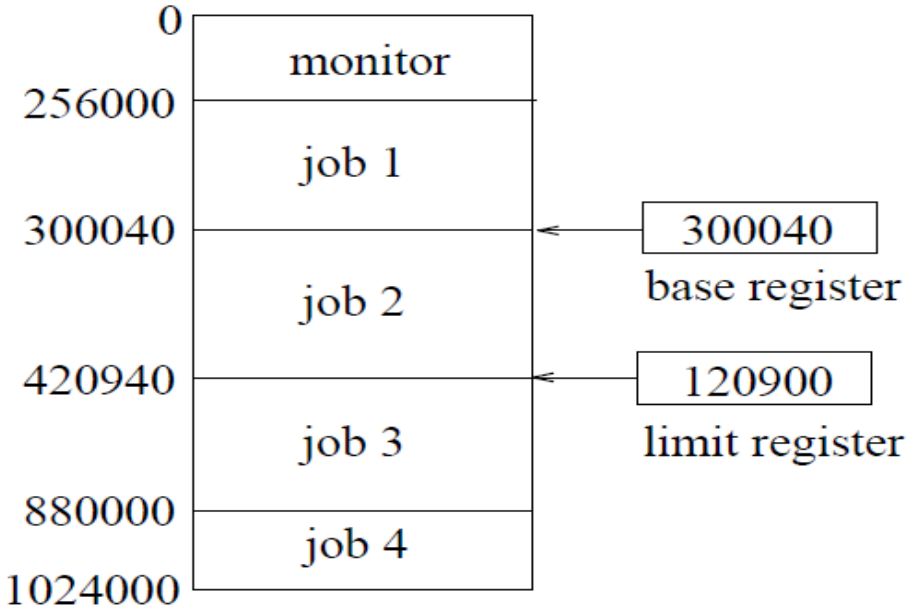
15. How can a user program disrupt the normal operations of a system?

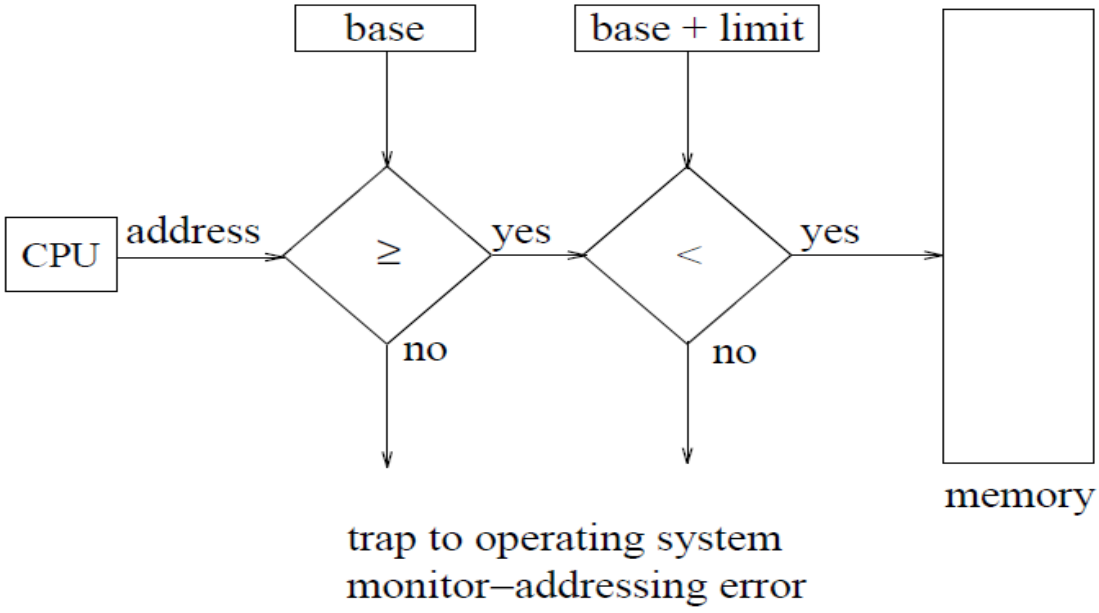
A user program may disrupt the normal operation of a system by

- Issuing illegal I/O operations
- By accessing memory locations within the OS itself
- Refusing to relinquish the CPU

16. How is the protection for memory provided?

The protection against illegal memory access is done by using two registers. The base register and the limit register. The **base register** holds the smallest legal physical address; the **limit register** contains the size of the range. The base and limit registers can be loaded only by the OS using special privileged instructions.





