Case Study

Submitted by:

Ananya Chakraborty Gourab

Course ID: CSE315

Course Name: Design of Operating System

Section:03

Submission Date: 20.08.22

Introduction:

Microsoft Windows, usually known as Windows or Windows OS, is a personal computer operating

System (OS) developed by Microsoft Corporation (PCs). The Windows operating system, which

Included the first graphical user interface (GUI) for IBM-compatible PCs, quickly dominated the

PC industry. Windows is installed on around 90% of personal computers. Windows NT and

Windows It is active Microsoft Windows families; these may contain subfamilies, such as

Windows Server or Windows Embedded Compact (Windows CE). Windows 9x, Windows

Mobile and Windows Phone are all defunct Microsoft Windows family.

A brief history of windows operating system:

The initial version of Windows, launched in 1985, was just a graphical user interface (GUI)

That served as an extension to Microsoft's existing disk operating system, or MS-DOS,

Based in part on licensed concepts used by Apple Inc. for its Macintosh System Software.

• Windows allowed DOS users for the first time 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.

• Following versions included more capability, such as native Windows File Manager,

Program Manager and Print Manager applications, as well as a more dynamic interface.

Microsoft also created specific Windows packages for enterprises, such as the networkable windows for Workgroups and the high-powered Windows NT.

• A perfect example is Windows 95. It was released for the general public in 1995, which

Completely merged Windows and DOS and had built-in Internet capabilities, including the

World Wide Web browser Internet Explorer.

• Microsoft unified its different Windows packages under a single banner with the 2001

Release of Windows XP, offering multiple editions for consumers, enterprises, multimedia

developers and others. Windows XP ditched the long-used Windows 95 kernel (core

Software code) in favor of a more capable code base, as well as a more user-friendly

interface and enhanced program and memory management.

• The very successful XP standard was followed in late 2006 by Windows Vista, which had

a problematic launch and encountered significant market opposition, soon gaining a

reputation for being a huge, sluggish, and resource-consuming system.

• In response to Vista's low adoption rate, Microsoft developed Windows 7 in 2009, an

Operating system with an interface identical to Vista but received positive feedback for its

Notable performance boost and low system requirements.

• In 2012, Windows 8 introduced a start screen with programs shown as tiles on a grid, as

Well as the option to synchronize settings, allowing users to go on to another Windows 8

System and utilize their chosen settings.

Microsoft unveiled Windows 10 in 2015, which included Cortana, a digital personal

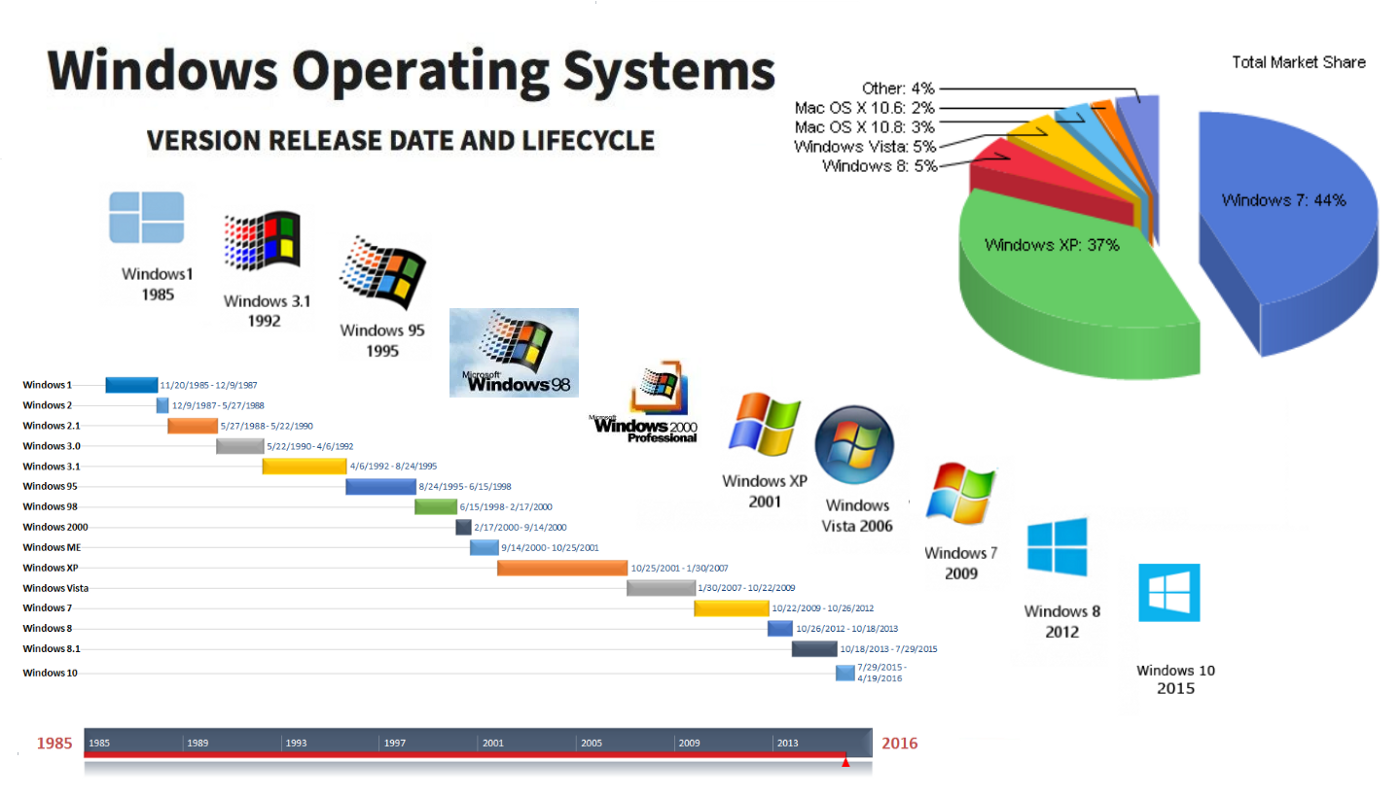
Assistant similar to Apple's Sire, as well as the Web browser Microsoft Edge, which

