Program : **B.Tech**

Subject Name: **Operating System**

Subject Code: **CS-405**

Semester: **4th**

UNIT-V

Distributed Operating System

These types of operating system is a recent advancement in the world of computer technology and are being widely accepted all-over the world and, that too, with a great pace. Various autonomous interconnected computers communicate each other using a shared communication network. Independent systems possess their own memory unit and CPU. These are referred as **loosely coupled systems** or distributed systems. These systems processors differ in sizes and functions. The major benefit of working with these types of operating system is that it is always possible that one user can access the files or software which are not actually present on his system but on some other system connected within this network i.e., remote access is enabled within the devices connected in that network.

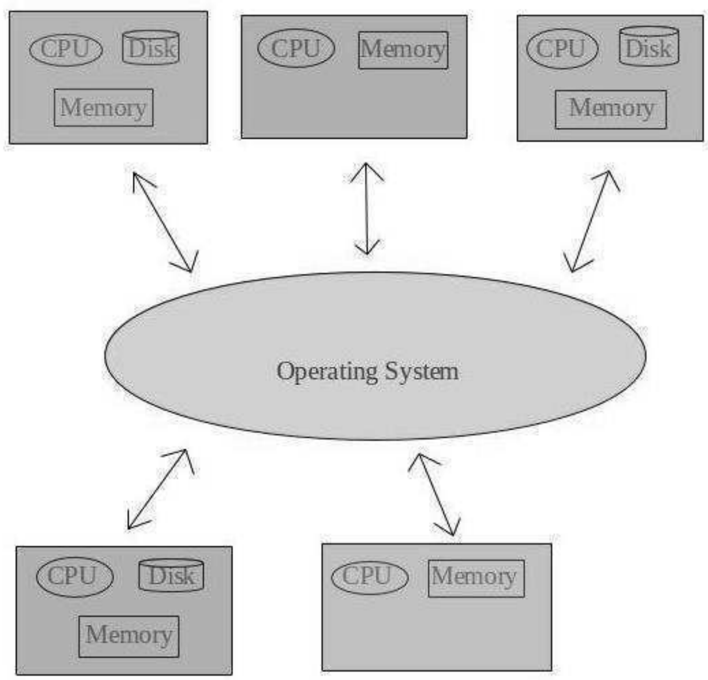


Fig. 5.1 Distributed Operating System

Advantages of Distributed Operating System:

- Failure of one will not affect the other network communication, as all systems are independent from each other
- Electronic mail increases the data exchange speed
- Since resources are being shared, computation is highly fast and durable
- Load on host computer reduces
- These systems are easily scalable as many systems can be easily added to the network
- Delay in data processing reduces

Disadvantages of Distributed Operating System:

- Failure of the main network will stop the entire communication
- To establish distributed systems the language which are used are not well defined yet
- These types of systems are not readily available as they are very expensive. Not only that the underlying software is highly complex and not understood well yet

Examples of Distributed Operating System are- LOCUS etc.

Network Operating System

These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions. These type of operating systems allows shared access of files, printers, security, applications, and other networking functions over a small private network. One more important aspect of Network Operating Systems is that all the users are well aware of the underlying configuration, of all other users within the network, their individual connections etc. and that's why these computers are popularly known as **tightly coupled systems**.

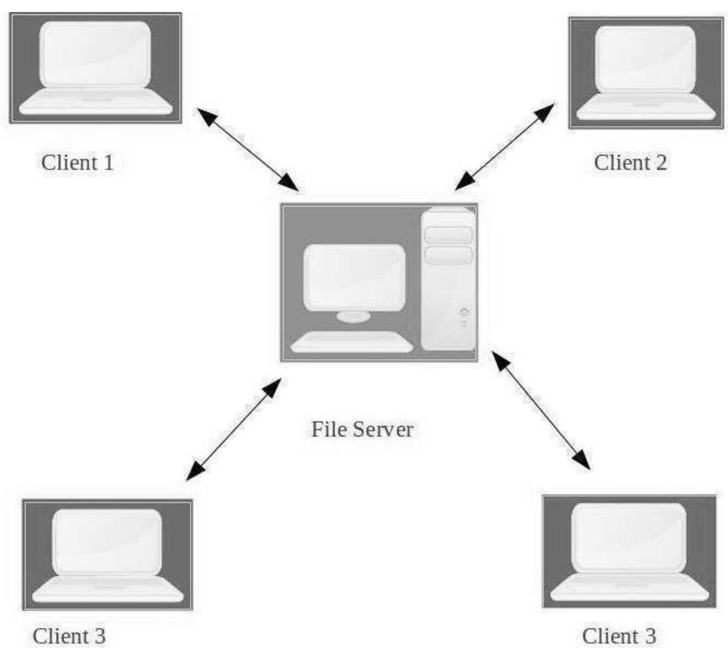


Fig. 5.2 Network Operating System

Advantages of Network Operating System:

- Highly stable centralized servers
- Security concerns are handled through servers
- New technologies and hardware up-gradation are easily integrated to the system
- Server access are possible remotely from different locations and types of systems

Disadvantages of Network Operating System:

- Servers are costly
- User has to depend on central location for most operations
- Maintenance and updates are required regularly

Examples of Network Operating System are: Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD etc.

Differences between Network Operating System and Distributed Operating System

Sr. No.	Network Operating System	Distributed Operating System
1	A network operating system is made up of software and associated protocols that allow a set of computer network to be used together.	A distributed operating system is an ordinary centralized operating system but runs on multiple independent CPUs.
2	Environment users are aware of multiplicity of machines.	Environment users are not aware of multiplicity of machines.
3	Control over file placement is done manually by the user.	It can be done automatically by the system itself.
4	Performance is badly affected if certain part of the hardware starts malfunctioning.	It is more reliable or fault tolerant i.e distributed operating system performs even if certain part of the hardware starts malfunctioning.
5	Remote resources are accessed by either logging into the desired remote machine or transferring data from the remote machine to user's own machines.	Users access remote resources in the same manner as they access local resources.

Multiprocessor Operating System

It refers to the use of two or more central processing units (CPU) within a single computer system. These multiple CPUs are in a close communication sharing the computer bus, memory and other peripheral devices. These systems are referred as *tightly* coupled systems.

These types of systems are used when very high speed is required to process a large volume of data. These systems are generally used in environment like satellite control, weather forecasting etc. The basic organization of multiprocessing system is shown in fig.