17. What are the various OS components?

The various system components are

- Process management
- Main-memory management
- File management
- I/O-system management
- Secondary-storage management
- Networking
- Protection system
- Command-interpreter system

18. What is a process?

A process is a program in execution. It is the unit of work in a modern operating system. A process is an active entity with a program counter specifying the next instructions to execute and a set of associated resources.

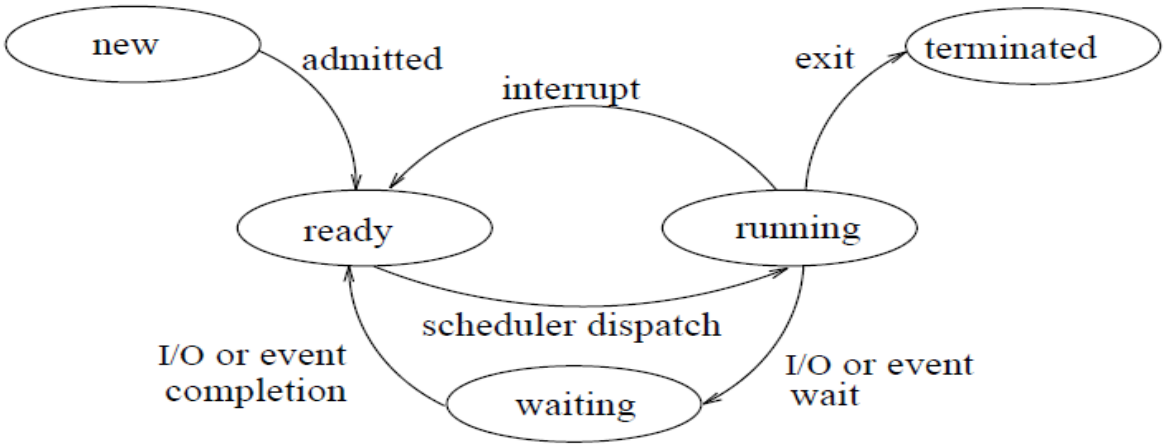
It also includes the process stack, containing temporary data and a data section containing global variables.

19. What is a process state and mention the various states of a process?

As a process executes, it changes state. The state of a process is defined in part by the current activity of that process. Each process may be in one of the following states:

As a process executes, it changes *state*.
- **New:** The process is being created.
- **Running:** Instructions are being executed.
- **Waiting:** The process is waiting for some event to occur.
- **Ready:** The process is waiting to be assigned to a processor.
- **Terminated:** The process has finished execution.

Diagram of process state:



20. What is process control block (PCB)?

Each process is represented in the operating system by a process control block also called a task control block. It contains many pieces of information associated with a specific process. It simply acts as a repository for any information that may vary from process to process.

It contains the following information:

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

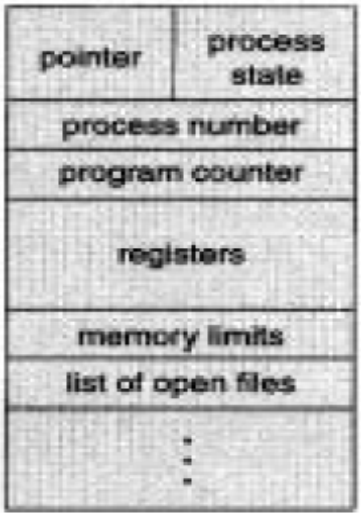
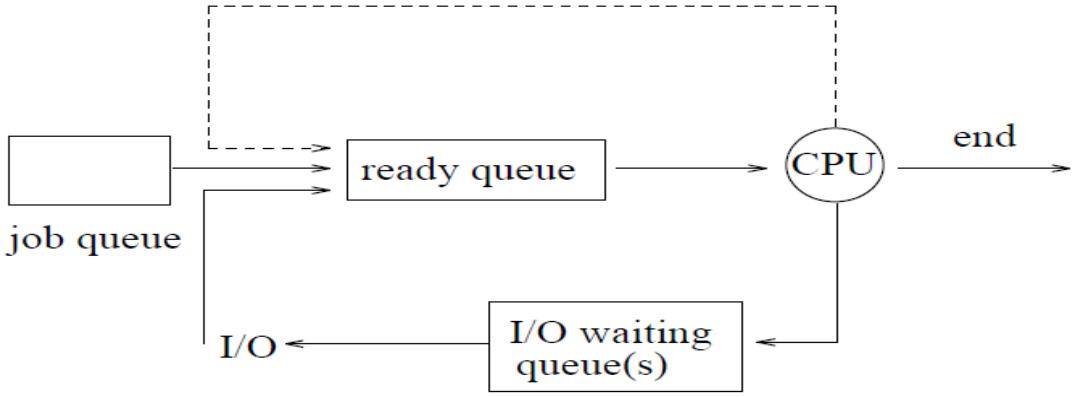


Figure 4.2 Process control block (PCB).

21. What are the use of job queues, ready queues & device queues?

As a process enters a system, they are put into a job queue. This queue consists of all jobs in the system. The processes that are residing in main memory and are ready & waiting to execute are kept on a list called ready queue. The list of processes waiting for a particular I/O device is kept in the device queue.



22. What is meant by context switch?

Switching the CPU to another process requires saving the state of the old process and loading the saved state for the new process. This task is known as context switch. The context of a process is represented in the PCB of a process.

Context-switch time is overhead; the system does no useful work while switching. Time dependent on hardware support.

