**UNIT-1**

**Overview of Operating System**

**Introduction:**

An operating system is a program that manages a computer’s hardware and provide a basis for application programs and act as an intermediary between the computer user and the computer hardware. Operating systems are vary based on accomplishing the various tasks and system. Some operating system are designed to optimize the utilization of hardware, some are design for supporting complex games, business application and some are designed to provide simple interface for the user. Therefore, some operating system are designed to be convenient, other to be efficient and other to be some combination of the two.

Operating system exist because they offer a reasonable way to solve the problem of creating a usable computing system. The fundamental goal of an operating system is to execute user programs and to solve the user problem. To accomplish this goal computer hardware and application software are developed. These program require certain common operation such as those controlling the I/O device. Thus the common function of controlling and allocating resources are then brought together into one piece of software which is known as the operating system.

An operating system is the one program running at all times on the computer which is known as **kernel.** Along with the kernel, there are other two types of program: system program which are associated with the operating but are not usually part of kernel and application program which includes all the program not associated with the operation of the system.

**Operating System Goals:**

* Execute the user program and to solve the user problem easily.
* Make the computer system convenient to use.
* Use the computer hardware in efficient manner.
* Proper utilization of the resources.

**Types of Operating System:**

Operating system varies according to the uses of the computing environment.

1. Batch Operating System:

In the latter half of the 20th century, computing resources were relatively scarce. For a period of time, system were either batch or interactive. Batch system processed the job in bulk with predetermined input from file or other data source. Each users do not interact with computer directly. They prepare the job which consists of program, data and some control information about the nature of the job and submit to the operator. The operator then sorts the program with the similar requirements into batches. To speed up the processing, job with the similar needs are batched together and run as a group. Here the operator function as the OS.

Advantages:

* Low setup cost over more job.
* Keeps machine busy while programmer thinks of problem and managing job.

Disadvantages:

* Lack of interaction between user and the job.
* Difficult to provide desired priority.
* If bugs are received, memory and register are dumped then job have to be submitted again.
* CPU is often idle because the speed of IO device is less than CPU processing speed.
* User have to wait for result until batches are collected and submitted.

1. Time Sharing Operating System:

Time sharing system used a timer and scheduling algorithm to cycle processes rapidly through the CPU, giving each user a share of the resources. Processor’s time which is shared among multiple users simultaneously is termed as time sharing. Here the CPU executes multiple jobs by switching among them, but the switches occurs so frequently that the user can interact with each program while it is running. The main objective of time sharing system is to minimize the response time. The operating system uses the CPU scheduling and multiprogramming to provide each user with a small portion of time. User processes and the system processes that provide the service to the user are managed such that each processes gets a slice of computer time.

Advantages:

* Provide quick response.
* Reduce CPU Idle time

Disadvantages:

* More complex job scheduling and memory management.
* Problem of reliability.
* Problem of data communication

1. Distributed Operating System:

A distributed system is a collection of physically separate, possibly heterogeneous computer system that are networked to provide user with access to the various resources that the system maintains. It uses multiple central processors to serve multiple real time application. Data processing jobs are distributed among the processors accordingly. The processors communicates each other through various communication line such as buses. Some operating system generalizes the network access as a form of file access with the detail of the networking contained in the network interface device driver and other makes users specifically invoke network function. Generally the system contains mix of the two. It provides less autonomous environment.

Advantages:

* Due to resource sharing facility user at one site may be able to use the resource available at another site.
* If one site fail, the remaining site can potentially continue operating.
* Reduction in delay in data processing.

Disadvantages:

* Harder to manage network.
* Provide less autonomous environment.

1. Network Operating System:

It is an operating system that provides feature such as file sharing across the network along with a communication scheme that allow different processes on different computers to exchange message. It runs on the server and provide the capability to manage data, users, security, application and networking function. A computer running a network operating system acts autonomously from all other computers on the network although it is aware of the network and is able to communicate with other networked computers.

Advantages:

* Security is server managed.
* Provides autonomous environment.
* Remote access to server is possible from different location and types of the system.

Disadvantages:

* High cost on managing server
* Dependency on a central location for most operation.
* Regular maintenance is required.