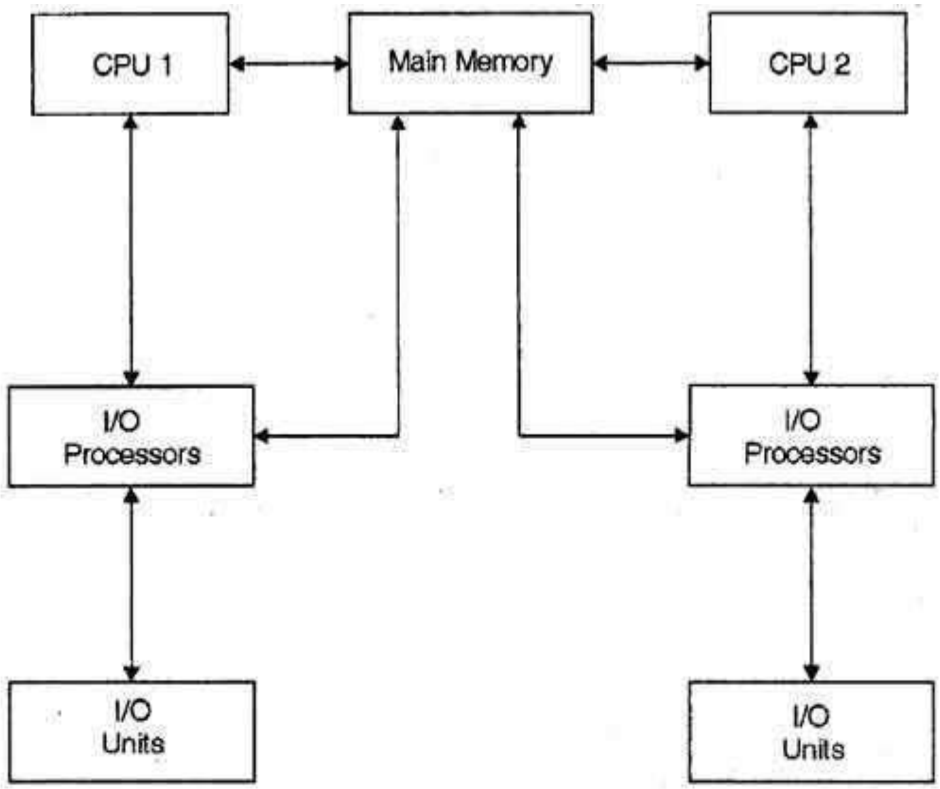


Fig. 5.3 Multiprocessor Operating System

Multiprocessing system is based on the symmetric multiprocessing model, in which each processor runs an identical copy of operating system and these copies communicate with each other. In this system processor is assigned a specific task. A master processor controls the system. This scheme defines a master-slave relationship. These systems can save money in compare to single processor systems because the processors can share peripherals, power supplies and other devices. The main advantage of multiprocessor system is to get more work done in a shorter period of time. Moreover, multiprocessor systems prove more reliable in the situations of failure of one processor. In this situation, the system with multiprocessor will not halt the system; it will only slow it down.

In order to employ multiprocessing operating system effectively, the computer system must have the followings:

1. Motherboard Support:

A motherboard capable of handling multiple processors. This means additional sockets or slots for the extra chips and a chipset capable of handling the multiprocessing arrangement.

2. Processor Support:

Processors those are capable of being used in a multiprocessing system.

The whole task of multiprocessing is managed by the operating system, which allocates different tasks to be performed by the various processors in the system.

Applications designed for the use in multiprocessing are said to be threaded, which means that they are broken into smaller routines that can be run independently. This allows the operating system to let these threads run on more than one processor simultaneously, which is multiprocessing that results in improved performance.

Multiprocessor system supports the processes to run in parallel. Parallel processing is the ability of the CPU to simultaneously process incoming jobs. This becomes most important in computer system, as the CPU divides and conquers the jobs. Generally the parallel processing is used in the fields like artificial intelligence and expert system, image processing, weather forecasting etc.

Locking system: In order to provide safe access to the resources shared among multiple processors, they need to be protected by locking scheme. The purpose of a locking is to serialize accesses to the protected resource by multiple processors. Undisciplined use of locking can severely degrade the performance of system. This form of contention can be reduced by using locking scheme, avoiding long critical sections, replacing locks with lock-free algorithms, or, whenever possible, avoiding sharing altogether.

Shared data: The continuous accesses to the shared data items by multiple processors (with one or more of them with data write) are serialized by the cache coherence protocol. Even in a moderate-scale system, serialization delays can have significant impact on the system performance. In addition, bursts of cache coherence traffic saturate the memory bus or the interconnection network, which also slows down the entire system. This form of contention can be eliminated by either avoiding sharing or, when this is not possible, by using replication techniques to reduce the rate of write accesses to the shared data.

False sharing: This form of contention arises when unrelated data items used by different processors are located next to each other in the memory and, therefore, share a single cache line: The effect of false sharing is the same as that of regular sharing bouncing of the cache line among several processors. Fortunately, once it is identified, false sharing can be easily eliminated by setting the memory layout of non-shared data.

Case Study: Linux:

Linux is one of popular version of UNIX operating System. It is open source as its source code is freely available. It is free to use. Linux was designed considering UNIX compatibility. Its functionality list is quite similar to that of UNIX.

Components of Linux System

Linux Operating System has primarily three components

- **Kernel** – Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.
- **System Library** – System libraries are special functions or programs using which application programs or system utilities accesses Kernel's features. These libraries

implement most of the functionalities of the operating system and do not requires kernel module's code access rights.

- **System Utility** – System Utility programs are responsible to do specialized, individual level tasks.

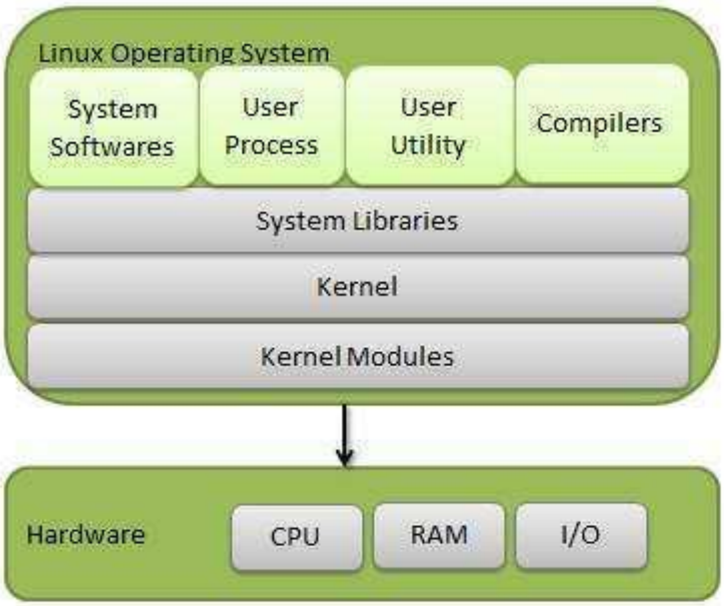


Fig. 5.4 Linux Operating System

Kernel Mode vs User Mode

Kernel component code executes in a special privileged mode called **kernel mode** with full access to all resources of the computer. This code represents a single process, executes in single address space and do not require any context switch and hence is very efficient and fast. Kernel runs each process and provides system services to processes, provides protected access to hardware to processes.

