1

Introduction

1.1 NEED OF OPERATING SYSTEM

Computer is an electronic device and its organization includes two main components, that is, hardware and software. All the hardware components work on the basis of software. The most basic or fundamental software is an operating system. An operating system is a program that manages the computer hardware. It also provides a basis for application program and acts as an intermediator between the computer user and computer hardware. It provides the working environment to execute a program in convenient and efficient manner. Operating system is responsible for overall functioning of machines from booting to resource management. An operating system does the following:

- Manages all the devices
- Provides interface between the user and the system
- Manages the memory allocation and deallocation to process
- Schedules the computational processes
- Manages the files and disks
- Protects the system and data
- Helps resources allocation and management
- Shares of resources in terms of time, space and CPU to achieve optimal utilization of resources.

1.2 OBJECTIVES OF OPERATING SYSTEM

An operating system is a program that controls the execution of application programs and acts as an interface between application and computer hardware. Operating system has three main objectives:

- **Convenience:** An operating system makes a computer more convenient to use.
- **Efficiency:** Allows the computer system resources to be used in efficeint manner.
- **Ability to Evolve:** An operating system should be constructed in such a way as to permit effective development, testing and introduction of new system functions without interfering with service.

Apart from the above three objectives, other objectives are:

- Increase productivity of computing resources.
- Facilitate the use of hardware.
- Optimal scheduling of computational activities to ensure good performance of system.
- Creation of user-friendly environment for development and execution of program.

1.3 OPERATING SYSTEM AND ITS FUNCTIONS

An operating system is a program that acts as an intermediary between the user of the computer and the computer hardware. The purpose of an operating system is to provide an environment in which the user can execute programs in a convenient and effcient manner.

The operating system must ensure correct operation of the computer system.

Mainframe operating systems are designed primarily to optimize utilization of hardware. Personal computer (PC) operating systems support complex business applications, games and everything in between.

A computer system can be roughly divided into four components:

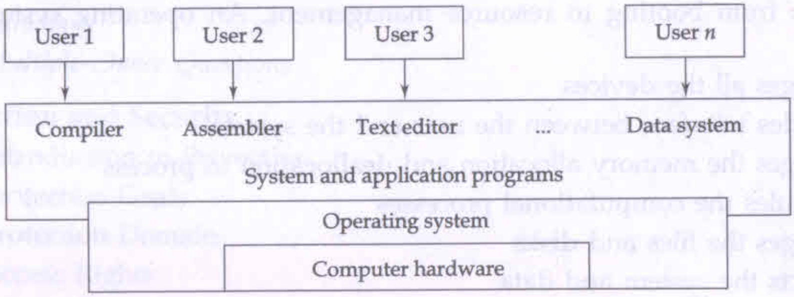

Fig. 1.1 Components of a computer system

1.4 VIEWPOINTS ON OPERATING SYSTEM

Operating system can be explored from two viewpoints:

1.4.1 User View

It varies by the interface being used. Most computer users sit in front of a PC, consisting of a monitor, keyboard, mouse and system unit. Such a system is designed for one user to monopolize its resources, to maximize the work that the user is performing. OS is mostly designed for *ease of use*. Some users sit at the terminal connected to a mainframe or minicomputer. Other users access the same computer through other terminals. These users share resources and may exchange information. OS is designed to maximize *resource-utilization*. Other users sit at workstations, connected to networks of other workstations and servers. These users have dedicated resource at their disposal, but they also share resources such as networking and servers – file compute and print servers. Thus, OS is designed to *compromise between individual usability and resource utilization*.

1.4.2 System View

From the computer's viewpoint, the OS is a program that is most intimate with the hardware. We can view an OS as a *resource allocator*. A computer system has many sources, hardware and software that may be required to solve a problem.

OS acts as the manager of CPU time, memory space, file storage space, I/O devices and so on. Facing numerous and possibly conflicting requests for resources, the OS must decide how to allocate them to specific programs and users so that it can operate the computer system efficiently and fairly.

An OS is a control program. A control program manages the execution of user programs to prevent errors and improper use of the computer. It is especially concerned with the operation and control of I/O devices.

1.5 FUNCTIONS OF OPERATING SYSTEM

The primary goal of a operating system is to provide convenience and efficeincy.

It keeps a track of who is using which resource, to grant resource request, to account for usage, and to meditate conflicting requests from different programs and users.

The OS is also responsible for security, ensuring that unauthorised users do not access the system. Following are the various functions of an OS:

1. **Device Interfacing:** OS should try and hide the complexity of interfacing to devices from the user program and the user. It should also try and configure the device to start up rather than getting the user to set them up.
2. **File-System:** An OS can create and maintain a file system, where users can create, delete and move files around a structured file system.