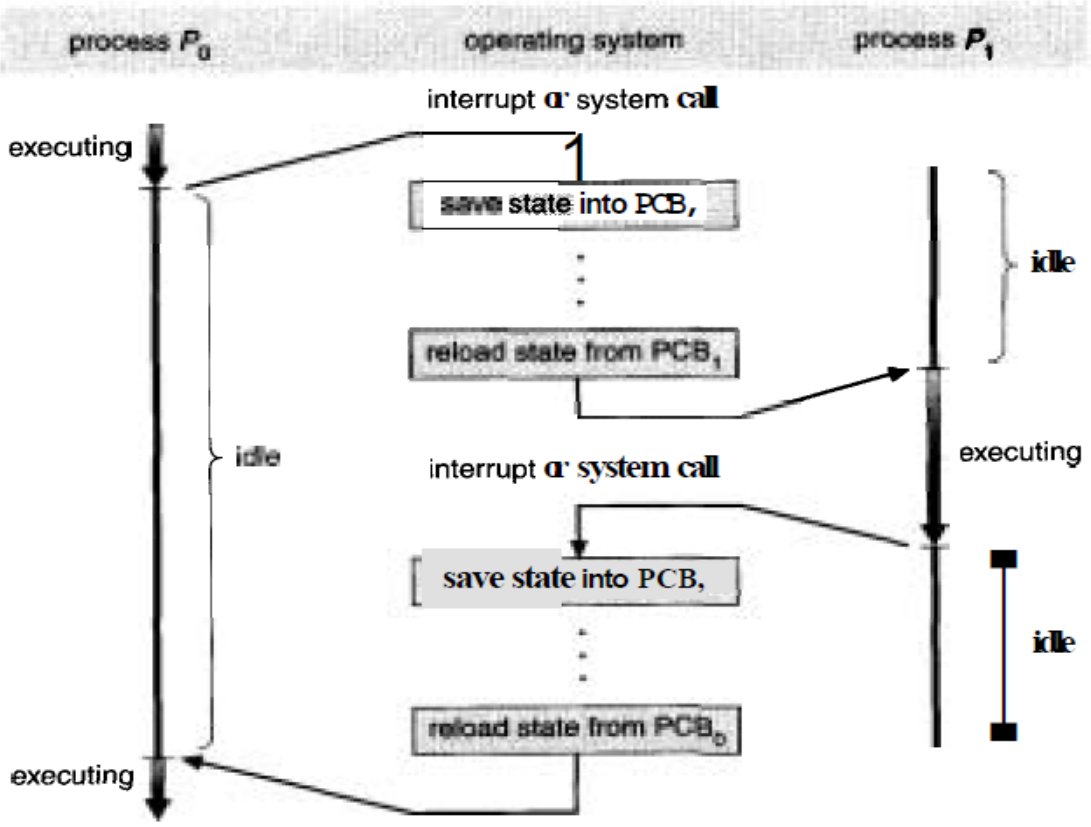


Figure 4.3 Diagram showing CPU switch from process to process.

23. What is spooling?

Spooling - Simultaneous peripheral operations on line. It is a high-speed device like a disk is interposed between a running program and a low –speed device involved with the program in input/output. It disassociates a running program from the slow operation of devices like printers.

24. What are System calls? How parameters are passed to the operating system?

System calls provide the interface between a process and the Operating system. System Calls are also called as Monitor call or Operating-system function call. When a system call is executed, it is treated as by the hardware as software interrupt. Control passes through the interrupt vector to a service routine in the operating system, and the mode bit is set to monitor mode.

Three general methods are used to pass parameters between a running program and the operating system:

- Pass parameters in *registers*.
- Store the parameters in a table in memory, and the table address is passed as a parameter in a register.
- *Push* (store) the parameters onto the *stack* by the program, and *pop* off the stack by the operating system.

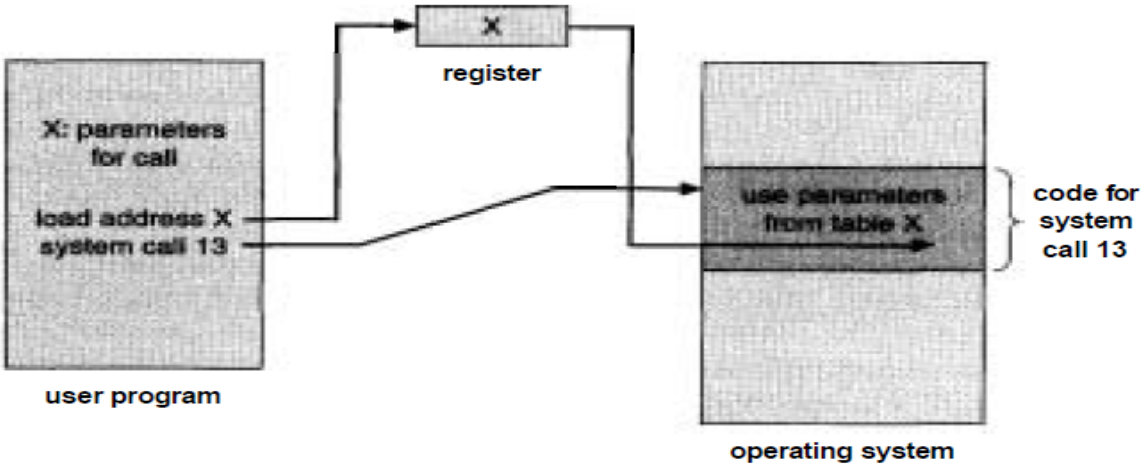


Figure 3.1 Passing of parameters as a table.

25. List the services provided by an Operating System?

- Program execution
- I/O Operation
- File-System manipulation
- Communications
- Error detection

26. What is the difference between Hard real time system and Soft real time system?

***Hard real-time* system**

- Secondary storage limited or absent; data stored in short-term memory, or read-only memory (ROM).
- Guarantees that critical tasks complete on time
- Conflicts with time-sharing systems; not supported by general-purpose operating systems.

***Soft real-time* system**

- Limited utility in industrial control or robotics.

