MODULE:4 REAL TIME OPERATING SYSTEMS

Operating System Basics

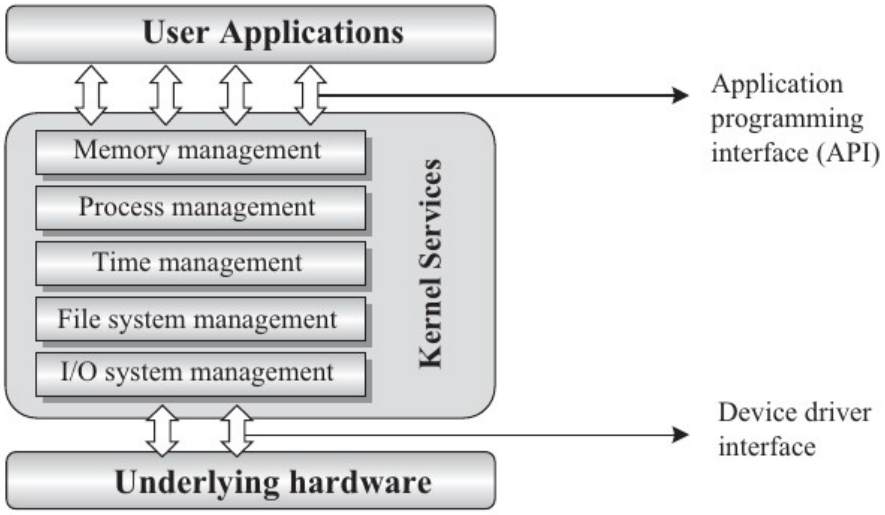
- The operating system is a bridge between the user applications and system resources (hardware)
- The OS manages the system resources and makes them available to the user applications/tasks on a need basis
- The primary functions of an operating system is:
  - Make the system convenient to use
  - Organise and manage the system resources efficiently and correctly

Kernal:

- The kernel is the core of the operating system
- It is responsible for managing the system resources and the communication among the hardware and other system services
- In a general purpose OS, the kernel contains different services to manage the following:

1. Process Management
2. Memory Management
3. File System Management
4. I/O System (Device) Management

Following figure shows the architecture of an operating system:



Functions of an operating system

1. *Process Management*: It deals with managing processes and tasks, such as:
   - Setting memory space for the process
   - Loading code to memory
   - Allocating system resources
   - Scheduling execution
   - Inter process communication and synchronization
   - Process termination

2. *Memory Management*: The memory management unit (MMU) of the kernel is responsible to
   - Keep the track of the each processes in memory area
   - Allocate and deallocate memory area for each process as needed

3. *File System Management:* The file system management service of Kernel is responsible for:
- The creation, deletion and alteration of files
- Creation, deletion and alteration of directories
- Saving of files in the secondary storage memory
- Providing a flexible naming convention for the files

4. *I/O System (Device) Management:* Kernel is responsible to:
- Maintain a list of all the I/O devices of the system
- Load or unload device drivers
- Route the I/O requests coming from different user applications to the appropriate I/O devices
- Communicate with I/O devices through device drivers.

Types of Operating Systems

Depending on the kernel services, operating systems can be broadly classified to two types:
1. General Purpose Operating System (GPOS)
2. Real-Time Operating System (RTOS)

General Purpose Operating System (GPOS)
- It is used in general computing systems
- Its kernel contains all services required for executing generic applications.
- It is normally non-deterministic in behaviour
- The services often results in random delays in application software that causes slow responsiveness of an application
- GPOS are generally used in Personal/Desktop computers
- Eg: Windows, Linux

Real-Time Operating System (RTOS)

A real time operating system is designed to execute the task within a specified deadline. It provides a predictable and deterministic behaviour for time-critical applications. The RTOS decides which applications should run in which order and how much time needs to be allocated for each application. Examples for RTOS are : MicroC/OS, VxWorks, QNX

Features of RTOS:
- Task are executed within specified deadline
- It provides predictable and deterministic performance
- Fast and responsive
- Used in time-critical applications
- Light weight compared to GPOS
- Real-Time kernel contains only the minimal set of services

Basic functions of a Real-time kernel:
- Task/Process management
- Task/Process scheduling
- Task/Process synchronisation
- Error/Exception handling
- Memory management
- Interrupt handling
- Time management

Real time systems can be classified in to two types:

1. Hard real time systems
2. Soft real time systems

Hard real time systems:

A Hard Real-Time system must meet the deadlines for a task without any slippage. Missing the deadline may produce catastrophic (cause sudden great damage) results including permanent data lose and irrecoverable damages to the system/users. Hard Real-Time systems emphasise the principle *'A late answer is a wrong answer'*. Eg: Air bag control systems in vehicles. When the air bag deployment task is triggered, the currently executing task must be pre-empted, the air bag deployment task should be brought into execution, and the necessary I/O systems should be made readily available for the air bag deployment task.