   Many systems organize files, in directories. In multi-users systems, these folders can have associated user.
3. **User:** This allows one or more users to log into a system. Thus, the OS must contain a user account database. Which contains user name, default home directory user passwords and user rights.
4. **Multiprocessing:** This allows two or more processes to be used at a time. Here the OS must decide if it can run the different processes on individual processors. It must also manage the common memory between processors.
5. **Memory Management:** This involves allocating, and often create a virtual memory for program. Paging which means organizing data so that the program data is loaded into pages of memory. Another method of managing memory is swapping. This involves swapping the content of the memory of disk storage.
6. **Multithreading:** Processes are often split into smaller tasks named threads. This thread allows smoother operations.

1.6 CLASSIFICATION OF OPERATING SYSTEMS

1.6.1 Batch Systems

Earlier the common input devices were card readers and tape drivers. The common output devices were line printers, tape drivers and card punches. The user did not interact directly with the computer systems. Rather, the user prepared a job which consisted of a program, data and some control information about the nature of job and submitted it to the computer operator. The job was usually in the form of punch cards.

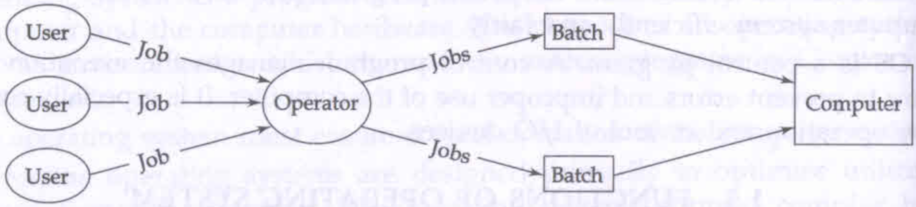

Fig. 1.2 Batch operating systems

1.6.2 Interactive Computer Systems

The interactive computer system is the system which provides direct communication between the user and computer. The user gives instructions to the operating system with input devices and waits for immediate response. When the OS finishes excecution of one command, it adds the next control statement, not from the card reader but from the keyboard in the form of command. Most systems have interactive editor for entering the program and then interactive debugger for assigning in debugging the programs.

Example: Programs like word processsors and spread sheet applications are interactive.

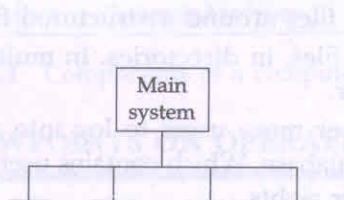

Fig. 1.3 Interactive system

1.6.3 Multiprogramming Operating Systems

When two or more programs are in memory at the same time showing the processor it is referred to as multiprogramming operating system. Generally, in multiprogramming a single processor is shared. It increases CPU utilization by organising jobs so that CPU always has one to excecute. The operating system keeps several jobs in the memory at a time and picks and begins to excecute one of the jobs in the memory.

It provides resource utilisation, but less user interaction.

Advantages

1. High CPU utilisation
2. It appears that many programs are alloted to CPU almost simultaneosly.

Disadvantages

1. CPU scheduling is required
2. To accommodate many jobs in memory, memory managenment is required.

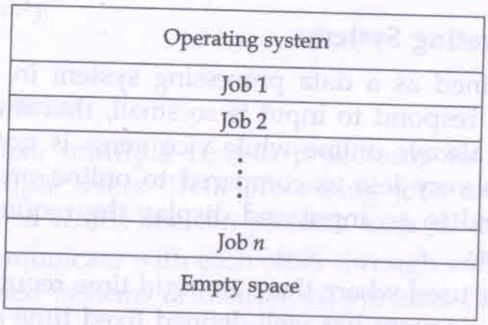

Fig. 1.4 Memory layout of OS

1.6.4 Multitasking Operating Systems

In multitasking operating system multiple programs can be run at a time. The operating system determines which application should run in what order and how much time is allowed for each application before giving another application a turn. Various processes share common processing resources.

Advantages

1. Several applications works simultaneously.
2. To the developer, the advantage is the ability to create applications that use more than one process and to create proceses that use more than one thread of excecuting.

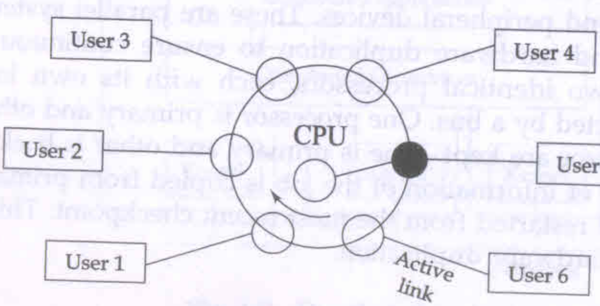

Fig. 1.5 Multitasking OS

1.6.5 Time Sharing Operating Systems

Time sharing is a technique which enables many people, locaed at various terminals to use a particular computer system simultaneously. Processor's time which is shared among multiple users simultaneously is termed time shaving. The main objective of time sharing system is to minimize response time.

