Chapter One
Operating System and System Software Overview

INTRODUCTION

Computer Software

A computer cannot do anything on its own. It must be instructed to do a job. So, it's necessary to specify a sequence of instructions that a computer must perform to solve a problem. Such a sequence of instructions written in a language understood by a computer is called a computer program. Computer software is the collection of computer programs and related data that provide instructions telling a computer what to do. Software is divided into 2types:

1. System Software
2. Application Software

1. System Software

System Software is a set of programs that manages and supports the operation of a computer. It controls the computer system and enhances its performance. It enables the application software to interact with the system hardware. That is system software act as a bridge between application software and computer hardware. Eg: Operating System, Compiler etc. System software can be broadly classified into 3 types:

- **i. System Control Programs**
- **ii. System Support Programs**
- **iii. System Development Programs**

System Control Programs: controls the execution of programs, manage the storage and processing resources of the computer and perform other management and monitoring functions. Eg: Operating System, Database Management Systems (DBMS).

System Support Programs: provide routine service functions to the other computer programs and computer users. Eg: utilities, libraries

System Development Programs: assists in the creation of application programs. Eg:
Language translators like compiler, assembler.

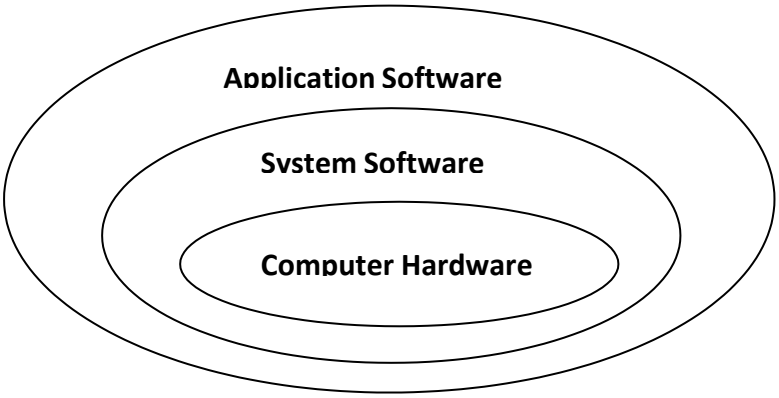
Application Software:

Application software consists of programs designed to perform specific tasks for users. .Eg: Media player, Ms. Word, Notepad etc. There are 2 types of application software:

General Purpose Application Software: These provide general user needs, not for a specific purpose. Eg: Spreadsheet program like Microsoft Excel, which can be used to perform different applications.

Special Purpose(Custom) Application Software: Typically used for specific applications. Eg: Turbo Tax, which is a special purpose application used to perform tax returns.

Relationship between system software and application software:



System software control and manages hardware thereby providing a platform for application software to operate. Application software helps user to accomplish one or more tasks using a computer through system software. For an application to run on a computer it needs to be allocate space from memory, allocated resources (CPU time, hardware like keyboard, display, printer etc), given access to system libraries, all of which is done by operating system which is a system software.

Differences between system software and application software (SystemSoftware Vs Application Software):

Category	Application Software	System Software
Definition	Application software is computer software designed to help the user to perform specific tasks	System software is computer software designed to operate the computer hardware and to provide a platform for running application software.
Purpose	It is specific purpose software.	It is general purpose software.
Environment	Application Software performs in an environment which created by Operating System	System Software create his own environment to run itself and run other application.
ExecutionTime	It executes as and when required	Some system software must executes all the time in computer.
Essentiality	Application software is not essentialfor a computer.	System software is essential for a computer
Number	The number of application software ismuch more than system software.	The number of system software is less than application software.
Machine dependent/ independent	Application software uses the computer for solving problems. That is, its focus ison application. So, its machine independent.	System software is intended to support the use and operation of acomputer. So, it's usually machine dependent.