1. Real Time Operating System:

A real time system is used when rigid time requirement have been placed on the operation of a processor or the flow of data. A real time system has well defined, fixed time constraints and processing must be done within that time limit otherwise system is consider to be fail. **It is the data processing system in which time interval to process and responds the input is small i.e. service is provided in less time.** Real time operating system are mostly used by the embedded system (such system which are designed to perform specific function and are controlled by a specific program for the operation).

Advantages:

* Fast response and processing time

Disadvantages:

* Fixed time constraints.

1. Open and Closed Source Operating Systems:

Open source operating system are those available in source code format rather than as compiled binary code. For e.g. Linux is the most famous open source operating system while Microsoft windows is a well-known example of closed source operating system. Apple’s Mac and IOS operating system comprises hybrid approach. Open source code is more secure than close source code because the code are view repeatedly.

**Function of an Operating System:**

1. Resource Management:

Operating system manages and protects all the resources that are attached to the system such as internal external memory, tasks, application, communication channel etc., handles and allocates resources to multiple users or multiple program running at the same time and also manages conflicting request for efficient and fair resource use (maximize throughput and minimum response time). The operating system will identify at which time which operation will be performed by the CPU, in which time memory is used by the program and which device will be responsible to process the incoming request.

1. Memory Management:

The main memory is the central to the operation of a modern computer system. Main memory is a large array of bytes and each byte has its own address. The central processor reads the instruction from the main memory during instruction fetch cycle and both read and write data from main memory during data fetch cycle. To improve the CPU utilization and the speed of the computer’s response time all the program should be kept in memory which required proper memory management.

Thus, operating system is responsible for the following activities for memory management:

* Keeping track of which part of memory are currently being used and who is using them.
* Deciding which processes or part of the processes and data to move into and out of memory.
* Allocating and deallocating memory space as needed.

1. Process Management:

A program does nothing unless its instruction are executed by a CPU. A program loaded in the memory and executing (i.e. a program in execution) is known as **process**. For e.g. a word processing program being run by an individual user on a PC is a process. A program is the passive entity, like the contents of a file stored on disk whereas a process is an active entity.

A process can be single threaded and multi-threaded and depending on this their execution and resource requirement are different. There might be several process executing concurrently and the resources are either given when it is created or allocated when it is running. So, for the proper execution of the process their proper management is required. Thus, the operating system is responsible for the following activities in connection with process management:

* Scheduling process and threads on CPU
* Creating and deleting both the user and system process.
* Resuming and suspending processes.
* Providing mechanism for process communication.

1. Storage Management:

To make the computer system convenient for users, the operating system provides a uniform, logical view of information storage. A file is a collection of related information defined by its creator. The operating system maps the file onto the physical media and access these files via the storage device. Computer can store information on several different types of physical media like magnetic tape and each device has its own physical organization. The operating system implements abstract concept of file by managing mass-storage media such as tapes and disks. Files are normally organized into directories to make them easier to use and when particular file is accessed by multiple user it may be desirable to control which user may access a file. The operating system handles both file system and mass storage system.

Following are the function of operating system in connection to file system and mass storage management:

* Creating and deleting file
* Creating and deleting directories
* Mapping file into secondary storage.
* Backing up file on stable storage media
* Free space management (mass storage management)
* Disk scheduling (mass storage management)
* Storage allocation (mass storage management)

1. Protection and Security:

Protection is any mechanism for controlling the access of processes or user to the resources defined by a computer system. This mechanism must provide means to specify the controls to be imposed and to enforce the control. If a computer has multiple users and allows concurrent execution of multiple processes then access to data must be regulated. Mechanism must ensure that resources can only be operated by those process that have gain proper authorization by operating system. By means of password and other techniques operating system prevents unauthorized access to program and data.

A system can have adequate protection but still be prone to failure and allow inappropriate access. For e.g. if a user’s authentication information is stolen then his/her data could be copied or deleted although protection is working. So, it is duty of security to prevent from external and internal attacks (e.g. virus, worms, denial of service attack etc.). Prevention of some of these attacks is considered to be an operating system function in some system while other system leave it to policy or additional software.

