

Case Study on Windows

1. What is Windows Operating System?

- 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.

Case Study on Windows

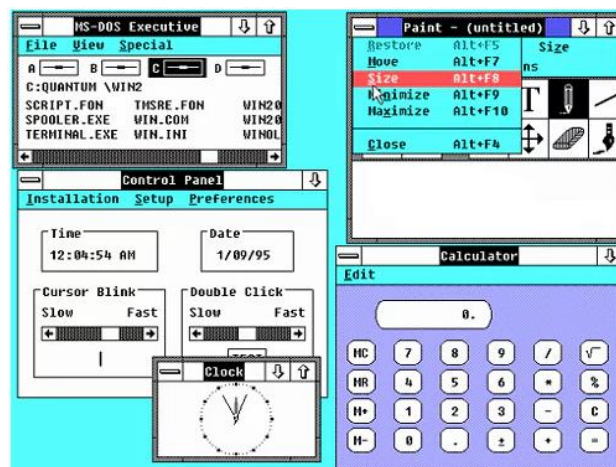
Windows 1



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



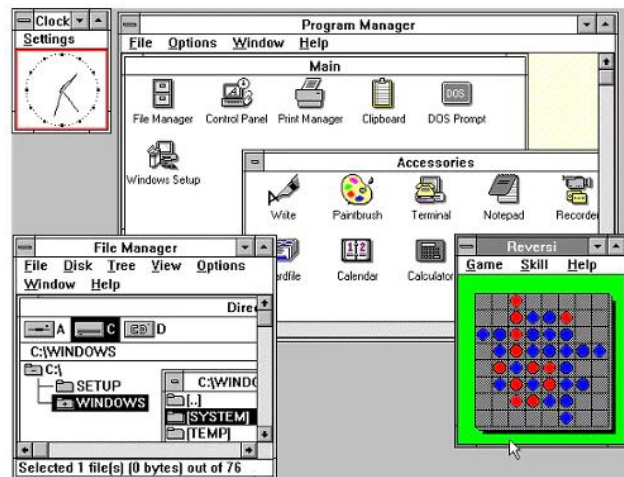
Two years after the release of Windows 1, Microsoft's Windows 2 replaced it in December 1987. The big innovation for Windows 2 was that windows could overlap each other, and it also introduced the ability to minimise or maximise windows instead of "iconising" or "zooming".

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

Case Study on Windows

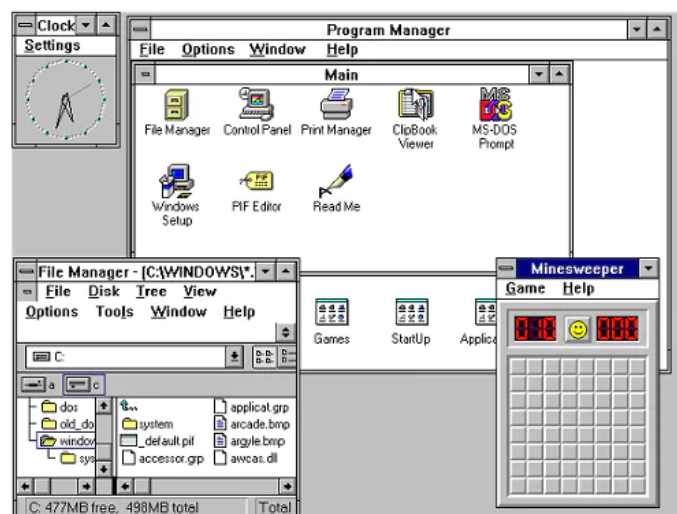
Windows 3



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

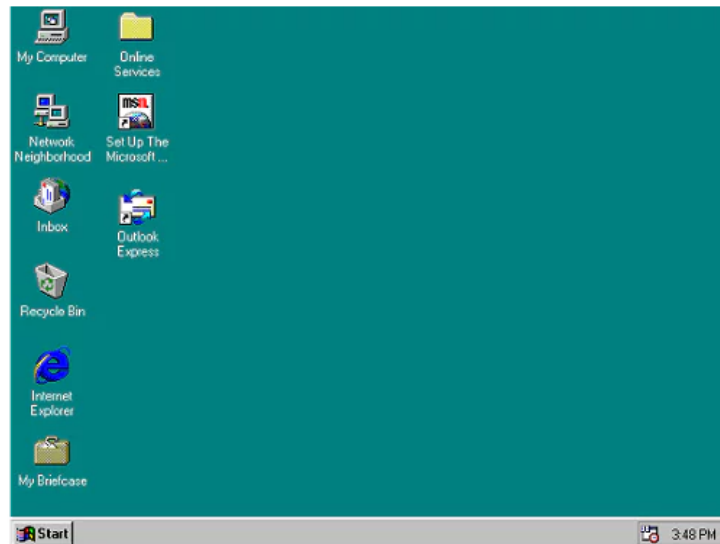


Windows 3.1 released in 1992 is notable because it introduced TrueType fonts making Windows a viable publishing platform for the first time.