DIFFERENT SYSTEM SOFTWARE

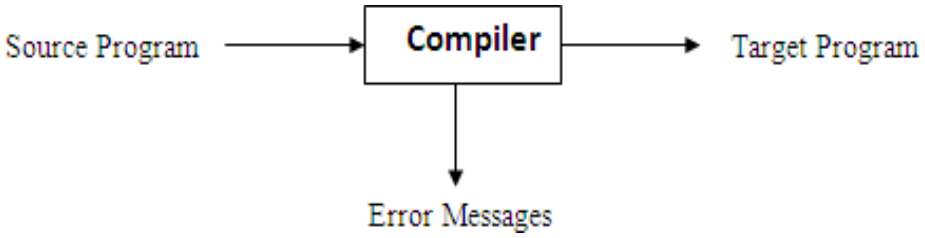
1. Language Translators (Compiler, Interpreter, Assembler, Macro Preprocessor)
2. Linker
3. Loader
4. Text Editor
5. Device Driver
6. Debugger
7. Operating Systems
8. Database Management Systems (DBMS)

1. Language Translators

It is the program that takes an input program in one language and produces an outputin another language.

a. Compiler

A compiler is a program that translates programs written in any high level language (source program) into its equivalent machine language program(target language). An important role of the compiler is to report any errors in the source program that it detects during the translation process.



b. Interpreter

An interpreter is also a language translator which converts source program in high level language into machine language, just like compiler did. It reads the source code oneline at a time, converts this line into machine code and executes it. The machine code is then discarded and next line is processed. It stops translating after the first error.

Compilers, on the other hand, translates the entire program in one go and then executes it. Compiler analyse the entire program, displays where errors have occurred. If errors are present, then program cannot run.

Interpreter works as follows. The interpreter reads the source program and stores it in memory. Program counter (PC) indicates which statement of the source program is to be interpreted next. During interpretation, it takes a source statement, determines its meaning and performs

the actions and PC is incremented. The interpretation cycle consists of the following steps:

- Fetch the statement.
- Analyze the statement and determine its meaning.
- Execute the meaning of the statement.

Advantages of interpreter:

Easier to use particularly for beginners, since errors are immediately displayed.

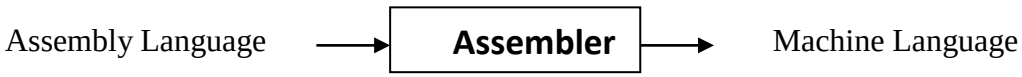
.**Disadvantage of interpreter:**

Every line has to be translated every time it is executed, even if it is executed many timesas the program runs. Because of this interpreters tend to be slow.

Example for interpreters: Basic on older home computers, script interpreters such as JavaScript.

c. Assembler

Assembler converts assembly language program into its equivalent machinelanguage.



Example: MASM (8086 assembler)

d. Macro Preprocessor

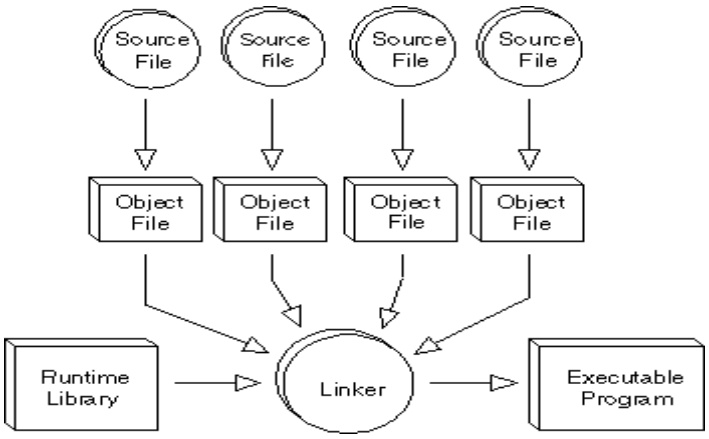
A macro processor is a program that reads a file and scans them for certain keywords. When a keyword is found, it is replaced by some text. The keyword/text combination is called a macro. A simple example is the C language pre-processor:

```
#define max 100; int i; for(i=0;i<max;i++)
{
.........
}
```

The C preprocessor reads the first line and stores it as macro definition. When it comes across the later reference of max in the for loop, it replaces with the macro definition 100. The output of the C processor is then fed to the C compiler.

2. Linker

A linker is a program that combines object modules to form an executable program. Many programming languages allow us to write different pieces of code, called *modules*, separately. This simplifies the programming task because we can break a large program into small, more manageable pieces. Eventually, though, we need to put all the modules together. This is the job of the linker. In addition to combining modules, a linker also replacessymbolic addresses with real addresses. Therefore, we may need to link a program even if it contains only one module.