Support code which is not required to run in kernel mode is in System Library. User programs and other system programs works in **User Mode** which has no access to system hardware and kernel code. User programs/ utilities use System libraries to access Kernel functions to get system's low level tasks.

Basic Features

Following are some of the important features of Linux Operating System.

- **Portable** – Portability means software can works on different types of hardware in same way. Linux kernel and application programs support their installation on any kind of hardware platform.
- **Open Source** – Linux source code is freely available and it is community based development project. Multiple teams work in collaboration to enhance the capability of Linux operating system and it is continuously evolving.
- **Multi-User** – Linux is a multiuser system means multiple users can access system resources like memory/ ram/ application programs at same time.

- **Multiprogramming** – Linux is a multiprogramming system means multiple applications can run at same time.
- **Hierarchical File System** – Linux provides a standard file structure in which system files/ user files are arranged.
- **Shell** – Linux provides a special interpreter program which can be used to execute commands of the operating system. It can be used to do various types of operations, call application programs. etc.
- **Security** – Linux provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data.

Architecture

The following illustration shows the architecture of a Linux system –

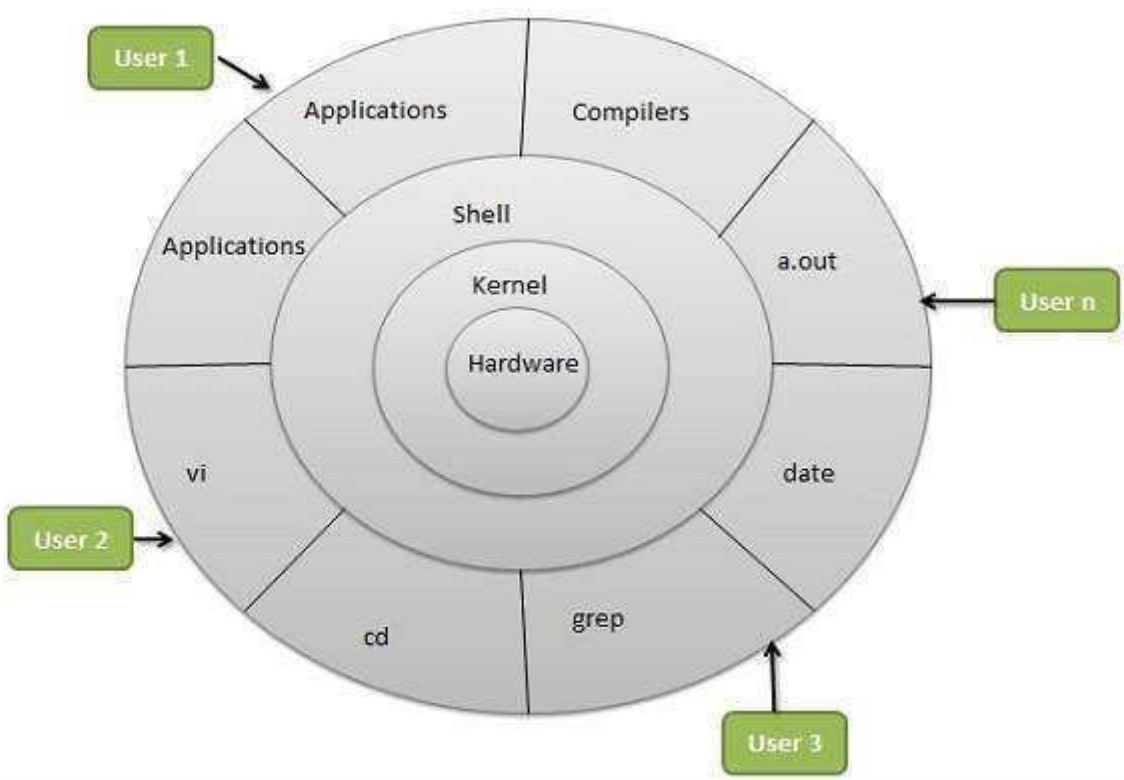


Fig. 5.5 Linux Operating System Architecture

The architecture of a Linux System consists of the following layers –

- **Hardware layer** – Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).
- **Kernel** – It is the core component of Operating System, interacts directly with hardware, provides low level services to upper layer components.
- **Shell** – An interface to kernel, hiding complexity of kernel's functions from users. The shell takes commands from the user and executes kernel's functions.
- **Utilities** – Utility programs that provide the user most of the functionalities of an operating systems.

The Shell

- The shell is the interface between the command language user and the OS
- The shell is a user interface and comes in many forms (Bourne Shell, sh; Berkeley C Shell, csh; Korn Shell, ksh; Restricted Shell, rsh)
- User allowed to enter input when prompted ($ or %)
- System supports all shells running concurrently. Appropriate shell is loaded at login, but user can usually (except in sh, rsh) dynamically change the shell
- A UNIX command takes the form of

  executable_file [-options] arguments

- The shell runs a **command interpretation loop**
  - accept command
  - read command
  - process command
  - execute command
- Executing the command involves creating a child process running in another shell (an environment within which the process can run). This is done by Forking.
- The parent process usually waits for the child to terminate before re-entering the command interpretation loop
- Programs can be run in the background by suffixing the command-line entry with an ampersand (&). Parent will not wait for child to terminate

The Processing Environment

Input and Output

- UNIX automatically opens three files for the process

  STDIN - standard input (attached to keyboard)

  STDOUT - standard output (attached to terminal)

  STDERR - standard error (attached to terminal)

- Because UNIX treats I/O devices as special types of files, STDIO can be easily redirected to other devices and files

  who > list _of _users

The Kernel

- Central part of the OS which provides system services to application programs and the shell
- The kernel manages processes, memory, I/O and the Timer - so this is not the same as the kernel that we covered in Lecture 3!
- UNIX supports multiprogramming
- Processes have their own address space - for protection
- Each process's process environment is composed of an unmodifiable re-entrant text (code) region, a modifiable data region and a stack region.
- The text region is shareable
- Processes can modify their environment only through calls to the OS

The File System

- UNIX uses HDS with root as its origin
- A directory is a special UNIX file which contains file names and their i-nodes (index nodes)
- Subdirectories appear as file entries
- Directories cannot be modified directly, but can are changed by the operating system when files and subdirectories are created and deleted
- File and Directory names must be unique within a particular directory (i.e., the path name must be unique)
- The File System is a data structure that is resident on disk
- It contains a super block (definition of the file system); an array of i-nodes (definition of the files in the system); the actual file data blocks; and a collection of free blocks
- Space allocation is performed in fixed-size blocks

The i-node contains:

the file owner's user-id and group-id

protection bits for owner, group, and world

the block locator

file size

accounting information

number of links to the file

file type

The Block Locator

- Consists of 13 fields
- First 10 fields points directly to first 10 file blocks
- 11th field is an indirect block address
- 12th field is a double-indirect block address
- 13th field is a triple-indirect block address

Permissions

- Each UNIX file and directory has 3 sets of permission bits associated with it
- These give permissions for owner, group and world
- System files (inc. devices) are owned by root, wizard, or superuser (terminology!)
- Root has unlimited access to the entire installation - whoever owns the files!