Operating system uses CPU scheduling and multiprogramming to provide each user a small portion of time. Time sharing provides direct access to large number of users where CPU time is divided among all the users on scheduled basis. OS allocares time to each user and when time expires, it passes control to rent user on the system. This short period of time is known as *time since* or *time quantum*. In Fig. 1.4 time sharing system is also shown.

1.6.6 Real-Time Operating Systems

Real-time system is defined as a data processing system in which the time interval required to process and respond to input is so small, that it controls the enviroment. Real-time processing is always online while vice versa is not true always. So in this system response time is very less as compared to online processing. The time taken by the system to respond to an input and display the required updated information is known as *response time*.

Real-time systems are used where there are rigid time requirements on the operation of a processor. Real time system has well-defined fixed time constraints otherwise the system will fail.

Ward Real-time System: Ward real-time systems guarantee that critical tasks are complete on time. Its application areas are flight control computer (FCC), dynamic positioning system (DPS) found in marine vessels, etc.

Soft Real-time System: Soft real-time are less restrictive. Critical real-time task gets priority over other tasks and the systems retain priority until the jobs are complete. This has limited utility in comparison to ward real-time system, for example, multimedia, virtual reality, advanced scientific projects like sea exploration, etc.

1.6.7 Parallel System

Recently there is a trend towards multiprocessor systems. Such systems have more than one processor in close comunication, sharing the computer bus, the clock and sometimes memory and peripheral devices. These are parallel systems. Some systems use both software and hardware duplication to ensure continuous operation. Such systems consist of two identical processors, each with its own local memory. The processors are connected by a bus. One processor is primary and other is backup. Two copies of each processor are kept—one is primary and other is backup. At some fixed checkpoints the state of information of the job is copied from primary to backup. The copy is activated and restarted from the most recent checkpoint. This is expensive and needs considerable hardware duplication.

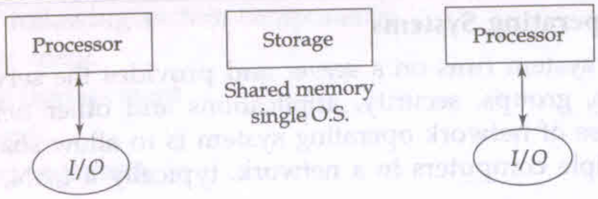

Fig. 1.6 Parallel system memory configuration

Advantages of a multiprocessor are parallel systems are:

1. Increased throughput
2. Save money by sharing peripherals, cabinets and power supplies.
3. Increased reliability
4. Fault-tolerant

1.6.8 Distributed Systems

Distributed systems use multiple central processors to serve multiple real-time applications and multiple users. Data processing jobs are distributed among processors accordingly to which one can perform each job most efficiently.

The processors communicate with each other through various communication lines. These are loosely coupled systems or distributed systems. Processors may vary in size and function.

Advantages of distributed system:

1. With resource sharing the user at one site may be able to use the resources available at another.
2. Speed up exchange of data with one another.
3. If one site fails in distributed system, the remaining sites can potentially continue operating.
4. Better services to customers.
5. Reduction of load on the main computer.

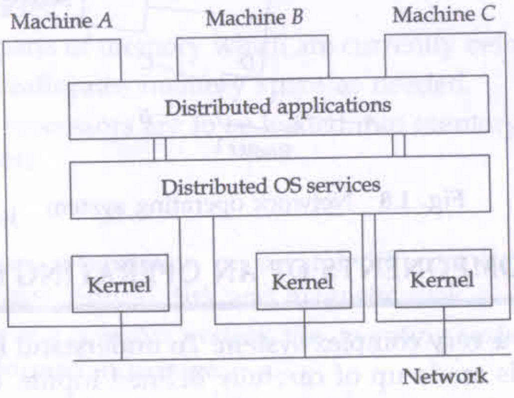

Fig. 1.7 Distributed system

1.6.9 Network Operating Systems

Network operating system runs on a server and provides the server the capability to manage data, users, groups, security, applications and other networking functions. The primary purpose of network operating system is to allow shared file and pointer access among multiple computers in a network, typically a LAN, private network or other network.

Advantages

1. Centralised servers are highly stable
2. Security is server managed
3. Upgrades to new technologies and hardware can be easily integrated into the system
4. Remote access to the server is possible from different locations and types of systems.