replaced Internet Explorer.

• Microsoft also stated that Windows 10 will be the final version of Windows, however

Windows 11 was released in October 5, 2021.

Timeline of Windows Operating System:



Windows 7:

Windows 7 is a personal computer operating system developed by Microsoft. It is the successor to the Windows Vista operating system, which debuted in 2006. The user's computer's operating system allows it to manage applications and execute vital functions. It is also a Graphical User

Interface (GUI) that allows you to visually engage with the features of your computer in a logical, enjoyable, and simple manner.

Windows 7 simplified the PC experience by improving previews on the Task bar, rapid searching for files or video, and simple sharing via Home-Group networking. It also claims to have increased speed by supporting 64-bit processors, which is becoming more common in desktop PCs. As a result, it gained critical praise, with reviews praising the operating system as a significant advance over its predecessor due to enhanced speed, a more intuitive UI, fewer User Account Control popup, and other platform enhancements.

Windows 7 design goals:

While developing Windows 7, Microsoft developed some well-defined design goals. According to Samuel Moreau, Microsoft's VP of Design and Research, customers buy Windows for the broad choice of software rather than the operating system itself. As a result, it is the job of third-party developers to guarantee that the overall Windows experience is all about making people successful.

1. Reduce concepts:

This design goal for Windows 7 describes what consumers should not do is blame themselves when they lack confidence in Windows. The answer is to reduce the number of "concepts" that are duplicates. In Windows Vista, for example, there are several outlook elements on the screen, such as the desktop, start menu, quick launch, taskbar, and system tray. It has been reduced to just one item in Windows 7.

B. Small things matter

Basically, a lot of minor things viewed by a lot of individuals over time add up. This can,

Of course, be good or bad. In Windows 7, the emphasis is on images rather than a browser. The command bar is softly colored to do this. Borders have been eliminated, but padding has been added. As a result, a few extra images are displayed for the same window size.

C. Resolve distractions

This design goal describes about resolving distractions rather than discoverability. In

Windows 7, for example, jump lists are activated by right-clicking the taskbar icons. In

Early iterations, however, they contained a split button that triggered jump lists. The issue with this option is that it displayed an excessive number of arrows on the taskbar area. To make the move to the right-click solution, the designers had to overcome their worries and trust that users would be able to find the feature.

D. Time matters

This design goal describes about the importance of time management. It makes no

Difference how lovely the application is if it is sluggish. Previously, changing shared

Printers in older Windows operating systems such as Windows Vista required 30 to 40

Steps, however Windows 7 reduces this to 3 to 5 steps.

E. Value the full lifecycle of the experience

This design goal suggests that the complete experience should be considered, not just when using the program, but from installation through removal or update.