3. Loader

A **loader** is the part of an operating system that is responsible for loading programs. Itis one of the essential stages in the process of starting a program, as it places programs into memory and prepares them for execution. Loading a program involves reading the contents of the executable file containing the program instructions into memory, and then carrying out other required preparatory tasks to prepare the executable for running. Once loading is complete, the operating system starts the program by passing control to the loaded program code.

4. Text Editors

A **text editor** is a type of program used for editing plain text files. Text editors are often provided with operating systems or software development packages. Example for text editors are Microsoft word, gedit in Linux etc.

5. Debuggers

A **debugger** or **debugging tool** is a computer program that is used to test and debug other programs.

6. Device Drivers

In computing, a **device driver** is a computer program that operates or controls a particular type of device that is attached to a computer. A Device Driver is glue between an OS and its I/O devices. They act as translators converting requests received from the operating system into commands that the devices can understand.

7. Operating System

It is the most important system program that act as an interface between the users and the system. It makes the computer easier to use. It provides an interface that is moreuser-friendly than the underlying hardware.

The functions of OS are:

1. Process management
2. Memory management
3. Resource management
4. I/O operations
5. Data management
6. Providing security to user's job.

8. Database Management System

A **database** is a collection of related data. Data can be consider as a piece of information which can be recorded.

Data Base Management System (DBMS) is a collection of programs that enablesusers to create and maintain a database. It is a general purpose software that facilitates theprocesses of defining, constructing, manipulating and sharing of database among various users and applications.

Defining a database means specifying the data types, structures and constraints for the data to be stored in the database.

Constructing the database is the process of storing the data on somestorage medium that is controlled by the DBMS.

Manipulating the database means processing the database. It includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the mini world and generating reports from the data.

Sharing a database allows multiple users and programs to access the database concurrently.

A **Database System** is a computerized record keeping system whose overall purpose is to store information and to allow users to retrieve and update that information on demand. Database system consists of a Database, Database Management System and an Application program. An application program accesses the database by sending requests or queries for data to the DBMS.

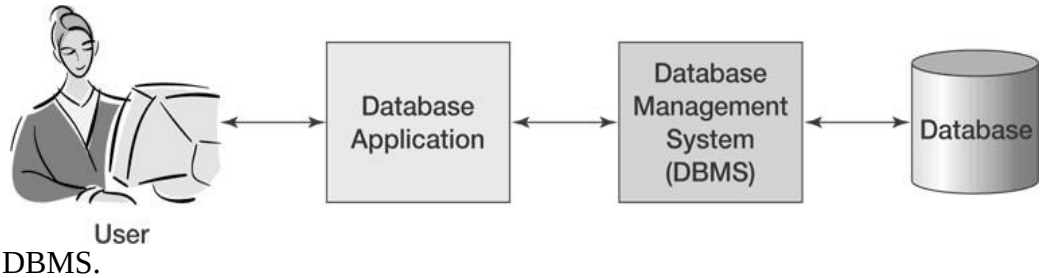


Fig: A Database System

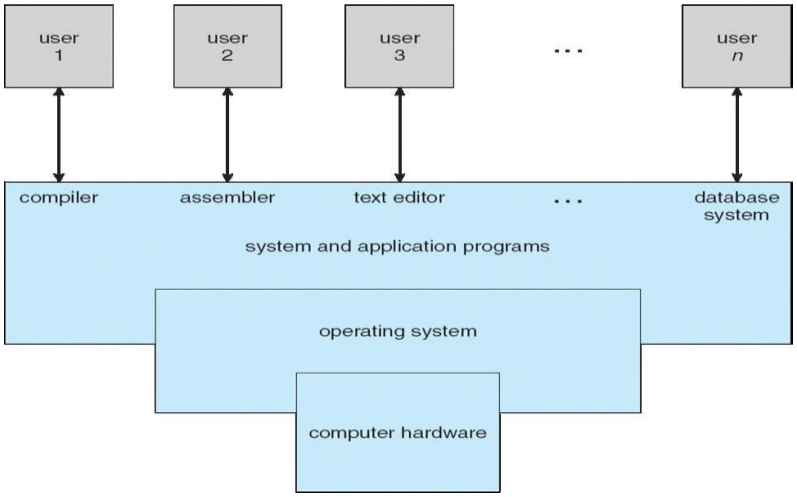
What is an Operating System?