Disadvantages

1. High cost of buying and running a server
2. Dependency on a centeral location for most operations
3. Regular maintenance and updates are required.

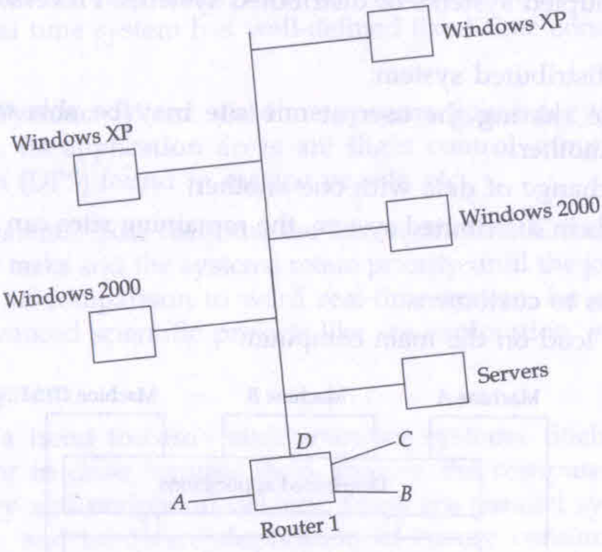

Fig. 1.8 Network operating system

1.7 COMPONENTS OF AN OPERATING SYSTEM

An operating system is a very complex system. To understand it, we will divide it into smaller pieces. An OS is made up of carefully defined inputs, outputs and functions.

It is made up of following system components:

1. Process management
2. Main memory management
3. File management.
4. I/O system management
5. Secondary storage management.
6. Networking
7. Protection system.
8. Command interpreter.

1.7.1 Process Management

(i) A process can be thought of as a program in execution, e.g., a batch job is a process.

(ii) A process needs certain resources including CPU time, memory, files and I/O devices to accomplish its task. These resources are either given to the process when it is created or allocated to it while it is running.

(iii) When the process terminates, the OS will reclaim any reusable resources.

(iv) A process is the unit of work in a system. Such a system consists of a collection of processses, some of which are OS processes and the rest of which are user processes.

1.7.2 Main Memory Management

(i) Main memory is a large array of words or bytes, ranging in size from hundreds of thousands to hundreds of millions.

(ii) Main memory is a repository of quickly accessible data shared by the CPU and I/O devices.

(iii) The central processor reads instructions from the main memory during the instruction-fetch-cycle and both reads and writes data from the main memory during the data-fetch-cycle.

(iv) The operating system is responsible for the following activities in connection with memory management.

- Keeps track of parts of memory which are currently being used and by whom.
- Allocates and deallocates memory space as needed.
- Decides which processors are to be loaded into memory when memory space becomes available.

1.7.3 File Management

(i) Computers can store information on several different types of physical media such as magnetic tape, optical disk and magnetic disk.

(ii) For convenient use of a complex system, the operating system provides a uniform logical view of information storage.

(iii) The operating system abstracts from the physical properties of its storage devices to define a logical storage unit, the file.

(iv) A file is a collection of related information defined by its creator. Files represent programs and data. A file consists of a sequence of bits, bytes, lines or records whose meanings are defined by their creator.

(v) Files are normally organised into directories to ease their use. When multiple users have access to files, it may be desirable to control by whom and in what ways files may be accessed.

(vi) The OS is responsible for the following activities in connection with file management.

- Creation and deletion of files
- Creation and deletion of directories
- The support of primitives for manipulating files and directories
- The mapping of files onto secondary storage
- The backup of file on stable storage media

1.7.4 I/O System Management

One of the purposes of an OS is to hide the pecularities of specific hardware devices from the usser.

The I/O subsystem consists of:

- A memory management component including buffering, caching and spooling
- A general device driver interface
- Drivers for specific hardware devices

1.7.5 Secondary Storage Management

(i) Main memory is too small to accommodate all data and programs and its data and program are lost when power is lost, hence, the computer system must provide secondary storage to back up main memory.

(ii) Most programs including compilers, assemblers, editors are stored on a disk until loaded into main memory and then use disk as both the source and destination of their processing.

The OS is responsible for the following activities in connection with disk management.

- Free space management
- Storage allocation
- Disk scheduling

1.7.6 Networking

(i) A distributed system is a collection of processors and does not share memory, peripheral devices or a clock.

(ii) Each processor has its local memory and clock and the processors communicate with one another through various communication lines like high speed buses or telephone lines.

(iii) The communication network design must consider routing and connection strategies and the problems of connection and security.

1.7.7 Protection System

(i) If a computer system has multiple users and allows the concurrent execution of multiple processors, then the various processes must be protected from each other's activity.

