**CPU Protection:**

In addition to protecting I/O and memory we must insure that the operating system maintains control. We must prevent the user from getting stuck in an infinite loop or not calling system services, and never returning control to the operating system. To accomplish this goal, we can use a timer.

Timer can be set to interrupt the computer after a specified period. The period may be fixed (for example, 1/60 second) or variable (for example, from 1 millisecond to 1 second) A variable timer is generally implemented by a fixed rate clock and a counter.

We can use the timer to prevent a user of program from running too long. A simple technique is to initialize a counter with the amount of time that a program is allowed to run.

The most common use of timer is to implement time sharing. In the most case, the timer could be set to interrupt every N millisecond, where N is the time slice that each user is allowed to execute before the next user get control of the CPU. The operating system is invoked to perform housekeeping tasks.

This procedure is known as a context switching, after a context switch stage, the next program continues with its execution from the point at which it left off.

**Operating System Structure**

In the following lectures we will consider the components and services that are provided by different operating systems.

**System Components**

Many modern computer systems share the goal of supporting the following components:

**Process management**

A process can be thought of a program in execution. A process needs certain resources to accomplish its task. Also the process various initialization values.

A process is the unit of work in a system. Such a system consists of a collection of processes, some of which are system processes others are user processes. All processes execute concurrently by multiplexing the CPU among them.

The OS responsible for the following activities in connection with process management:

Creation and deletion both user and system processes.

Suspending and resuming processes.

Providing mechanisms for process synchronization.

Providing mechanisms for process communication.

Providing mechanisms for deadlock handling.

**Main Memory Management**

The main memory is the central to the operation of a modern computer system. For a program to be executed it must mapped to absolute addresses and loaded to the M.M.

The OS responsible for the following activities in connection with M.M management:

Keeping track of which parts of memory are currently being used and by whom.

Deciding which processes are to be loaded into memory when memory space become available.

Allocating and deallocating memory space as needed.

**File Management**

For convenient use of the computer, the OS provides a uniform logical view of information storage. The OS abstracts from the physical properties of its storage device to define the logical storage unit, the file.

A file is acollection of related information defined by its creator. These files are organized in directories to ease their use.

The OS responsible for the following activities in connection with file management:

Creating and deleting files.

Creating and deleting directories.

Supporting primitives for manipulating files and directories.

Mapping files onto secondary storage.

Backing up files on stable storage media.

**I/O System Management**

One of the purposes of OS is to hide the peculiarities of specific hardware devices. The OS responsible for the following activities in connection with I/O system management:

A memory management component that includes buffering, caching and spooling.

A general device driver interface.

Derivers for specific hardware devices.

**Secondary Storage Management**

The computer system must provide secondary storage to back up main memory because that are hold by MM are lost when power is switched of f and the MM is too small to accommodate all data programs. The OS responsible for the following activities in connection with disk management:

Free space management

Storage allocation

Disk scheduling

**Networking**

A distributed system collects physically separate heterogeneous system into a single coherent system, providing the user with the access to various resources that the system maintain. Access to a shared resource allows computation speed up, increase functionality, increase data arability, and enhance reliability.

**Protection System**

Protection is any mechanism for controlling the access programs, processes, or users to the resources defined by the computer system. This mechanism must provide means for specification of the controls to be imposed and means for enforcement. Protection can improve reliability by detecting latent errors at the interfaces between component subsystems.

**Command Interpreter System**

Command Interpreter System is the interface between the user and the OS. Some of these Command Interpreter System are user friendly such

as mouse based window and menus. In other shells commands are typed on a keyboard.

**Operating System Services**

An operating system provides an environment for the execution of programs. It provides certain services to programs and to the users of these programs. The specific services provided differ from one operating system to another but we can identify common classes. These operating system services are provided for the convenience of the programmer, to make the programming task easier.

1. Program execution

2. I/O operation

3. File system manipulation

4. Communications

5. Error detection

6. Resource allocation

7. Accounting

8. Protection

**System Calls**

System calls provide the interface between a process and the operating system. These calls are generally available as assembly language

instructions and they are usually listed in the various manuals used by assembly language.

**System Programs**

System programs provide a convenient environment for program development and execution. Some of them are simply user interfaces to system calls others are considerably more complex. They can be divided into these categories:

File management

Status information

File modification

Programming language support

Program loading and execution

Communications

**System Structure**

A system as large and complex as a modern operating system must be engineered carefully if it is to function properly and to be modified easily. There are three different system structures:

Simple structure

Layered Approach

Microkernal

**System Design and Implementation**

The problems and steps of system design and implementation are as follows:

Design Goals

Mechanisms and Policies

Implementation