Setuid

- When you need to change your password, you need to modify a file called /etc/passwd. But this file is owned by root and nobody other than root has write permission!
- The **passwd** command (to change passwords) is owned by root, with execute permission for world.
- The setuid is a bit which when set on an executable file temporarily gives the user the same privileges as the owner of the file
- This is similar in concept to some OS commands executing in Supervisor mode to perform a service for an otherwise unauthorised process

Process Management

- Description of Process Management in SunOS

Scheduling

- Priority-based pre-emptive scheduling. Priorities in range -20 to 20. Default 0.
- Priorities for runnable processes are recomputed every second
- Allows for ageing, but also increases or decreases process's priority based on past behaviour
- I/O-bound processes receive better service
- CPU-bound processes do not suffer indefinite postponement because the algorithm `forgets' 90% CPU usage in 5*n seconds (where n is the average number of runnable processes in the past 60 seconds)

Signals

- Signals are software equivalents to hardware interrupts used to inform processes asynchronously of the occurrence of an event

Interprocess Communication

- UNIX System V uses semaphores to control access to shared resources
- For processes to exchange data or communicate, pipes are used
- A pipe is a unidirectional channel between 2 processes
- UNIX automatically provides buffering, scheduling services and synchronisation to processes in a pipe line
- The presence of a pipe causes the processes in the pipe line to share STDIO devices

  who | grep cstaff

- The output from **who** is directed to a buffer. **grep** will take its input from this buffer. The output from **grep** will be displayed on the terminal

Timers

- UNIX makes 3 interval timers available to each process
- Each counts down to zero and then generates a signal
- The first runs continuously
- The second runs while a process is executing process code
- The third runs while the process executes process code or kernel code

Memory Management

Address Mapping (Virtual Storage) - Paged MMS

- Virtual address V is dynamically translated to real address (P, D)
- Direct Mapping is used, with the Page Map held in a high-speed RAM cache
- Each Page Map Entry contains a modified bit, an accessed bit, a valid bit (if the page is resident in PM) and protection bits
- The system maintains 8 page maps - 1 for the kernel (not available to processes) and 7 for processes (contexts)
- 2 context registers are used - one points to the running process's page map and the other to the kernel's page map
- The replacement strategy replaces the page that has not been active for longest (LRU)

Paging

- SunOS maintains 2 data structures to control paging
- The free list contains empty page frames
- The loop contains an ordered list of all allocated page frames (except for the kernel)
- The pager ensures that there is always free space in memory
- When a page is swapped out (not necessarily replaced) the system judges whether the page is likely to be used again

- If the page contains a text region, the page is added to the bottom of the free list, otherwise it is added to the top
- When a page fault occurs, if the page is still in the free list it is reclaimed

I/O

Data

- All data is treated as a byte stream
- UNIX does not impose any structure on data - the applications do
- So data can be manipulated in any way - but programmers must explicitly structure the data

Devices

- A device is just a special type of file
- These files can have protection bits, so that a printer, e.g., cannot be read
- Permission to use sensitive devices, e.g., magnetic disk, is restricted to root and all other users have to use system calls to executable files which have their setuid bit set

Case Study: Windows

- Windows OS, computer operating system (OS) developed by Microsoft Corporation to run personal computers (PCs). Featuring the first graphical user interface (GUI) for IBM-compatible PCs, the Windows OS soon dominated the PC market. Approximately 90 percent of PCs run some version of Windows.
- The first version of Windows, released in 1985, was simply a GUI offered as an extension of Microsoft's existing disk operating system, or MS-DOS. Based in part on licensed concepts that Apple Inc. had used for its Macintosh System Software, Windows for the first time allowed DOS users to visually navigate a virtual desktop, opening graphical "windows" displaying the contents of electronic folders and files with the click of a mouse button, rather than typing commands and directory paths at a text prompt.
- Subsequent versions introduced greater functionality, including native Windows File Manager, Program Manager, and Print Manager programs, and a more dynamic interface. Microsoft also developed specialized Windows packages, including the networkable Windows for Workgroups and the high-powered Windows NT, aimed at businesses. The 1995 consumer release Windows 95 fully integrated Windows and DOS and offered built-in Internet support, including the World Wide Web browser Internet Explorer.
- With the 2001 release of Windows XP, Microsoft united its various Windows packages under a single banner, offering multiple editions for consumers, businesses, multimedia developers, and others. Windows XP abandoned the long-used Windows 95 kernel (core software code) for a more powerful code base and offered a more practical interface and improved application and memory management. The highly successful XP standard was succeeded in late 2006 by Windows Vista, which experienced a troubled rollout and

met with considerable marketplace resistance, quickly acquiring a reputation for being a large, slow, and resource-consuming system. Responding to Vista's disappointing adoption rate, Microsoft developed Windows 7, an OS whose interface was similar to that of Vista but was met with enthusiasm for its noticeable speed improvement and its modest system requirements.

History of Windows:

Microsoft's Windows operating system was first introduced in 1985.

Windows 1

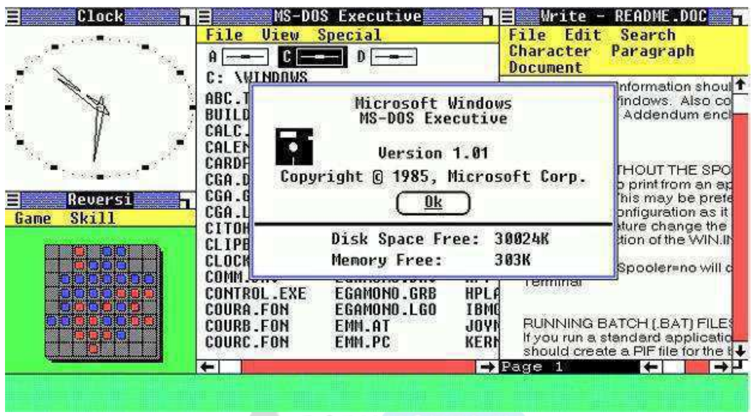


Fig. 5.6 Windows 1

This is where it all started for Windows. The original Windows 1 was released in November 1985 and was Microsoft's first true attempt at a graphical user interface in 16-bit.

Development was spearheaded by Microsoft founder Bill Gates and ran on top of MS-DOS, which relied on command-line input.