Case Study on Windows

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



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.

Case Study on Windows

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.

Case Study on Windows

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.

2. Explain operating system structure with diagram.

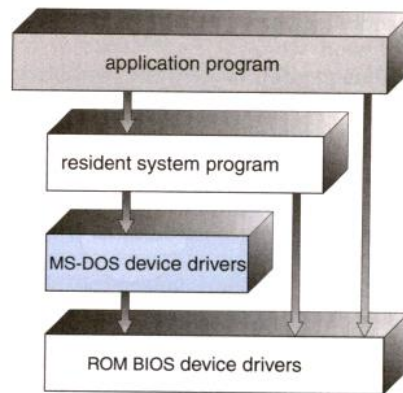
The design of an operating system architecture traditionally follows the separation of concerns principle. This principle suggests structuring the operating system into relatively independent parts that provide simple individual features, thus keeping the complexity of the design manageable.

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.

Case Study on Windows

1. Simple Structure

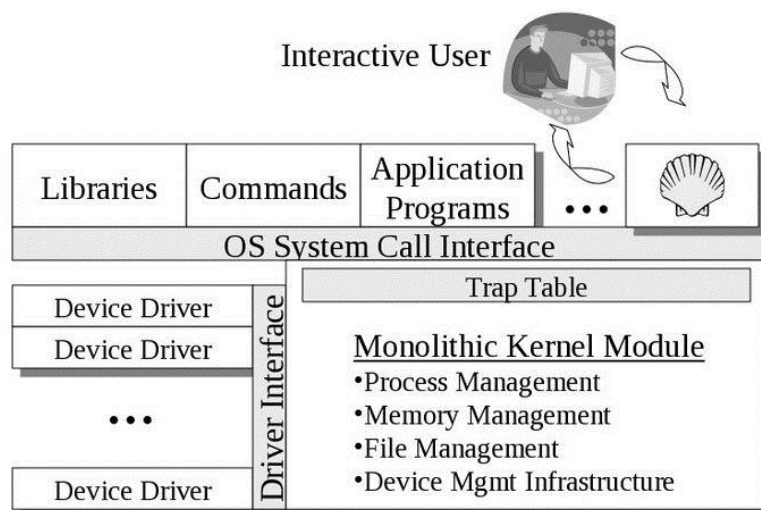


In MS-DOS, applications may bypass the operating system.

- 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.



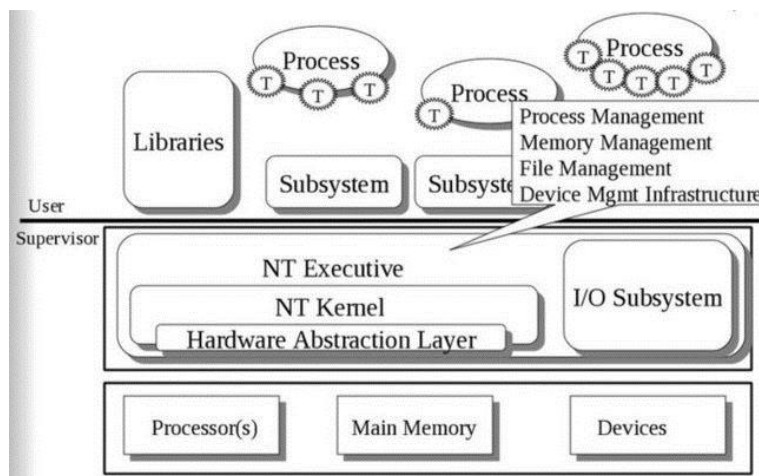
A monolithic kernel, such as Linux and other Unix systems.

Case Study on Windows

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 don'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.



The Microsoft Windows NT Operating System. 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.