- Critical real-time task gets priority over the other tasks, and retains that priority until it completes
- Useful in applications (multimedia, virtual reality) requiring advanced operating-system features.

27. Write the difference between multiprogramming and non multi-programming?

The operating system picks and begins to execute one of the jobs in the memory. Eventually, the job may have to wait for some task, such as a tape to be mounted, or an I/O operation to complete. In a non multi-programmed system, the CPU would sit idle. In a multiprogramming system, the operating system simply switches to and executes another job. When that job needs to wait, the CPU is switched to another job, and so on. Eventually, the first job finishes waiting and gets the CPU back. As long as there is always some job to execute, the CPU will never be idle.

28. What are the design goals of an operating system?

The requirements can be divided into two basic groups: User goals and System goals. Users desire that the system should be convenient and easy to use, easy to learn, reliable, safe and fast. The Operating system should be easy to design, implement, and maintain. Also it should be flexible, reliable, error free and efficient. These are some of the requirements, which are vague and have no general solution.

29. What are the five major categories of System Calls?

- Process Control
- File-management
- Device-management
- Information maintenance
- Communications

30. What is the use of fork and execve system calls?

Fork is a System calls by which a new process is created. Execve is also a System call, which is used after a fork by one of the two processes to replace the process memory space with a new program.

31. What is a thread?

A thread otherwise called a lightweight process (LWP) is a basic unit of CPU utilization, it comprises of a thread id, a program counter, a register set and a stack. It shares with other threads belonging to the same process its code section, data section, and operating system resources such as open files and signals.

32. What are the benefits of multithreaded programming?

The benefits of multithreaded programming can be broken down into four major categories:

- Responsiveness
- Resource sharing
- Economy
- Utilization of multiprocessor architectures

33. Compare user threads and kernel threads.

User threads	Kernel threads
User threads are supported above the kernel and are implemented by a thread library at the user level	Kernel threads are supported directly by the operating system
Thread creation & scheduling are done in the user space, without kernel intervention. Therefore they are fast to create and manage	Thread creation, scheduling and management are done by the operating system. Therefore they are slower to create & manage compared to user threads
Blocking system call will cause the entire process to block	If the thread performs a blocking system call, the kernel can schedule another thread in the application for execution

34. Define thread cancellation & target thread.

The thread cancellation is the task of terminating a thread before it has completed. A thread that is to be cancelled is often referred to as the target thread. For example, if multiple threads are concurrently searching through a database and one thread returns the result, the remaining threads might be cancelled.

35. What are the different ways in which a thread can be cancelled?

Cancellation of a target thread may occur in two different scenarios:

- *Asynchronous cancellation:* One thread immediately terminates the target thread is called asynchronous cancellation.
- *Deferred cancellation:* The target thread can periodically check if it should terminate, allowing the target thread an opportunity to terminate itself in an orderly fashion.

36. What are the two levels in threads?

Thread is implemented in two ways.

1.User level

2.Kernel level

Module - I OS, PROCESS AND THREADS

Part - B

1. **What are the system components of an operating system and explain them?**

Common System Components

- Process Management
- Main Memory Management
- File Management
- I/O System Management
- Secondary Management
- Networking
- Protection System
- Command-Interpreter System

2. **Define system calls. Write about the various system calls.**

Introduction

Types of System Calls

- Process control
- File management
- Device management
- Information maintenance
- Communications

3. What is a process? Explain the process control block and the various process states.

Introduction

- An operating system executes a variety of programs:
  - Batch system – jobs
  - Time-shared systems – user programs or tasks
- Textbook uses the terms job and process almost interchangeably.
- Process – a program in execution; process execution must progress sequentially
- A process includes:
  - program counter
  - stack
  - data section

Process State

- new: The process is being created.
- running: Instructions are being executed.
- waiting: The process is waiting for some event to occur.
- ready: The process is waiting to be assigned to a process.
- terminated: The process has finished execution.