- A program that acts as an intermediary between a user of a computer and the computer hardware
- 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

Components of a Computer System

An operating system is an important part of almost every computer system. A computer system can be dividedroughly into four components.

I. Hardware
II. Operating system
III. The application programs
IV. Users



Hardware – provides basic computing resources

CPU, memory, I/O devices

Operating system- Controls and coordinates use of hardware among various applications and users

Application programs – define the ways in which the system resources are used to solve thecomputing problems of the users

Word processors, compilers, web browsers, database systems, video games

Users - People, machines, other computers

Computer system architecture

Computer architecture means design or construction of a computer. A computer system may be organizedin different ways. Some computer systems have single processor and other have multiprocessors. So, computer systems categorized in these ways.

1. Single Processor Systems
2. Multiprocessor Systems
3. Clustered Systems

Single Processor

Some computers use only one processor such as microcomputers.

On a single processor system, there is only one CPU that perform all the activities in the computer system, However,most of these systems have other special purpose processors, such as I/O Processor that move data rapidly among different components of the computer system.

Multi-Processor systems

Some systems have two or more processors. These systems are also known as parallel systems or tightly coupledsystems.

Mostly the processors of these systems share the common system bus (electronic path) memory and peripheral(input/output) devices. These systems are fast in data processing and have capability to execute more than oneprogram simultaneously on different processors. This type of processing is known as multiprogramming or multiprocessing.

Advantages:

I. Increased Throughput (means number of jobs completed by a CPU within a time period):By increasing the number of processors, we expect to get more work done in less time

II. Economical: Multiprocessor systems can cost less than equivalent multiple single-processor systems,because they can share peripherals, mass storage, and power supplies.

III. Increased Reliability: If functions can be distributed properly among several processors, then the failureof one processor will not halt the system, only slow it down.

Clustered systems are another form of multiprocessor system. This system also contains multiple processors but itdiffers from multiprocessor system.

The clustered system is composed of **multiple individual systems that connected together**. In clustered system,also individual systems or computers share the same storage and liked to gather via local area network.

A special type of software is known as cluster to control the node the systems.

What Operating Systems Do??

- **Role of OS depends on the point of view**

User's view

- In the case of Personal Computer, the operating system is designed mostly for ease of use, with some attention paid to performance and none paid to resource utilization—how various hardware and software resources are shared.
- In other cases, a user sits at a terminal connected to a mainframe or a minicomputer. Other users are accessing the same computer through other terminals. These users share resources and may exchange information. The operating system in such cases is designed to maximize resource utilization — to assure that all available CPU time, memory, and I/O are used efficiently
- In the case of hand held devices such as lap top or smart phones, operating system is designed mostly for ease of use and for resource utilization (battery life).

System View

- From system's view OS is the program most intimately involved with the hardware. So we can

view OSas a resource allocator
- OS must decide and users so that it can operate the computer system efficient and fairly
- OS act as a control program that manages the execution of user programs and control various I/O devices

➢ OS is a **resource allocator**
- Manages all resources
- Decides between conflicting requests for efficient and fair resource use

➢ OS is a **control program**
- Controls execution of programs to prevent errors and improper use of the computer

Functions of Operating Systems

Process Management

A process is a program in execution. A process needs certain resources, including CPU time, memory,files, and I/O devices, to accomplish its task.

The operating system is responsible for the following activities in connection with process management.

➢ Process creation and deletion.

➢ Process suspension and resumption.Provision of mechanisms for:

Process synchronizationProcess communication

Main-Memory Management

Memory is a large array of words or bytes, each with its own address. It is a repository of quickly accessible datashared by the CPU and I/O devices.

Main memory is a volatile storage device. It loses its contents in the case of system failure.

The operating system is responsible for the following activities in connections with memory management:

➢ Keep track of which parts of memory are currently being used and by whom.

➢ Decide which processes to load when memory space becomes available.

➢ Allocate and deallocate memory space as needed.

File Management

A file is a collection of related information defined by its creator. Commonly, files represent programs (both sourceand object forms) and data.

The operating system is responsible for the following activities in connections with file management:

➢ File creation and deletion.