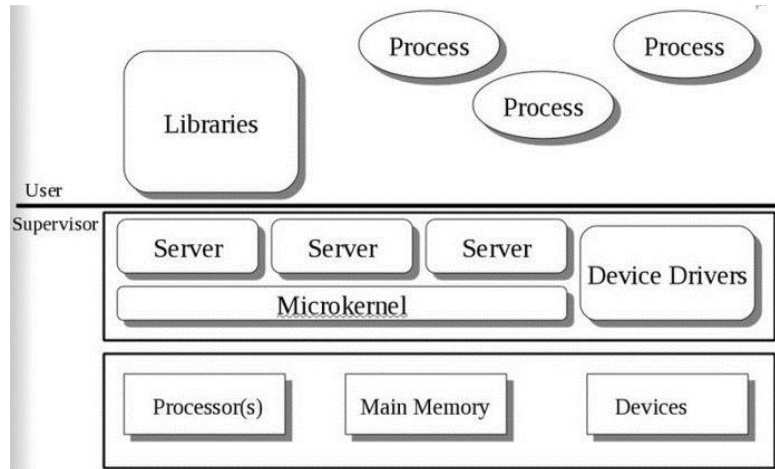
The benefits of the microkernel are as follows:

- Extending the operating system becomes much easier.
- Any changes to the kernel tend to be fewer, since the kernel is smaller.

Case Study on Windows

- The microkernel also provides more security and reliability.

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



A Microkernel architecture.

3. Write a note on 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.

Case Study on Windows

Figure 1

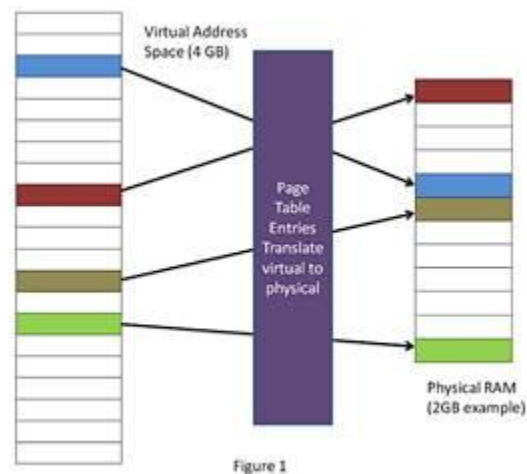
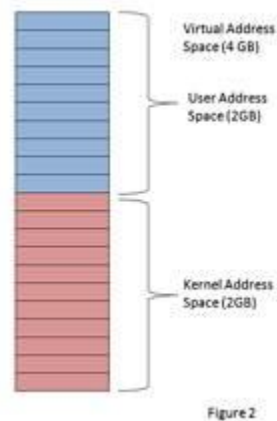


Figure 1 shows the memory component of the Windows XP and Windows 7 Task Manager. Note that there are fundamental differences between Windows XP, Vista and Windows 7 Task Manager versions.

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 Figure 2, where data is stored. Each cell shown in the table is a unique "address" where data is stored.

Figure 2



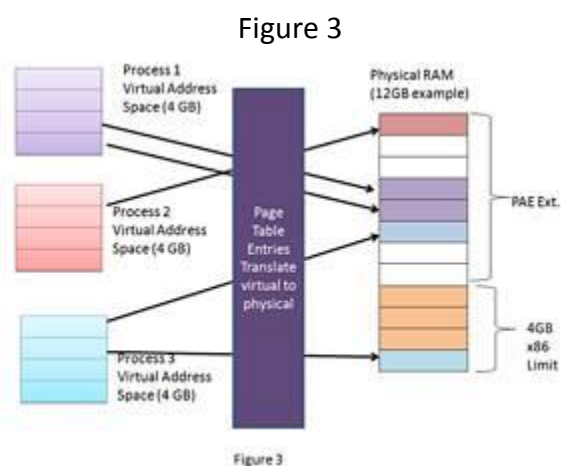
Case Study on Windows

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.



Case Study on Windows

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. Explain 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.

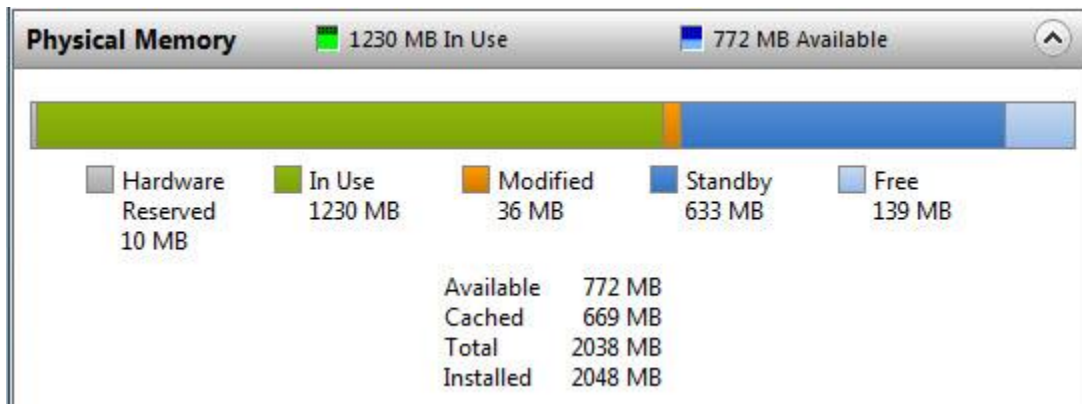
Memory allocation	Description
Hardware Reserved	Memory that is reserved for use by the BIOS and some drivers for other peripherals
In Use	Memory that is used by process, drivers, or the operating system
Modified	Memory whose contents must go to disk before it can be used for another purpose
Standby	Memory 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