(ii) Protection refers to the mechanism for controlling the access of programs; processes, or users to the resources defined by a computer system.

1.8 OPERATING SYSTEM STRUCTURE

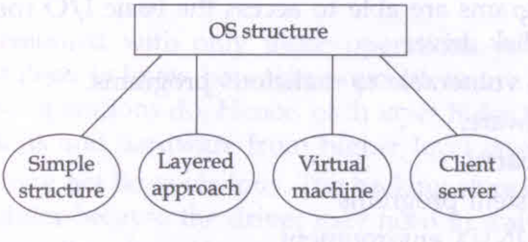

Fig. 1.9 OS structure

1.8.1 Simple Structure

(i) An OS is written as a collection of procedures, each of which can call any of the other one whenever it needs.

(ii) Basic structure of the OS can be divided into 3 parts:
- A main program that invokes the requested service procedure
- A set of service procedures that carry out system calls
- A set of utility procedures that help the service procedure

(iii) For each system call, there is one service procedure, that take care of it. The utility procedures do things that are really needed by several service procedures.

For example: MS DOS structure.

Structure of MS-DOS is shown in Fig. 1.10.

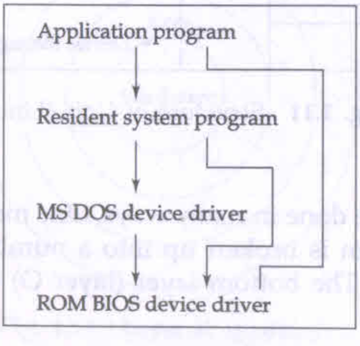

Fig. 1.10 Structure of MS-DOS

1.8.2 Command Interpreter System

(i) Command interpreter system is the interface between the user and the OS. Some OSs include command interpreter in the kernel.

(ii) Command line interpreter is often known as shell. OSs are frequently differentiated in the area of the shell.

(iii) One style of user friendly interface is the mouse based window and menu system is the Macintosh and Mirosoft Windwos.

The problems in this system architecture are:

1. Interfaces and levels of functionality are not well separated.
2. Application programs are able to access the basic I/O routines to write directly to display and disk drive.
3. Leaves MS DOS vulnerable to malicious programs.
4. Limited by hardware.
5. Has 2 separate parts:
   (a) Kernel (b) system programs

This is for UNIX/LINUX environment.

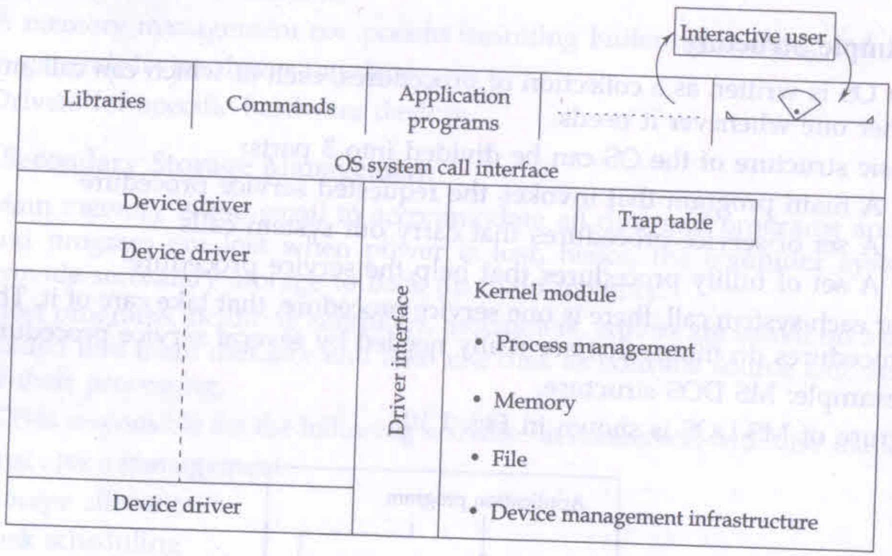

Fig. 1.11 Structure of Unix/Linux

1.8.3 Layered Approach

Modulation of a system can be done in many ways. One method is the layered approach, in which the operating system is broken up into a number of layers (or levels), each built on top of lower layers. The bottom layer (layer O) is the hardware, (layer N) is the user interface.

The main advantage of the layered approach is modularity. The layers are selected such that each uses functions (or operations) and services of only lower level layers. This approach simplifies debugging and system verification.

The first layer can be debugged without any concern for the rest of the system, because by definition, it uses only the basic hardware (which is assumed correct) to implement its functions. Once the first layer is debugged, its correct functioning can be assumed while the second layer is debugged and so on.