➢ Directory creation and deletion.

➢ Support of primitives for manipulating files and directories.

➢ Mapping files onto secondary storage.

➢ File backup on stable (nonvolatile) storage media

I/O System Management

The I/O system consists of:

✦A buffer-caching system

- ✦A general device-driver interface
- ✦Drivers for specific hardware devices

Secondary-Storage Management

Since main memory (primary storage) is volatile and too small to accommodate all data and programs permanently,the computer system must provide secondary storage to back up main memory.

Most modern computer systems use disks as the principle on-line storage medium, for both programs anddata.

The operating system is responsible for the following activities in connection with disk management:

- ✦Free space management
- ✦Storage allocation
- ✦Disk scheduling

Networking (Distributed Systems)

A distributed system is a collection processor that do not share memory or a clock. Each processor has its ownlocal memory.

The processors in the system are connected through a communication network.Communication takes place using a protocol.

A distributed system provides user access to various system resources.

Access to a shared resource allows:

- ✦Computation speed-up
- ✦Increased data availability
- ✦Enhanced reliability

Protection System

Protection refers to a mechanism for controlling access by programs, processes, or users to both system anduser resources.

The protection mechanism must:

- ✦distinguish between authorized and unauthorized usage.
- ✦specify the controls to be imposed.
- ✦provide a means of enforcement.

Command-Interpreter System

Many commands are given to the operating system by control statements which deal with:

- ✦process creation and management
- ✦I/O handling
- ✦secondary-storage management
- ✦main-memory management
- ✦file-system access
- ✦protection

✦networking

Operating System Interfaces and implementation

User Interface

`Almost all operating systems have a User Interface(UI). This interface can take several forms. One is aCommand Line Interface (CLI) or command Interpreter and another is a Graphical User Interface (GUI).

1. A command-line interface, or **command interpreter**, that allows users to directly enter commands to beperformed by the operating system.
2. a graphical user interface, or GUI, that allows users to interface with the operating system via i/o devicesand icons

Command Interpreters

Some operating systems include the command interpreter in the kernel. These interpreters are known as **shells**.Eg: ***Bourne shell***, ***C shell***, ***Bourne-Again shell***, ***Korn shell***

The main function of the command interpreter is to get and execute the next user-specified command. Many of the commands given at this level manipulate files: create, delete, list, print, copy, execute, and so on. Thesecommands can be implemented in two general ways.

1. The command interpreter itself contains the code to execute the command. For example, a command todelete a file may cause the command interpreter to jump to a section of its code that sets up the parameters and makes the appropriate system call.
2. Implements most commands through system programs. In this case, the command interpreter does not understand the command in any way; it merely uses the command to identify a file to be loaded into memory and executed.

Graphical User Interface

A second strategy for interfacing with the operating system is through a user-friendly graphical user interface, orGUI.

Users employ a mouse-based window and-menu system characterized by a **desktop** metaphor. The user moves the mouse to position its pointer on images, or **icons**, on the screen (the desktop) that represent programs, files, directories, and system functions. Depending on the mouse pointer's location, clicking a button on the mouse can invoke a program, select a file or directory—known as a **folder**—or pull down a menu that contains commands.

Because a mouse is impractical for most mobile systems, smart phones and handheld tablet computers typically use a touch screen interface. Here, users interact by making **gestures** on the touch screen—for example, pressing and swiping fingers across the screen

System calls

System call is the programmatic way in which a computer program requests a service from the kernel of theoperating system it is executed on.

-Provide an interface to the services that are available to O.S

- Written in high level language (C, C++)
- To understand system calls, first one needs to understand the difference between kernel mode and user mode of aCPU.

Program execution done in two modes

3. User mode
4. Kernel mode

User mode- does not have direct access to memory, hardware and resources. If a program crashes, it does notaffect the system.

Kernel mode- direct access to memory, hardware and resources. In this mode, if a program crashes entire system crashes.

When a program in user mode requires access to RAM or a hardware resource, it must ask the kernel toprovide access to that resource. This is done via something called a system call.

When a program makes a system call, the mode is switched from user mode to kernel mode. This is calleda **context switch**.