**Computer System Organization and Operation:**

A modern general purpose computer system consist of one or more CPUs and a number of device controller connected through a common bus that provide access to the shared memory. Each device control is in charge of a specific type of device for e.g. audio device, disk drives, video display etc.). The CPU and device controllers can execute in parallel, competing for memory cycle. To ensure orderly access to shared memory, a memory controller synchronizes access to the memory.

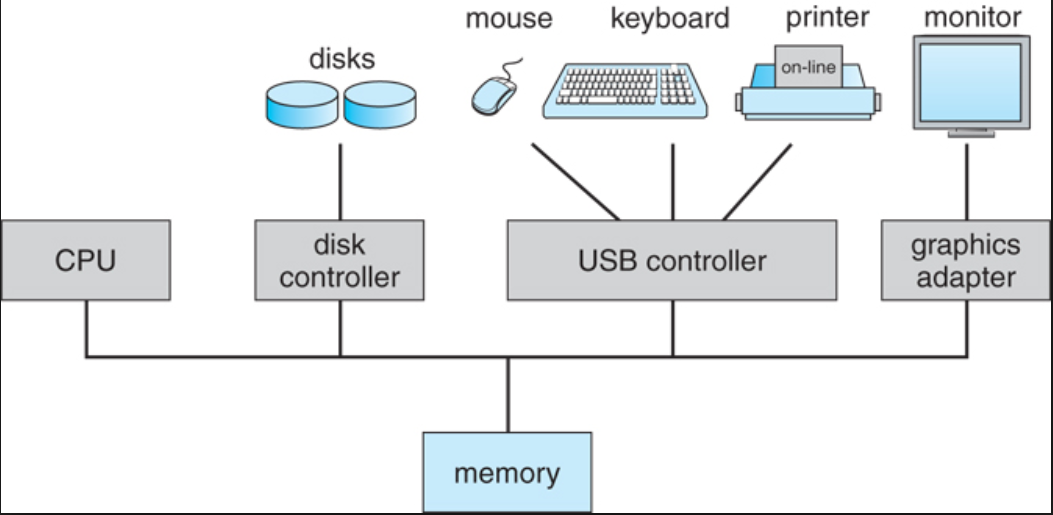


Figure: Structure of modern computer system

When the computer is powered on or start running, an initial program is run known as bootstrap program which is stored within the computer hardware in ROM. Bootstrap program initialize all the aspect of the system from CPU register to device controller to memory content. To load the operating system and to start executing the system, the bootstrap program must locate the operating system kernel and load it into memory. Once the kernel is loaded and executing. It can starts providing service to the system and its user.

The occurrence of any event is signaled by an interrupt from either the hardware or the software. When the CPU is interrupted, it stops what it is doing and transfer the execution to a fixed location which contains the starting address where the service routine for an interrupt is located. Then the interrupt service starts and on completion CPU resumes the interrupted computation.

**Operating System Structure and Operation:**

Operating system provides the environment within which program are executed. Many operating system are interrupt driven. If there are no any process to execute, no I/O device to service and no any users to respond then operating system sits idle waiting for event. Events are always signaled by the occurrence of interrupt or a trap. A **trap or exception** is a software generated interrupt caused either by an error (for e.g. division by zero) or by specific request from user program. For each type of interrupt, separate segments of code in the operating system determines what action should be taken. To deal with the interrupt, interrupt service routine are provided. A properly designed operating system must ensure that an incorrect or malicious program cannot cause other programs to execute incorrectly.

In multiprogramming system, the operating system keeps the several jobs in memory simultaneously as shown in figure below. Generally, main memory is too small to accommodate all the jobs so, the jobs are kept on the disk in the job pool. Job pool consist of all the processes residing on the disk awaiting allocation of main memory. The operating system picks and begin to execute one of the jobs in memory. If the jobs need to wait for some other task like I/O operation then the CPU switches to another job and so on.



In the time-sharing system, the CPU executes multiple job by switching among them. Here, the user gives instruction to the operating system using input devices and wait for immediate result on an output devices. So the response time should be short typically less than one second. The operating system uses the CPU scheduling and multiprogramming to provide each user with a small portion of time shared computer. Operating system also uses memory management, job scheduling, CPU scheduling for managing all the jobs and loading them into memory.