F. be great at look and do

This final design goal implies that when connecting a phone to a computer, it should not

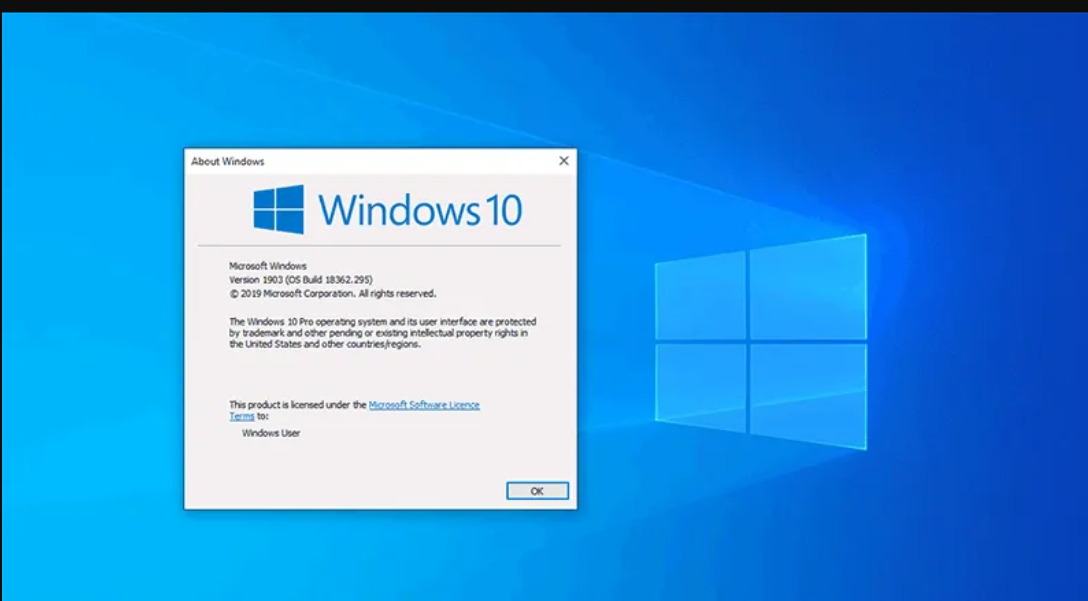
Be treated as just another "generic device." The product symbol shines through the UI in

Windows 7 to make people proud of their gadget.

Windows 7 structure

Core of windows 7

The CPU, which stands for Central Processing Unit, is the most important component of every Computer. The CPU receives instructions and then executes them. If a CPU can only handle one Set of instructions at a time, it has a single core. A CPU is termed dual-core if it can process two Sets of instructions at the same time. A quad-core CPU can process four sets of instructions at the same time. The more cores a processor has, the more instruction sets it can handle at the same time. The Windows core features a shared set of APIs that enable complete binary compatibility across a wide range of platforms for the first time. For almost a decade, Microsoft has been launching whole sets of operating systems (OS) with numerous similarities, including Windows Phone, Windows CE, and Windows RT. The majority of them were the same, but the underlying layer differed. Windows Core is a very compact and condensed subset of all Windows versions. Windows 7 was created to be compatible with today's multi-core CPUs. Windows 7 32-bit editions can handle up to 32 processing cores, while 64-bit editions can support up to 256 processor cores. Commercial servers, workstations, and other high-end PCs may include several physical processors. Windows 7 Professional, Enterprise, and Ultimate support two physical CPUs, which results in the greatest performance on these PCs. Only one physical CPU will be recognized by Windows 7 Starter, Home Basic, and Home Premium.



Windows 7 file system structure:

A file system is a method of organizing, storing, and naming data on various storage mediums. Windows 7 makes use of the NTFS file system, which is the most widely used system today. The MFT (Master File Table) is a table that maintains the addresses of sectors where file contents are physically stored. This is critical information; without it, the operating system has no idea where the requested object resides or where it should move the hard drive's reading head. Windows Without a doubt, the most critical folder on the hard disk's system partition. It saves all system executables, drivers, and libraries, among other things. As a result, deleting files from that folder is strongly discouraged. It's worth noting that this folder might have a different name.

Windows. old

The folder contains the previous operating system. If Windows 7 was installed over an existing system without first formatting the drive, the folder is formed. This folder is unnecessary for the vast majority of users and may be safely removed.