Windows 2

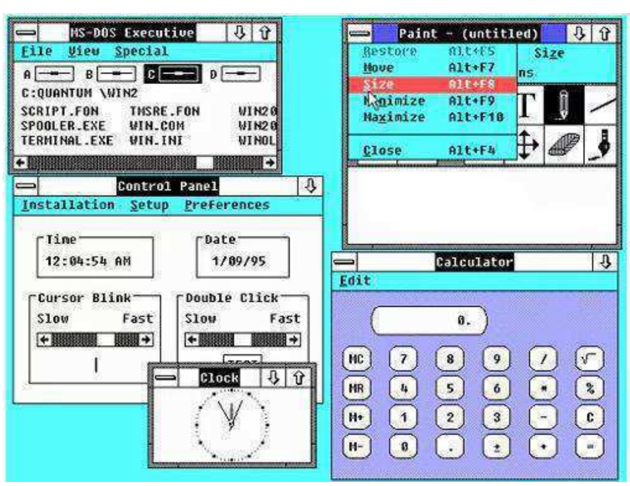


Fig. 5.6 Windows 2

Two years after the release of Windows 1, Microsoft's Windows 2 replaced it in December 1987. The big innovation for Windows 2 was that windows could overlap each other, and it also introduced the ability to minimise or maximise windows instead of "iconising" or "zooming".

The control panel, where various system settings and configuration options were collected together in one place, was introduced in Windows 2 and survives to this day.

Microsoft Word and Excel also made their first appearances running on Windows

Windows 3

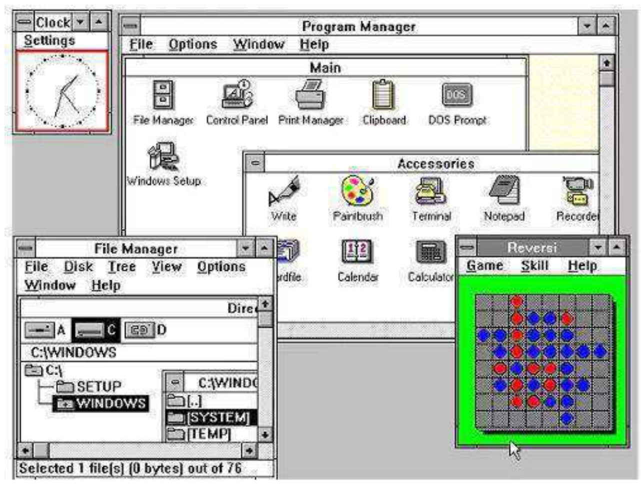


Fig. 5.7 Windows 3

The first Windows that required a hard drive launched in 1990. Windows 3 was the first version to see more widespread success and be considered a challenger to Apple's Macintosh and the Commodore Amiga graphical user interfaces, coming pre-installed on computers from PC-compatible manufacturers including Zenith Data Systems.

Windows 3 introduced the ability to run MS-DOS programmes in windows, which brought multitasking to legacy programmes, and supported 256 colors bringing a more modern, colorful look to the interface.

Windows 3.1

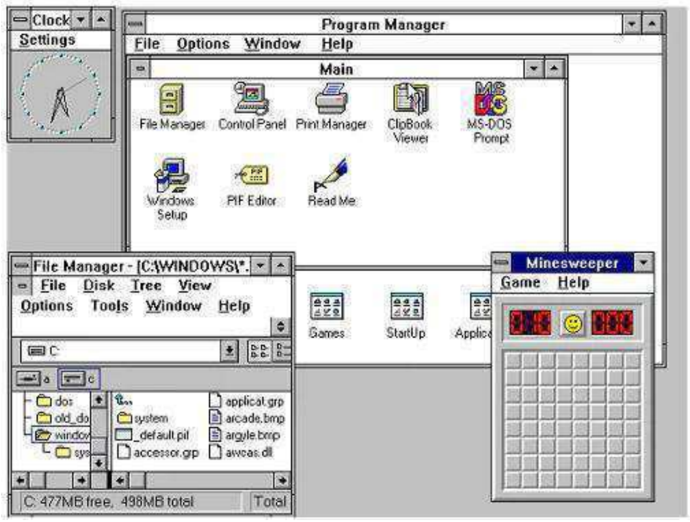


Fig. 5.8 Windows 3.1

Windows 3.1 released in 1992 is notable because it introduced TrueType fonts making Windows a viable publishing platform for the first time.

Minesweeper also made its first appearance. Windows 3.1 required 1MB of RAM to run and allowed supported MS-DOS programs to be controlled with a mouse for the first time. Windows 3.1 was also the first Windows to be distributed on a CD-ROM, although once installed on a hard drive it only took up 10 to 15MB (a CD can typically store up to 700MB).

Windows 95

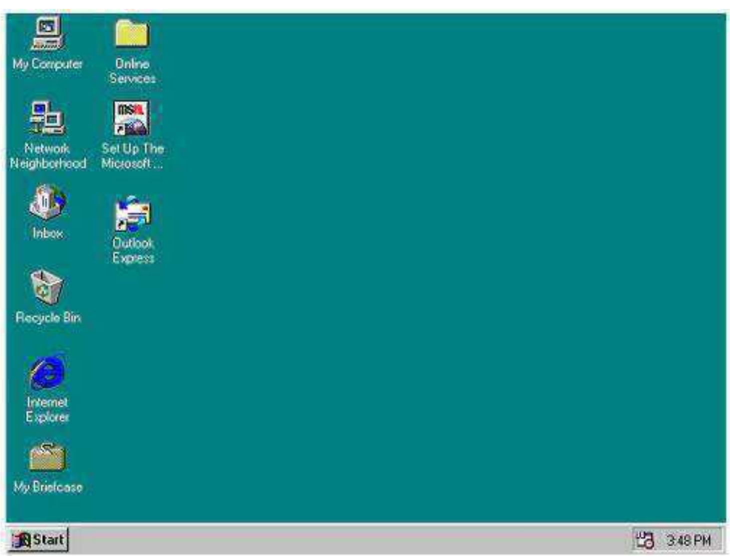


Fig. 5.9 Windows 95

As the name implies, Windows 95 arrived in August 1995 and with it brought the first ever Start button and Start menu.

Windows 98

Released in June 1998, Windows 98 built on Windows 95 and brought with it IE 4, Outlook Express, Windows Address Book, Microsoft Chat and NetShow Player, which was replaced by Windows Media Player 6.2 in Windows 98 Second Edition in 1999.

Windows ME

Considered a low point in the Windows series by many – at least, until they saw Windows Vista

– Windows Millennium Edition was the last Windows to be based on MS-DOS, and the last in the Windows 9x line.