**System Structure or Two Views of Operating system:**

**Operating System Service:**

An operating system provides an environment for the execution of programs. I provides certain service for the programs and for the user of the program. The service provided are differ from one operating system to another. Following figure shows the one view of various operating system service and how they are interrelate. Here, service provided are divided into two set, one which is helpful for the user and another one is for ensuring efficient operation of the system itself.

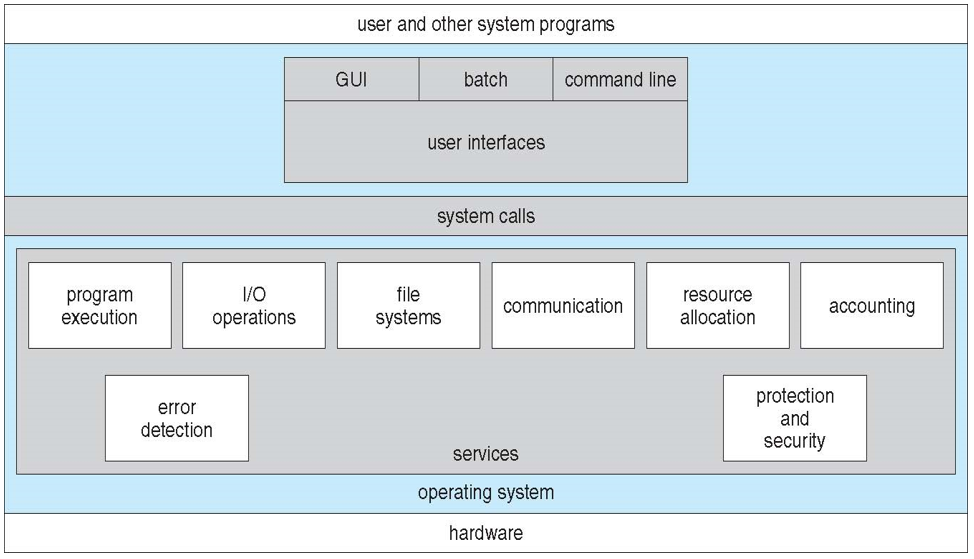


Figure: A view of operating system services

Following are the operating system service that are helpful to the user:

1. User Interface:

Almost all operating system have a user interface which can take several forms. One is command line interface (CLI) which uses the text command and a method for entering them. Another is batch interface, in which commands and directives to control those command are entered into files and executed. Another is a graphical user interface (GUI) which is commonly used now days and is a window system with pointing device to direct I/O, choose from menus, makes selection and keyword to enter text.

1. Program Execution:

The system must be able to load a program into memory and to run that program. The program must be able to end the execution either normally or abnormally. These all are controlled by operating system.

1. I/O Operation:

A running program may require I/O operation which may involve file or an I/O device and for specific device, special function may require. For efficiency and protection user cannot control I/O devices directly so, operating system must provide means to do I/O.

1. File-System Manipulation:

Program needs to read and file and directories. They also need to create and delete them, search them, list the information etc. some operating system includes permission management to allow or deny access to files or directories based on ownership where some provides a variety of file system to allow personal choices and to provide specific features.

1. Communication:

There are many situation in which one process needs to exchange the information to another process and such communication may occur between the processes that are executing in same computer or executing on different computer tied together by some form of network. Communication may be implemented via **shared memory** in which two or more processes read and write to shared section of memory or **message passing** in which packet of information in predefined formats are moved between processes by an operating system.

1. Error Detection:

The operating system needs to be detecting and correcting errors constantly. Error may occur in the CPU and memory hardware (such as power failure), in I/O device (such as connection failure to network, lack of paper in printer) and in the user program (such as attempt to access an illegal memory, over uses of CPU time). For each type of error, operating system should take the appropriate action to ensure correct and consistent computing. Sometimes it might terminate an error causing process or it might halt the system if it has no choice.