If an error is found during debugging of a particular layer, the error must be on that layer, because the layers below it are already debugged. Thus, the design and implementation of the system are simplified when the system is broken down into layers.

Each layer is implemented with only those operations provided by lower-level layers. A layer does not need to know how these operations are implemented, it needs to know only what these operations do. Hence, each layer hides the existence of certain data structures, operations and hardware from higher level layers.

Other requirements may not be so obvious. The backing-store driver would normally be above the CPU scheduler because the driver may need to wait for J/O and the CPU can be rescheduled during this time. However, on a large system, the CPU scheduler may have more information about all the active processes than can fit in memory. Therefore, this information may need to be swapped in and out of memory, requiring the backing store driver routine to be below the CPU scheduler.

Fewer layers with more functionality are designed, providing most of the advantages of modularized code while avoiding the problems of layer definition and interaction.

For example, consider the history of windows NT. The first release had a highly layer oriented organisation, however, it delivered low performance compared to that of Windows 95. Window NT 4.0 partially redressed the performance problem by moving layers from user space to kernel space and more closely integrating them.

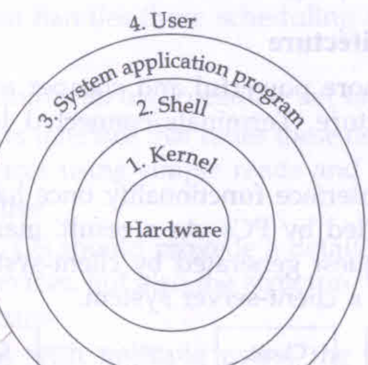

Fig. 1.12 Layered approach

1.8.4 Virtual Machine

Virtual Machine developed by IBM originally called CP/CMS and later renamed as VM/370 was based on two things: multiprogramming and an extended machine with a more convenient interface than the base hardware.

The heart of the system, known as virtual machine monitor, runs on the base hardware and does multiprogramming providing several virtual machines to the next layer.

Virtual machines are exact copies of base hardware including kernel user mode I/O interrupts. Different virtual machines run different operating systems on the single interactive system called conventional monitor system (CMS). When a CMS program executes a system call, the call is trapped to the operating in its own VM. CMS then issues the normal hardware I/O instruction for reading its virtual disk. These instructions are trapped by VM/370.

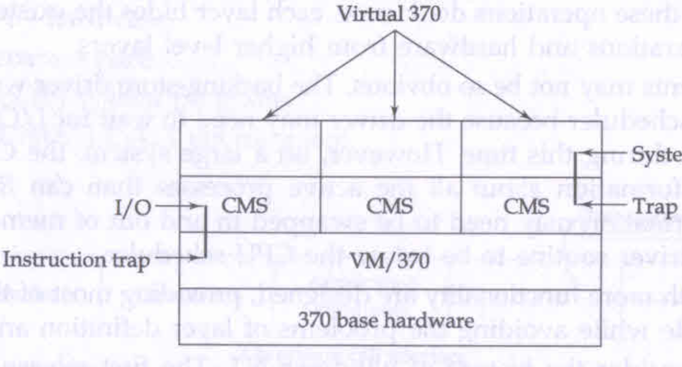

Fig. 1.13 Virtual M/C

Advantages

- Provides security and privacy
- Software development program.

1.8.5 Client-Server Architecture

PC's have become faster, more powerful and cheaper as designer have shifted away centralized system architecture. Terminals connected to centralized are now being supplanted by PCs.

Correspondingly, user interface functionality once handled directly by centralized system is increasingly handled by PCs. As a result, many of today's system act as a server system to satisfy request generated by client-system. This form of specialized distributed system is called a client-server system.

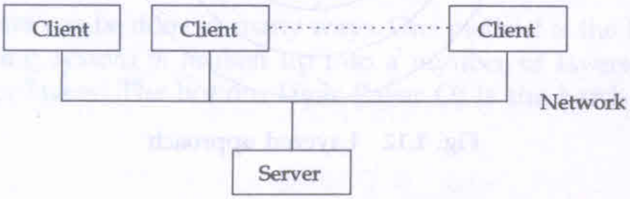

Fig. 1.14 Client server architecture

Server system can be broadly categorised as computer servers and file servers.

- The computer server system provides an interface to which a client can send a request to perform an action, in response the server executes the action/request and sends back the result to the client.
- The file server system provides a file-system interface, where clients can create, update, read and delete files.

Advantages

- Adaptable to distributed system.
- Minimal amount of mechanism into kernel.
- Still leave decision power on server.

Disadvantage

- Some operations are difficult in user space.