Generally, system calls are made by the user level programs in the following situations:

A. Creating, opening, closing and deleting files in the file system.

B. Creating and managing new processes.

C. Creating a connection in the network, sending and receiving packets.

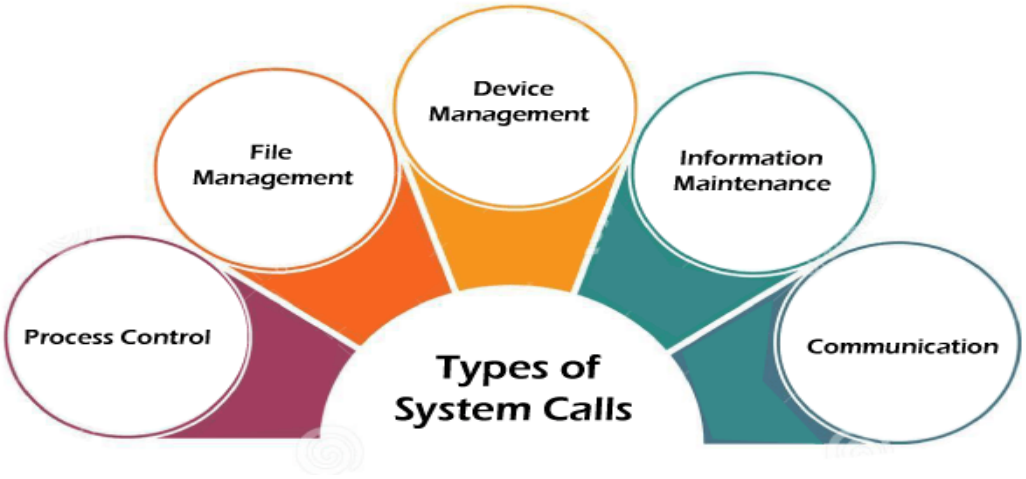
D. Requesting access to a hardware device, like a mouse or a printer.

E. system supports these two modes.

System calls are usually made when a process in user mode requires access to a resource. Then it requests thekernel to provide the resource via a system call.

Types of System Calls

There are commonly five types of system calls. These are as follows:



Examples of Windows and UNIX system calls

There are various examples of Windows and UNIX system calls. These are as listed below in the table:

Process	Windows	Unix
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	Fork() Exit() Wait()
File Manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	Open() Read() Write() Close()
Device Management	SetConsoleMode() ReadConsole() WriteConsole()	Ioctl() Read() Write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	Getpid() Alarm() Sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	Pipe() Shmget() Mmap()
Protection	SetFileSecurity() InitializeSecurityDescriptor() SetSecurityDescriptorgroup()	Chmod() Umask() Chown()

Types of the operating system

- **Batch Operating System:** This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having the same requirements and groups them into batches.
- **Time-sharing operating System:** This type of operating system allows many users to share computer resources. (Max utilization of the resources).

- **Distributed operating System:** This type of operating system manages a group of different computers and makes appear to be a single computer. These operating systems are designed to operate on a network of computers. They allow multiple users to access shared resources and communicate with each other over the network. Examples include Microsoft Windows Server and various distributions of Linux designed for servers.
- **Network operating system:** This type of operating system running on a server and provides the capability to manage data, users, groups, security, applications, and other networking functions.
- **Real-time operating system:** This type of operating system serves real time system and the time interval required to process and respond to inputs is very small. These operating systems are designed to respond to events in real-time. They are used in applications that require quick and deterministic responses, such as embedded systems, industrial control systems, and robotics.
- **Multiprocessing operating system:** Multiprocessor operating systems are used in operating systems to boost the performance of multiple CPUs within a single computer system. Multiple CPUs are linked together so that a job can be divided and executed more quickly.
- **Single-User Operating Systems:** These operating systems are designed to support a single user at a time. Examples include Microsoft Windows for personal computers and Apple macOS.
- **Multi-User Operating Systems:** These operating systems are designed to support multiple users simultaneously. Examples include Linux and Unix.
- **Embedded Operating Systems:** These operating systems are designed to run on devices with limited resources, such as smartphones, wearable devices, and household appliances. Examples include Google's Android and Apple's iOS.
- **Cluster Operating Systems:** These operating systems are designed to run on a group of computers, or a cluster, to work together as a single system. They are used for high-performance computing and for applications that require high availability and reliability. Examples include Rocks Cluster Distribution and OpenMPI.