4. Explain process creation and process termination

Process Creation

- ν Parent process create children processes, which, in turn create other processes, forming a tree of processes.
- ν Resource sharing
  - Φ Parent and children share all resources.
  - Φ Children share subset of parent's resources.
  - Φ Parent and child share no resources.
- ν Execution
  - Φ Parent and children execute concurrently.
  - Φ Parent waits until children terminate.
- ν Address space
  - Φ Child duplicate of parent.
  - Φ Child has a program loaded into it.
- ν UNIX examples
  - Φ fork system call creates new process
  - Φ exec system call used after a fork to replace the process' memory space with a new program.

Process Termination

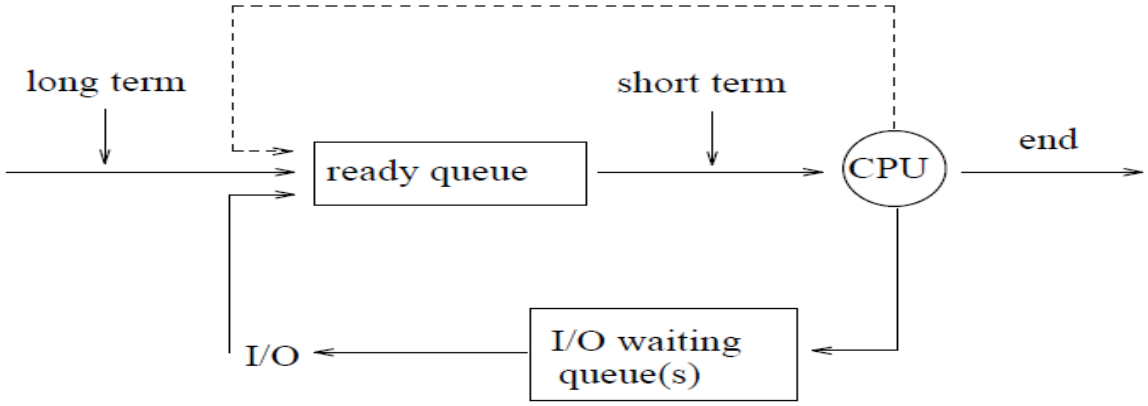
- ν Process executes last statement and asks the operating system to decide it (exit).
  - Φ Output data from child to parent (via wait).
  - Φ Process' resources are deallocated by operating system.
- ν Parent may terminate execution of children processes (abort).
  - Φ Child has exceeded allocated resources.
  - Φ Task assigned to child is no longer required.
  - Φ Parent is exiting.
    1. Operating system does not allow child to continue if its parent terminates.
    2. Cascading termination.

5. Write short notes on different schedulers.

***Long-term scheduler* (*job scheduler*)** – selects which processes should be brought into the ready queue.

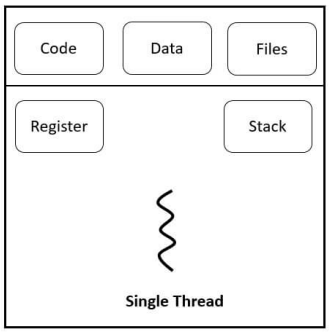
***Short-term scheduler* (*CPU scheduler*)** - selects which process should be executed next and allocates CPU.

Short-term scheduler is invoked very frequently (milliseconds) Þ(must be fast).
Long-term scheduler is invoked very infrequently (seconds, minutes) Þ(may be slow).
The long-term scheduler controls the ***degree of multiprogramming*.**



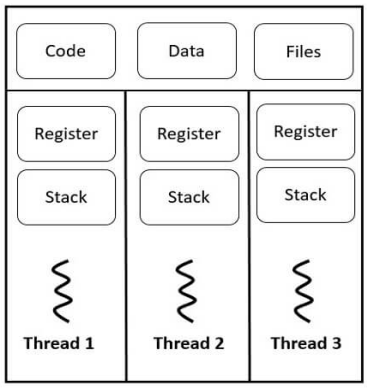
6. Illustrate the different types of multithreading models.

Single Thread:



Traditional Process with single thread

Multithreading:



Process with Multiple threads

Multithreading Models:

- **Many to one**
- **One to one**
- **Many to many**