1.9 OPERATING SYSTEM SERVICES

Operating system services and facilities can be grouped into the following areas:

1. **Program development**
   (i) An operating system provides editors and debuggers to assist the programmer in creating programs.
   (ii) Usually, these services are in the form of utility programs and not strictly part of core sperating system. They are supplied with operating system and referred as application program development tools.
2. **Program execution**
   (i) A number of tasks need to be performed to execute a program. Instructions and data must be loaded into the main memory. I/O devices and files must be initialized.
   (ii) The operating system handles these scheduling duties for the user.
3. **Access to I/O devices**
   (i) Each I/O device requires its own peculiar set of instructions for operations.
   (ii) OS provides a uniform interface that hides these details, so that the programmer can access such devices using simple reads and writes.
4. **Controlled access to filter**
   (i) In case of file access, OS should provede a detailed understanding of not only the nature of I/O devices, but also the structure of data contained in the files on the storage medium.
   (ii) In case of a system with multiple users, the OS may provide protection mechanism to control access to files.
5. **System access**
   (i) In case of shared or public systems, the OS controls access to the system as a whole.

(ii) The access function must provide protection of resources and data from unauthorized users.

6. **Error detection and response**
   (i) Various types of errors can occur while a computer system is running which includes internal and external hardware errors.
   (ii) In each case, OS must provide a response that clears error condition with least impact on the running applications.

7. **Accounting**
   (i) A good OS collects a usage statistics for various resources and monitors performance parameters.
   (ii) On any system, this information is useful in anticipating need for future enhancements.

1.10 KERNELS

The kernel is the heart of the operating system. It interacts with hardware and most of the tasks like memory management, task scheduling, file management, I/O management etc. There are two types of kernels: monolithic kernel and microlithic kernel.

1.10.1 Monolothic Kernel

Monolithic kernel OS can be considered or a single large static binary file process running entirely in a single address space. Basic OS services such as process management, memory management, interrupt handling, I/O communication, file systems, device derivers, etc., all run in kernel space. Entire services are loaded on boot up and reside in memory and work is done using system calls. Linux is example of monolithic kernel based OS.

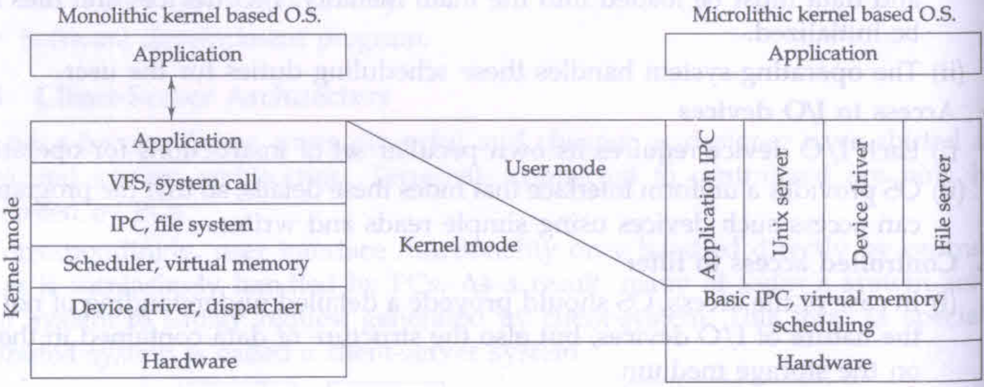

Fig. 1.15 Monolihic vs microlithic kernels

1.10.2 Microlithic Kernel

The idea behind micro kernel OS is to reduce the kernel to only basic process communication and I/O control and let other system services use the user space. These processers are called servers and kept separate in different address spaces. Contrary to monolithic OS where services are directly involved and invoked, communication in microkernel is done via message passing. Mac OS and Nin NT are examples.

Comparison between Monolithic and Microlithic Kernel

Table 1.1 Monolithic vs Microlithic

Monolithic	Microlithic
1. Kernel size is large	1. Kernel size is small
2. OS is complex to design	2. OS is easy to design, and install
3. Different modules can interact less with each other	3. Contains different modules on which work is performed
4. New modules can be dynamically linked, i.e., at run time	4. These can communicate with those modules by linking them
5. All OS services run along main kernel set in same memory architecture	5. They are used in embedded systems
6. Ex. UNIX, MS-DOS, SOLARIS	6. Ex. AIX, OS, Amega OS, Symbian OS

1.10.3 Re-entrant Kernel

A re-entrent kernel enables processes to give away the CPU while in kernel mode, not hindering other processes from also entering kernel mode.

A typical use case is IO wait. The process wants to read a file. It calls a kernel function for this. Inside the kernel function, the disk controller is asked for data. Getting the data will take some time and the function is blocked during that time.

With a re-entrant kernel, the scheduler will assign the CPU to another process unit an interrupt from the disk controller indicates that the data is available and our thread can be resumed.