Released in September 2000, it was the consumer-aimed operating system twined with Windows 2000 aimed at the enterprise market. It introduced some important concepts to consumers, including more automated system recovery tools.

Windows 2000

The enterprise twin of ME, Windows 2000 was released in February 2000 and was based on Microsoft's business-orientated system Windows NT and later became the basis for Windows XP.

Windows XP

Arguably one of the best Windows versions, Windows XP was released in October 2001 and brought Microsoft's enterprise line and consumer line of operating systems under one roof.

It was based on Windows NT like Windows 2000, but brought the consumer-friendly elements from Windows ME. The Start menu and task bar got a visual overhaul, bringing the familiar green Start button, blue task bar and vista wallpaper, along with various shadow and other visual effects.

Windows Vista

Windows XP stayed the course for close to six years before being replaced by Windows Vista in January 2007. Vista updated the look and feel of Windows with more focus on transparent elements, search and security. Its development, under the codename "Longhorn", was troubled, with ambitious elements abandoned in order to get it into production.

Windows 7

Considered by many as what Windows Vista should have been, Windows 7 was first released in October 2009. It was intended to fix all the problems and criticism faced by Vista, with slight tweaks to its appearance and a concentration on user-friendly features and less "dialogue box overload".

Windows 8

Released in October 2012, Windows 8 was Microsoft's most radical overhaul of the Windows interface, ditching the Start button and Start menu in favour of a more touch-friendly Start screen.

The new tiled interface saw programme icons and live tiles, which displayed at-a-glance information normally associated with "widgets", replace the lists of programmes and icons. A desktop was still included, which resembled Windows 7.

Windows 8.1

A free point release to Windows 8 introduced in October 2013, Windows 8.1 marked a shift towards yearly software updates from Microsoft and included the first step in Microsoft's U-turn around its new visual interface.

Windows 10

Announced on 30 September 2014, Windows 10 has only been released as a test version for keen users to try. The "technical preview" is very much still a work in progress.

Windows 10 represents another step in Microsoft's U-turn, bringing back the Start menu and more balance to traditional desktop computer users.

Windows operating system structure

The design of operating system architecture traditionally follows the separation of concerns principle. This principle suggests structuring the operating system into relatively independent parts that provide simple individual features, thus keeping the complexity of the design manageable.

Besides managing complexity, the structure of the operating system can influence key features such as robustness or efficiency:

- The operating system possesses various privileges that allow it to access otherwise protected resources such as physical devices or application memory. When these privileges are granted to the individual parts of the operating system that require them, rather than to the operating system as a whole, the potential for both accidental and malicious privileges misuse is reduced.
- Breaking the operating system into parts can have adverse effect on efficiency because of the overhead associated with communication between the individual parts. This overhead can be exacerbated when coupled with hardware mechanisms used to grant privileges.

1. **Simple Structure**

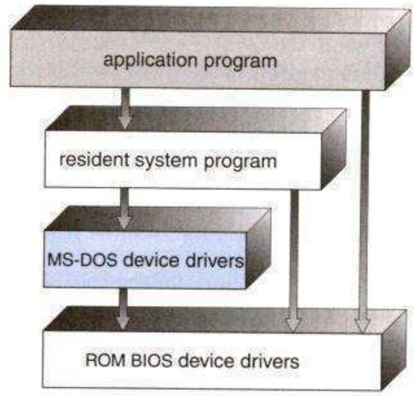


Fig. 5.10 Simple Structure

In MS-DOS, applications may bypass the operating system.

- Operating systems such as MS-DOS and the original UNIX did not have well-defined structures.
- There was no CPU Execution Mode (user and kernel), and so errors in applications could cause the whole system to crash.

2. **Monolithic Approach**

- Functionality of the OS is invoked with simple function calls within the kernel, which is one large program.
- Device drivers are loaded into the running kernel and become part of the kernel.

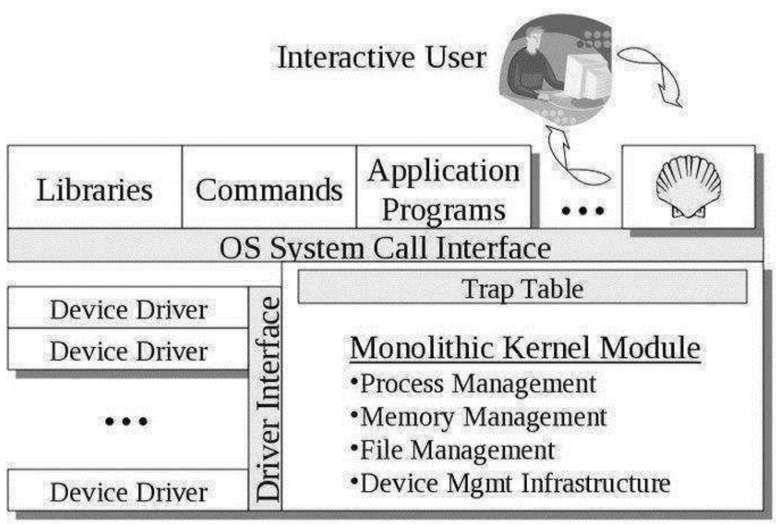


Fig. 5.11 Monolithic Structure

3. **Layered Approach**

This approach breaks up the operating system into different layers.

- This allows implementers to change the inner workings, and increases modularity.
- As long as the external interface of the routines doesn't change, developers have more freedom to change the inner workings of the routines.
- With the layered approach, the bottom layer is the hardware, while the highest layer is the user interface.
- The main advantage is simplicity of construction and debugging.
- The main difficulty is defining the various layers.
- The main disadvantage is that the OS tends to be less efficient than other implementations.

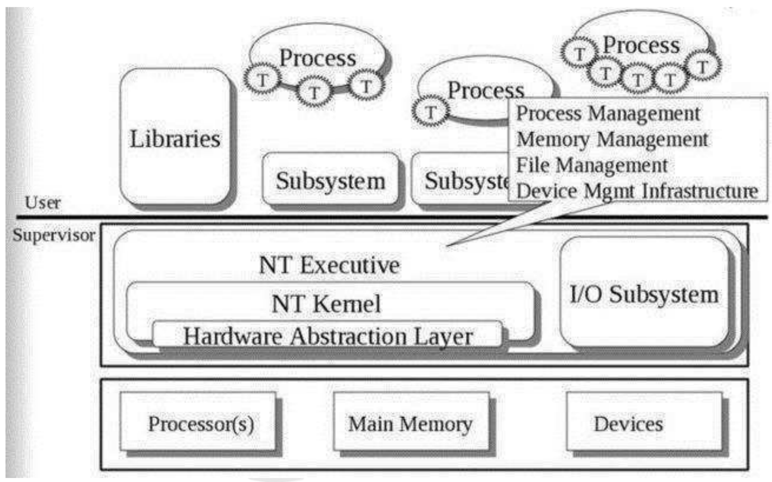


Fig. 5.12 Layered approach

The Microsoft Windows NT Operating System has the lowest level is a monolithic kernel, but many OS components are at a higher level, but still part of the OS.