Case Study on Windows

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

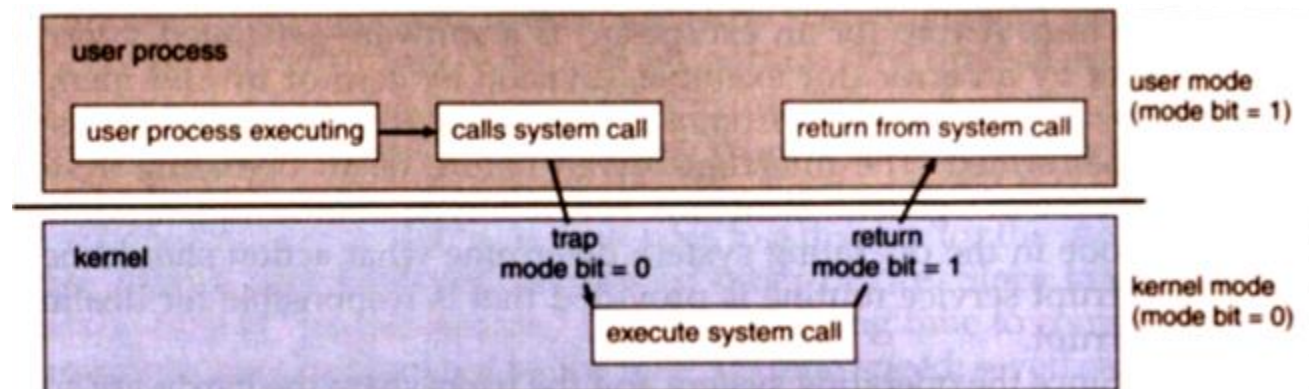
To find out how memory is being used on your computer, type **Resource Monitor** in start search and hit Enter.

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



5. Explain any 5 memory management system calls.

System Calls



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

Case Study on Windows

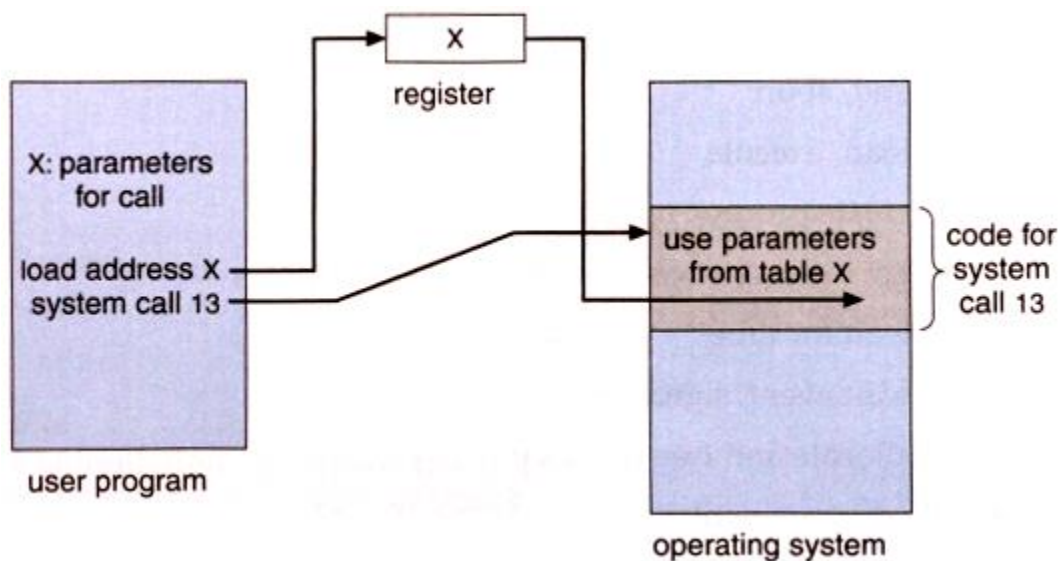
Application developers often do not have direct access to the system calls, but can access them through an application programming interface (API). The functions that are included in the API invoke the actual system calls. By using the API, certain benefits can be gained:

- 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 than 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.



Types of System Calls

There are 5 different categories of system calls:

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

Case Study on Windows

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 interprocess 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.