This process can still access IO, like user i/P. The system stays responsive and CPU time waste due to IO wait is reduced. This is pretty much standard for today's desktop OS.

If the kernel is not re-entrant, a process can only be suspended while it is in user mode because all kernel threads share the same memory and corruption would occur if execution would jump between them arbitraily.

All UNIX kernels are re-entrants means several processes may be executing in kernel mode at the same time. A one way to provide re-entrance is to write functions so that they modify any local variables and do not alter global data structures. Such functions are called re-entrant functions.

If a hardware interrupt occurs, a re-entrant kernel is able to suspend the current running process even if that process is in kernel mode. This capability is very important because it improves the throughput of the devices controllers that issue interrupts

A kernel control path denotes the sequence of instructions executed by the kernel to handle a system call, an excepion or an interrupt.

Summary

- In this chapter we have studied about the operating system, its objectives, functions and the basic structure.
- The operating system acts as a virtual machine, a resource manager, a service provider and as a security agent.
- Similar jobs are batched together and executed in groups. It is called batch processing system. The main function of batch processing system is to automatically keep executing one job to the next job in the batch.
- Multiprogramming operating system is used to increase the system utilization efficiency.
- The operating system allowing execution of multiple tasks at one time, is classified as a multiprocessing operating system.
- Real-time operating system is used when there are rigid time requirements on the operations of a processor or the flow of data. Real time operating system has well defined, fixed time constraints.
- System calls provide an interface to the service mode available by an operating system. These calls are generally available as routines written in C and C++ or these calls are generally available as assembly language instruction.
- Operating system architecture based on layered approach consists of multiple layers each of which is built on top of lower layers.
- The monolithic approach defines a high level virtual interface over the hardware, with a set of primitives or system calls to implement operating system services such as process manager concurrency, memory management, etc.
- Microkernel provides basic functionality that allows the execution of servers.
- A re-entrant kernel enable process to give away the CPU while in kernel mode, not hindering other process from entering kernel mode. All UNIX kernels are re-entrants.

Multiple-Choice Questions

1. Which technique was introduced because a single job could not keep both the CPU and the I/O devices but
   (a) Time-sharing (b) Spooling
   (c) Preemptive scheduling (d) Multiprogramming
2. CPU performance is measured through
   (a) Throughput (b) MHz
   (c) Flaps (d) None of the above
3. PCB stands for
   (a) Program Control Block (b) Process Control Block
   (c) Process Communication Block (d) None of the above

4. Software is a program that directs the overall operation of the computer facilitates its use and interacts with the user. What are the different types of this software.
   (a) Operating system (b) Language compiler
   (c) Utilities (d) All of the above
5. Which of the following is contained in process control block (PCB)?
   (a) Process number (b) List of open files.
   (c) Memory limits (d) All of the above
6. Supercomputers typically employ
   (a) Real-time OS (b) Multiprocessor OS
   (c) Desktop OS (d) None of the above
7. Real-time systems are
   (a) Primarily used on mainframe computers
   (b) Used for monitoring events as they occur
   (c) Used for program development
   (d) Used for real time interactive users
8. In ____________ OS, the response time is very critical.
   (a) Multitasking (b) Batch
   (c) Online (d) Real-time
9. Which of the following is not an advantage of multiprograming?
   (a) Increased throughput
   (b) Shorter response time
   (c) Decreased operating system overhead
   (d) Ability to assign priorities to jobs.
10. The program is known as ____________ which interacts with the inner part of called kernel.
    (a) compiler (b) device driver
    (c) protocol (d) shell
11. Information about a process is maintained in a
    (a) Stack (b) Translation Lookaside Buffer
    (c) Process Control Block (d) Program Control Block
12. UNIX operating system is a
    (a) Time sharing operating system (b) Multi-user operating system
    (c) Multi-tasking operating system (d) All the above
13. Distributed OS works on the ____________ principle.
    (a) File foundation (b) Single system image
    (c) Multisystem image (d) Networking image

14. A __________ is a softwre that manages the time of a microprocessor to ensur that all time critical events are processed as efficiently as possible this softwar allows the system activities to be divided into multiple independent element called tasks.

(a) Kernel (b) Shell

(c) Processor (d) Device driver

15. The one program running at all times on the computer is called

(a) Kernel (b) Device Controller

(c) Shell (d) Multiprocessor

16. There are types of multiprocessing

(a) 3 (b) 4

(c) 2 (d) None of the above

Answers

1. (d)	2. (a)	3. (b)	4. (d)	5. (d)	6. (b)
7. (b)	8. (d)	9. (c)	10. (d)	11. (c)	12. (d)
13. (b)	14. (a)	15. (a)	16. (c)		