4. **Microkernels**

This structures the operating system by removing all nonessential portions of the kernel and implementing them as system and user level programs.

- Generally they provide minimal process and memory management, and a communications facility.
- Communication between components of the OS is provided by message passing.

The benefits of the microkernel are as follows:

- Extending the operating system becomes much easier.
- Any changes to the kernel tend to be fewer, since the kernel is smaller.
- The microkernel also provides more security and reliability.

Main disadvantage is poor performance due to increased system overhead from message passing.

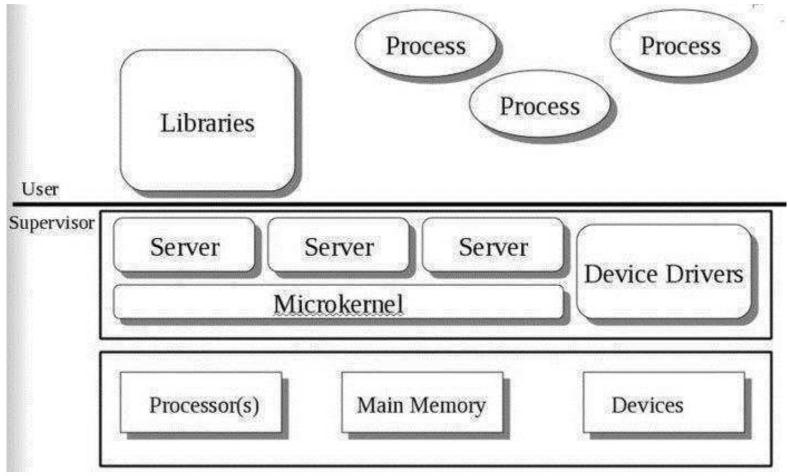


Fig. 5.13 Micro Kernel

Memory management

Every Windows administrator has to field user complaints about client performance. Client-system performance can be affected by factors such as memory, CPU, disk and the network. Of these factors, the most confusing is memory management, which admins need to understand for making informed decisions and troubleshooting. Users typically equate adding memory to resolving performance bottlenecks, and it's relatively cheap and easy to add memory.

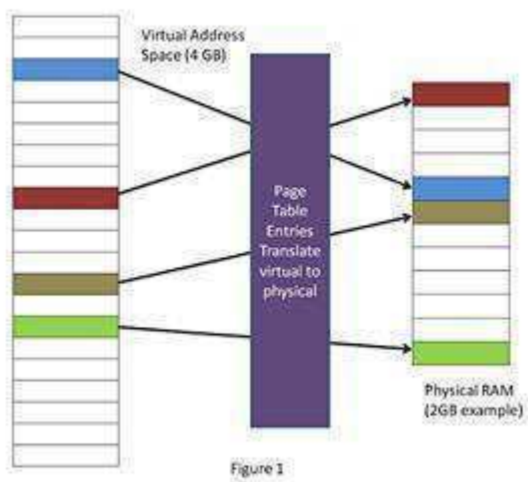


Fig. 5.14 Memory management

Above Fig. shows the memory component of the Windows XP and Windows 7 Task Manager. Note that there are fundamental differences between Windows XP, Vista and Windows 7 Task Manager versions.

It's important to know the difference between physical and virtual memory. Physical memory is the amount of physical RAM available in the computer. Physical memory can be visualized as a table shown in below Figure, where data is stored. Each cell shown in the table is a unique "address" where data is stored.

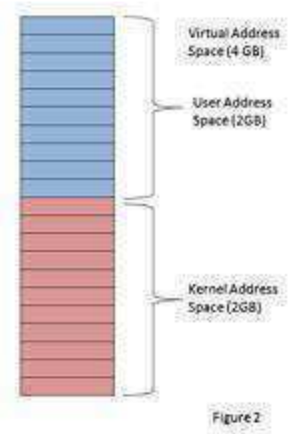


Fig. 5.15 Memory management

Virtual memory essentially allows each process -- applications, dynamic link libraries (DLLs), etc.

To operate in a protected environment where it thinks it has its own private address space. Figure 1 shows the virtual memory table for a process on a computer with 2 GB of RAM. The CPU translates or maps the virtual addresses into physical addresses in RAM using page table entries (PTEs).

Virtual memory limits

The virtual address space for 32-bit architecture has a physical limit of about 4 GB, regardless of the amount of RAM in the computer. Windows divides this into two sections, as shown in Figure 2: user space and kernel space. The addresses in the kernel space are reserved for system processes. Only those in the user space are accessible for applications. So, each application has a virtual memory limit of 2 GB. Again, this is regardless of physical RAM. That means that no process can ever address more than 2 GB of virtual address space by default. Exceeding this limit produces an "out of virtual memory" error and can occur even when plenty of physical memory is available.

Note that, as shown in Figure 2, the use of virtual memory allows the three applications, each with 2 GB of virtual address space, to share the 2 GB RAM in the computer. This is accomplished by paging infrequently used data to disk, then paging it back to RAM when needed.

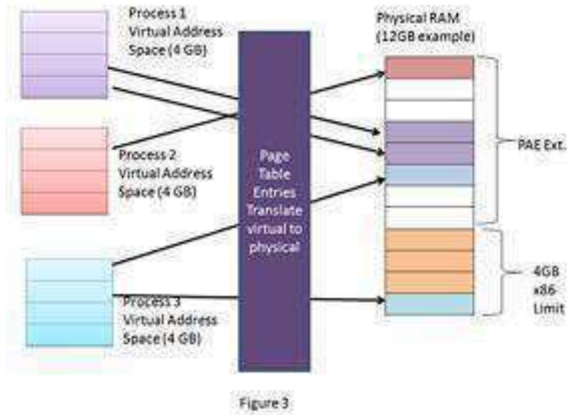


Fig. 5.16 Memory management

Processes will run faster if they reside in memory as opposed to requiring the memory manager page data in from the disk and put it back in memory. Thus, more memory in the system allows more processes to reside in memory and reduces paging from disk.

4. Physical memory management

- Windows reports how much physical memory is currently installed on your computer along with how much memory is available to the operating system and the hardware reserved memory.
- Windows may show that the usable memory may be less than the installed memory (RAM). The indicative Usable memory is a calculated amount of the total physical memory minus "hardware reserved" memory.

Physical Memory

One of the most fundamental resources on a computer is physical memory. Windows' memory manager is responsible with populating memory with the code and data of active processes, device drivers, and the operating system itself. Because most systems access more code and data than can fit in physical memory as they run, physical memory is in essence a window into the code and data used over time. The amount of memory can therefore affect performance, because when data or code a process or the operating system needs is not present, the memory manager must bring it in from disk.