Following are the set of operating system function exists for ensuring the efficient operation of the system:

1. Resource allocation:

When there are multiple users or multiple jobs running at the same time, resource must be allocated to each of them. The operating system manages different types of resources. Some (such as CPU cycle, main memory) may have special allocation code whereas other (such as I/O devices) may have general request and release code. For the uses of best CPU time, operating system have CPU scheduling which takes different aspect like speed of CPU, number of register available etc.

1. Accounting:

Operating system keep track of which user uses how much and what kinds of computer resources. This recording may be used for accounting for accumulating usage statistics. These records may be valuable tool for researcher who wish to reconfigure the system to improve computing service.

1. Protection and Security:

Protection involves ensuring that all access to the system resource is controlled. Security of the system from outsiders is also important and security is provided by means of password to authenticate user for the access to the system. This helps to defend from unwanted access from third party.

**User and Operating System Interface:**

There are several ways for user to interface within the operating system. Here, we discuss two fundamental approaches. One is a command line interface of command interpreter which allow user to directly enter the commands to be performed by the operating system and another one allows a user to interface with operating system via a GUI.

* Command Interpreters:

Some operating system include the command interpreter in the kernel while other (window and UNIX) treat the command interpreter as a special program that is running while a job is initiated. On the system with multiple command interpreter the interpreter is known as shell. The main function of command interpreter is to get and execute the user specified command and many of the command given at this level manipulate the file like create, delete, list, print etc.

These commands are generally implemented in two ways. In one approach, the command interpreter itself contains the code to execute the command. Here the command causes the interpreter to jump in the section of code that setup the parameter and make appropriate system call. In another approach, (used by UNIX) implement most of the command through system program. In this case command interpreter does not understand any command and merely uses the command to identify a file to be loaded into memory.

* Graphical User Interface (GUI):

Here, rather than entering the command, user employs mouse based widow and menu system characterized by a desktop metaphor. A user uses mouse to point to the application by clicking to the icon in the screen. Microsoft’s first version of window was based on the addition of a GUI interface to the MS-DOS operating system.

Smartphones and handheld tablets uses a touch screen interface and users interact by making gestures on the touchscreen interface for e.g. pressing and swiping the figures. Traditionally, UNIX system were dominated by command line interface. Nowadays, UNIX and Linux have command line interpreter with optional GUI interfaces.

**Operating System Structure:**

An operating system must be engineered carefully to function properly and to be modified easily. A common approach is to partition the task into small component or module and should be well define portion of the system with clearly defined input, output and functions. Following are the structure of operating system.

1. Simple Structure:

Many operating system do not have well defined structures. Such system started as small, simple and limited systems and then grew beyond the original scope. MS DOS is an example of such system which was written to provide the most functionality in the least space so it was not carefully divided into modules. The interface and level of functionality are not well separated. Application program are able to access the basic I/O routines to write directly to disk driver due to which it is vulnerable from malicious attack causing whole system to be failed when user’s program fails. It was also limited by the hardware. Following figure shows the structure.

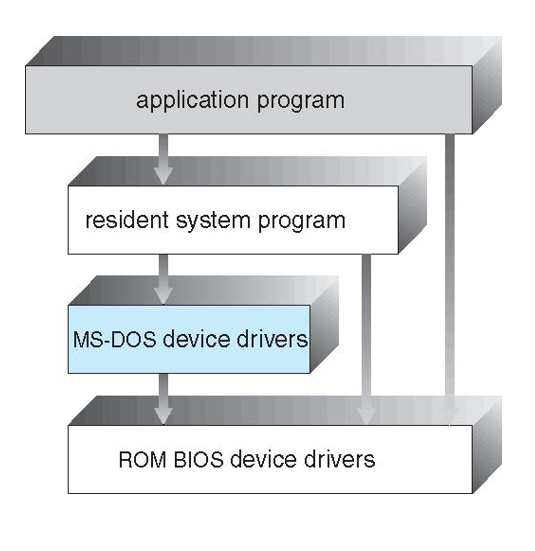


Figure: MS-DOS Layer Structure

Another example of limited structuring is the original UNIX operating system. UNIX initially was limited by hardware functionality and consist of two separate parts kernel and the system programs. The kernel is further separated into a series of interfaces and device drivers. Traditional UNIX system structure is shown in figure below:

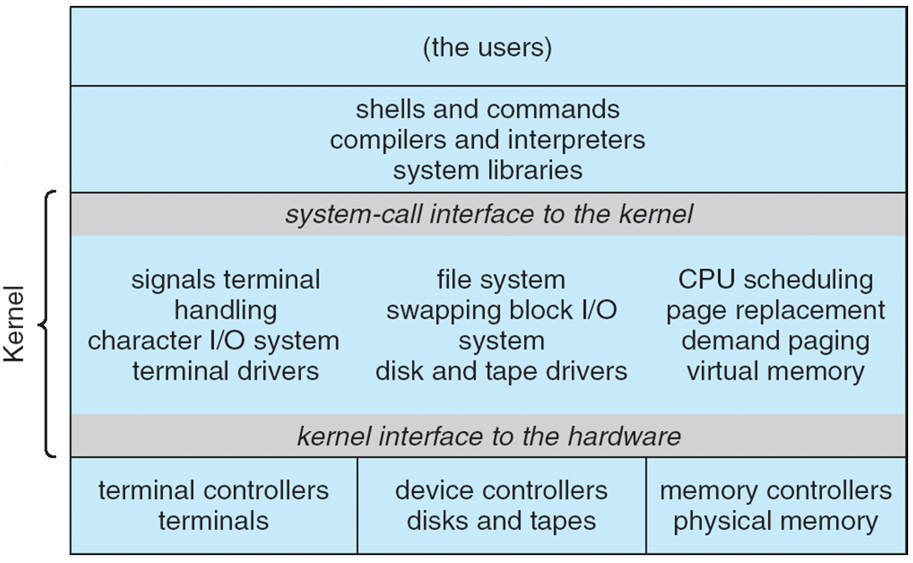


Figure: Traditional UNIX system structure

Everything below the system call interface and above the physical hardware is the kernel and provides the file system, CPU scheduling, memory management etc. through system call. This bulk structure was difficult to maintain and also the overhead in the system call interface or in communication within kernel.

1. Layered approach:

Operating system can be broken down into smaller form with proper hardware support due to which much more control over the computer can be retained. A system can be made modular in many ways and one method is the layered approach in which operating system is broken into a number of layers (levels). The bottom layer is the hardware and the highest layer (layer n) is the user interface. The layer consists of the data structure and set of routines that can invoked by the higher level layers (i.e. higher layer invoke operation on lower layer). The layers only knows what a particular operations do, it does not need to know how the operation are implemented.

Following figure shows the structure of layered approach.

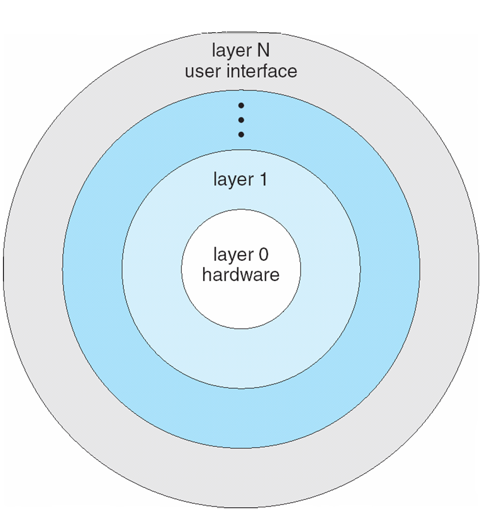


Figure: A layered operating system

1. Microkernels:

This method structures the operating system by removing all non-essential components from the kernel and implementing them as a system and user level program. This results in smaller kernel and provide minimum process and memory management. The main function of microkernel is to provide communication between the client program and the various service that are also running in the user space. Communication is provided through message passing (message is exchanged between the process either directly or indirectly through a common mailbox. The client program and service never interact directly. They have to communicate indirectly by changing message with the microkernel.

The advantage of this structure is that it makes extending of OS easier. All services are added to user space and do not require modification of the kernel which makes changes to be fewer. It also provide more security and reliability since, most of the service are running as a user and if a service fail the rest of the OS remain untouched.

The following figure shows the structure of microkernel approach.