Users

User profiles are stored in this folder. It is usually found in the system partition's root folder. It has a handful of regular folders as well as one for each user account in the system. These folders include a variety of user subfolders such as the Desktop, Documents, Pictures, Favorites, and so on. The contents of the folders are entirely up to the account's owner. Other subfolders hold information from various apps, browsers, and so on: AppData, ApplicationData, Cookies, Local Settings, and so on. These folders are hidden, and users should not edit their contents unless they are absolutely certain of what they are doing.

Boot

This folder includes boot files for the operating system. It is hidden, and changing its contents is not advised.

Program data

In the folder, installed apps save their data, setup files, and other information. You obviously do not want to remove anything from it.

Recovery

This folder includes the image needed to access the system Recovery Console. It is also concealed and the user cannot modify its contents.

$Recycle.bin

This is the Windows Recycle Bin folder. Deleted files are saved in this folder. Although the folder is hidden, the user can safely remove both its contents and the folder itself. This is equivalent to emptying the Recycle Bin or a portion of it. It's worth noting that each hard disk partition has its own $Recycle. Bin folder that saves data that have been removed from it.

Kernel approach of windows 7

Memory analysis is a well-established approach for malware analysis that is increasingly being utilized for operating system incident response. The Volatility memory analysis framework (The

Volatility Foundation, 2014) comes with a variety of Windows profiles pre-installed. For example, if a user's studying a Windows 7 image, the profile may be Win7SP1x64.

Microsoft Visual Studio is the most commonly used compiler on Windows computers (MSVCC). This PDB file contains a variety of useful information for a memory analysis framework, including struct members/memory layout to interpret memory contents, global constants to order list, function addresses to locate memory functions and enumeration to represent one of a set of choices using an integer. The question of how to obtain this struct layout information mechanically remains unanswered.

The kernel technique is required for developing any operating system in order to tackle this challenge. Microsoft has a fantastic kernel strategy that has made a major difference in the

development of Windows 7.

Microsoft Windows 7 kernel versions contains the following parts:

Characterizing variability version, global constant version & identifying binary version

Kernel characterization version variability:

The following process occurs in this version

• The \_EPROCESS. Vad Root is the Vad's placement inside the process. This is used to Count the number of process allocations.

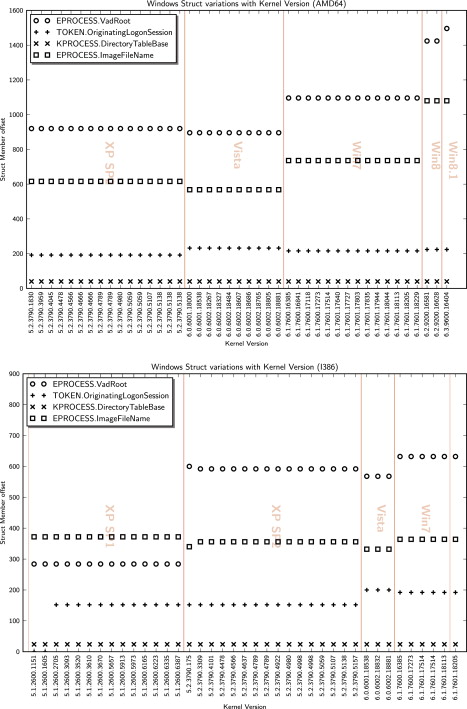
• The \_KPROCESS.DirectoryTableBase specifies the location of the Directory Table Base

(i.e., the value put into the CR3 register), which is required for the Virtual Address Space

abstraction to be built.

• The \_EPROCESS.ImageFileName variable contains the file name of the currently

Executing program. This field may, for example, include "csrss.exe."



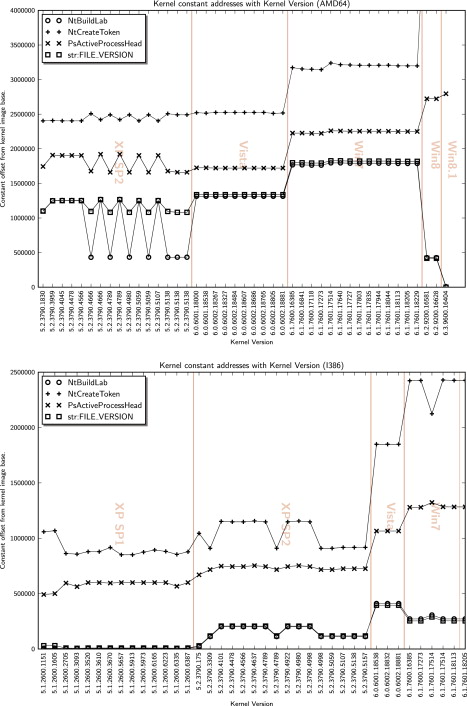
Kernel global constants variability:

The following process occurs in this version

• The NT version string (for example, "7600.win7 rtm.090713-1255") is stored in NtBuildLab. This is used to identify the kernel that is currently operating.