Soft real time systems:

Real-Time Operating System that does not guarantee to meet the deadlines are called Soft real time systems. But it offers its maximum effort to meet the deadlines. In this system, missing the deadlines for tasks are acceptable to an extend. A Soft Real-Time system emphasises the principle *'A late answer is an acceptable answer, but it could have done bit faster'*. Eg: ATM (Automatic Teller Machine)

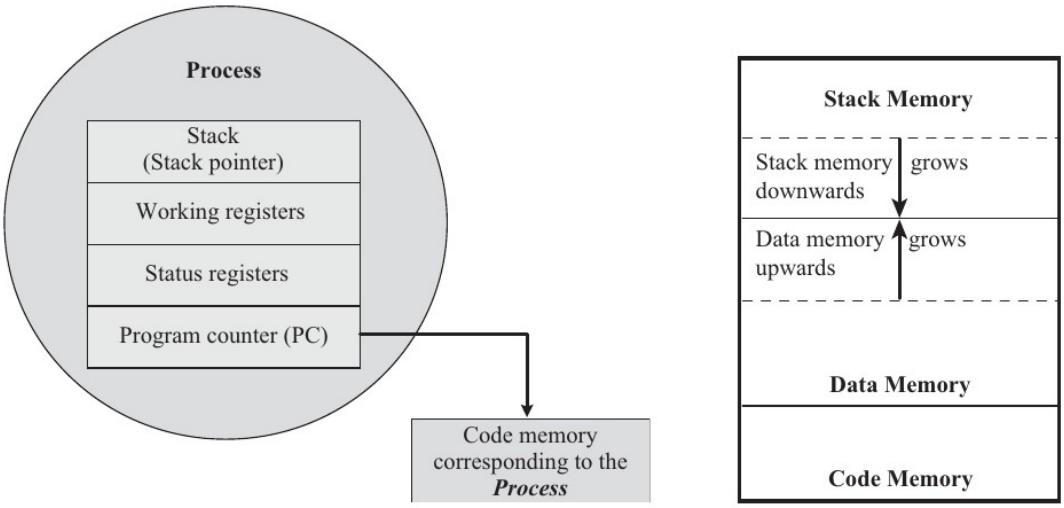
PROCESS, TASK, THREADS

PROCESS:

- A process is a program or part of a program in execution. Multiple processes (instances) of the same program can execute simultaneously.
- To execute a process it requires the system resources like CPU, memory, I/O devices etc.
- The concept of 'Process' leads to concurrent execution of tasks and thereby the efficient utilisation of the CPU and other system resources

Structure of a Process:

To achieve the concurrent execution, CPU is shared among processes. For this, a process keeps all properties of the CPU such as the contents of working registers, status register, PC etc. and is considered as a virtual processor. When the process gets it turn for execution, the properties kept by the process is mapped to the physical registers of the CPU. Following figure shows the structure of a process and its memory organisation.

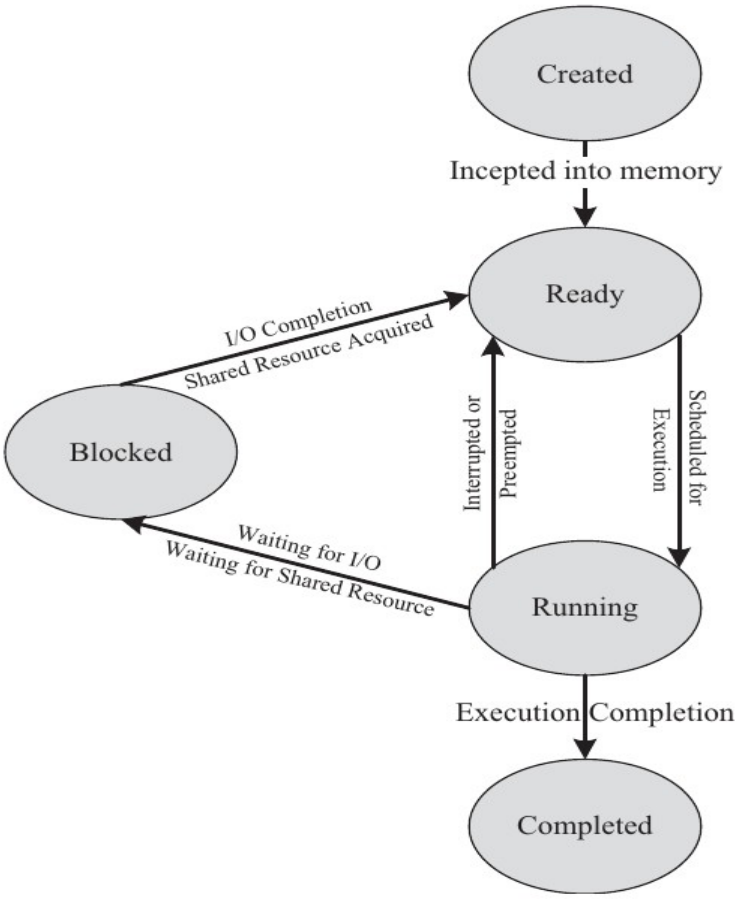


Process States and its Transition

A process has five states:

1. Created state
2. Ready state
3. Running state
4. Blocked state
5. Completed state

The cycle through which a process changes its state from 'created' to 'completed' is known as 'Process Life Cycle'. Each state in the life cycle indicate the current status of the process with respect to time and also provides information on what it is allowed to do next. Following figure shows the life cycle states and transition of a process.



Created state : The OS recognizes the process but no resources are allocated.
Ready state : The process is placed in the 'Ready list queue' and waiting for execution
Running state : Source code of the process is being executed
Blocked state : The process is temporarily suspended for accessing other resources
Completed state : The execution of the process is completed.