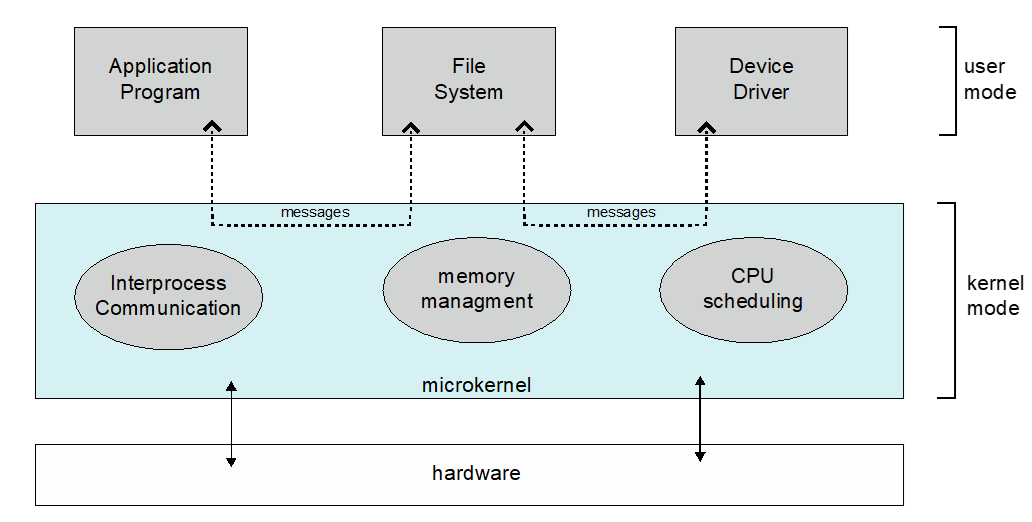


Figure: Structure of microkernel

**System Calls:**

System calls provide an interface to the services made available by an operating system. These calls are generally available as routines written in C and C++. System executes thousands of system call per second. Following example illustrate how system call are used.

Let us consider we need to write a simple program to read data from one file and copy them to another file. For this the program will need the name of the files at first which can be specified in many ways. In mouse based system it can be done by choosing the file name displayed in window. This sequence requires many I/O system call.

Once the file name have been obtained, the program must run the input file and create output file. This operation also required another system call. If the error occur on processing the input file program should respond with error message, which requires another system call. When the both files are set up we enter in a loop that reads from the input file and writes to the output file (another system call).

Each read and write operation must return status information regarding various possible error conditions. This required another form of system call. Finally, after all the entire file is copied program may closed both files and return the status information, which requires another system call. This sequence are shown in figure below.

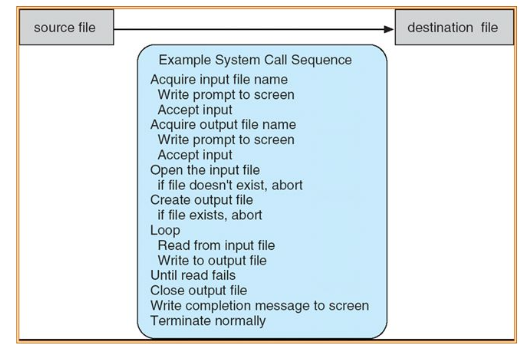


Figure: Example of how system call are used

Following table shows the types of system calls provided by operating system:

|  |  |  |
| --- | --- | --- |
| Types | For Windows | For UNIX |
| Process Control | CreateProcess()  ExitProcess()  WaitForSingleObject() | Fork()  Exit()  Wait() |
| File Manipulation | CreateFile()  ReadFile()  WriteFile()  CloseHandle() | Open()  Read()  Write()  Close() |
| Device Manipulation | SetConsoleMode()  ReadConsole()  WriteConsole() | Ioctl()  Read()  Write() |
| Information Maintenance | GetCurrentProcessID()  SetTimer()  Sleep() | Getpid()  Alarm()  Sleep() |
| Communication | CreatePipe()  CreateFileMapping()  MapViewOfFile() | Pipe()  Shm.open()  Map() |
| Protection | SetFileSecurity()  InitializeSecurityDescriptor()  SetSecurityDescriptor() | Chmod()  Unmask()  Chown() |
|  |  |  |