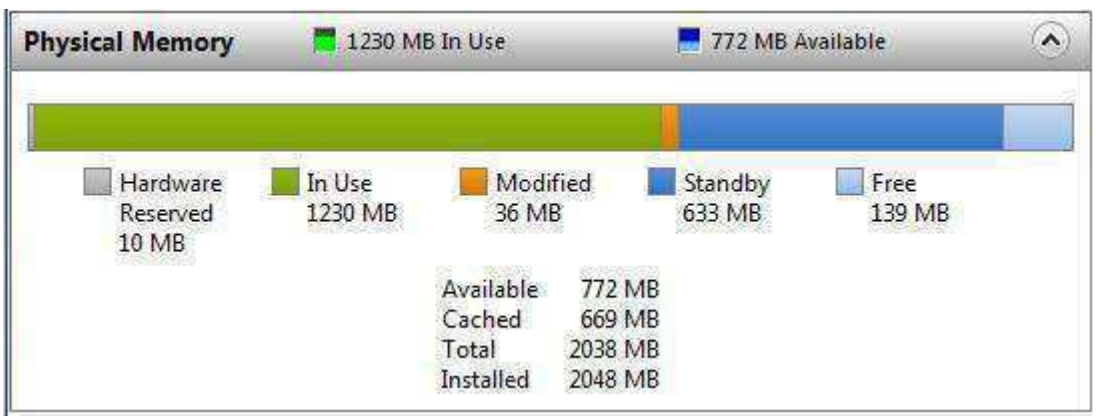
Besides affecting performance, the amount of physical memory impacts other resource limits. For example, the amount of non-paged pool, operating system buffers backed by physical memory, is obviously constrained by physical memory. Physical memory also contributes to the system virtual memory limit, which is the sum of roughly the size of physical memory plus the maximum configured size of any paging files. Physical memory also can indirectly limit the maximum number of processes, which I'll talk about in a future post on process and thread limits.

Memory allocation	Description
Hardware Reserved	Memory that is reserved for use by the BIOS and some drivers for other peripherals
In Use	Memory that is used by process, drivers, or the operating system
Modified	Memory whose contents must go to disk before it can be used for another purpose
Standby	Memory that contains cached data and code that is not actively in use
Free	Memory that does not contain any valuable data and that will be used first when processes, drivers, or the operating system need more memory

Memory allocation	Description
Available	Amount of memory (including standby and free memory) that is immediately available for use by processes, drivers, and the operating system
Cached	Amount of memory (including standby and modified memory) that contains cached data and code for rapid access by processes, drivers, and the operating system
Total	Amount of physical memory that is available to the operating system, device drivers, and processes
Installed	Amount of physical memory installed in the computer

To find out how memory is being used on your computer, type **Resource Monitor** in start search and hit Enter.

Click the Memory tab, and view the Physical Memory section at the bottom of the page.



5. **Memory management system calls.**

System Calls

The system call provides an interface to the operating system services.

Application developers often do not have direct access to the system calls, but can access them through an application programming interface (API). The functions that are included in the API invoke the actual system calls. By using the API, certain benefits can be gained:

- Portability: as long a system supports an API, any program using that API can compile and run.
- Ease of Use: using the API can be significantly easier then using the actual system call.

System Call Parameters

Three general methods exist for passing parameters to the OS:

1. Parameters can be passed in registers.
2. When there are more parameters than registers, parameters can be stored in a block and the block address can be passed as a parameter to a register.
3. Parameters can also be pushed on or popped off the stack by the operating system.

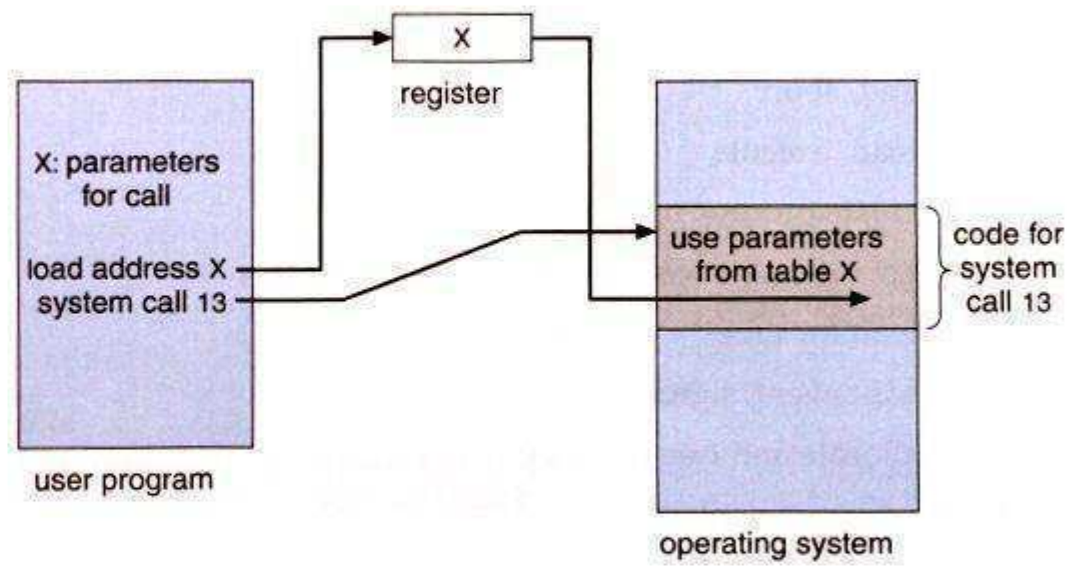


Fig. 5.17 System Call

Types of System Calls

There are 5 different categories of system calls:

Process control, file manipulation, device manipulation, information maintenance and communication.

Process Control

A running program needs to be able to stop execution either normally or abnormally. When execution is stopped abnormally, often a dump of memory is taken and can be examined with a debugger.

File Management

Some common system calls are create, delete, read, write, reposition, or close. Also, there is a need to determine the file attributes – get and set file attribute. Many times the OS provides an API to make these system calls.

Device Management

Process usually requires several resources to execute, if these resources are available, they will be granted and control returned to the user process. These resources are also thought of as devices. Some are physical, such as a video card, and others are abstract, such as a file.

User programs request the device, and when finished they release the device. Similar to files, we can read, write, and reposition the device.

Information Management

Some system calls exist purely for transferring information between the user program and the operating system. An example of this is time, or date.

The OS also keeps information about all its processes and provides system calls to report this information.

Communication

There are two models of interposes communication, the message-passing model and the shared memory model.

- Message-passing uses a common mailbox to pass messages between processes.
- Shared memory use certain system calls to create and gain access to create and gain access to regions of memory owned by other processes. The two processes exchange information by reading and writing in the shared data.