• PsActiveProcessHead is the active process list's head. This is necessary in order to list the processes that are currently executing.

• An example of a kernel function is NtCreateToken. This is usually found in the text part of the PE file.



Windows kernel binary identification

In short, the Kernel Binary Auto detection technique, is as follows:

• Scan the picture for common Windows executable names (e.g. "csrss.exe", "cmd.exe" and so on). The Aho-Corasick method is used in this scan to look for all strings at once.

• For each strike, brute drive the DTB across all ten conceivable offsets. The KUSER

SHARED DATA.Nt Major Version and KUSER SHARED DATA.Nt Minor Version elements are used to validate the DTB. It is safe to hardcode this struct since it must be found in a fixed position in memory and always have the same layout (Skape, 2005).

• As a result, without knowing anything about the profile or the kernel version, we can validate the DTB and kernel address space.

Universal app platform

The UAP is a collection of contracts and their many variants. This is the new platform for developers to create their applications. The UAP is based on the Windows Core, which is a set of Low-Level APIs organized as Contracts.

Components of Windows 7

Windows 10 consists of the following components:

Desktop

It is the very first screen that you will see once the windows start. Here you will see

“My Computer”, “My Documents”, “Start Menu”, “Recycle Bin”, and the shortcuts

of any applications that you might have created.

Taskbar

At the bottom, you will see a row which is known as the taskbar. It has the currently running applications, you can also pin applications that you frequently use by using an option Pin to Taskbar”. Start Menu this is located in the bottom left corner of Windows OS GUI. This is the place where the user can search for any setting and for any application for their use. Users can uninstall or repair applications from the control panel. The user can do a lot of activities just by searching through the start menu.

My computer

When you double click on “My Computer” menu, it will let you navigate between your different computer drives and the control panel to ols. You can see and manage the contents that are inside your drive.

Recycle Bin

When you delete an item from any of your drives by making use of “delete” button or even by simply clicking right clicking and selecting “delete” option, it is not deleted completely, instead, it is moved to “Recycle Bin” folder of Windows. You can recover your content if you have deleted it by mistake from here or if you choose to delete the items from here, it will get deleted permanently. Should you wish to delete the item in first go itself without moving it to recycle bin, you canuse the key “Shift+Del • Start Menu:

By clicking the start menu, in the bottom left corner of the screen, a vertical

window consisting of the recently opened applications and saved locations will pop- up. Although the Start Menu was a major component of Windows before Windows

8, It was removed from Windows 8 and then brought back in Windows 10.

Maximize/Minimize/Close Buttons:

These buttons are located at the top right corner of our opened documents, and the area used to close, minimize, or maximize the document window. They help us jump from one task to another fast and let us decide whether we want to close an application or resize its area on the screen or just hide it for a few

My Computer Right Click Menu:

When we right-click on My Computer or any other file or folder, we get a menu where we can look into different options related to that specific file, for example, Properties, etc.

Shortcut:

A shortcut creates a button or icon which typically is located on the Desktop. By clicking on this Shortcut, we can quickly open the document or application of which it is a shortcut. It helps us save the tedious task of going to the main directory again and again and saves our time.

Mouse Functions:

The mouse is an input device which is essential in the working of a computer. It performs several important functions on Windows like Scrolling, Right and Left Clicks, etc. It performs another very important function of modern windows which is pointing towards different things and giving special instructions whenever needed.

Highlight:

When we have opened a document, we can easily highlight the required portion of

our document by using Mouse. It is essential for documents and helps keep track of

useful information.

Copy/Cut/Paste:

These options are one of the most essential components of Windows. The copy is

used to copy a portion of a document from one document to another or a file or

folder from one location to another. The paste is used to past e the copied item on

the desired location. While Cut is used to move an item to our desired location in

the computer.

Toolbar:

The toolbar is a simple row where we can see different options to customize the

look of our opened window. It has two types, Formatting Toolbar, and Standard

Toolbar. The standard toolbar consists of options like new documents, save a

document, etc. While Formatting Toolbar consists of options like font size, font

type, etc.

Drag/Drop:

Dragging an object means to move an object (file or folder) from one location to another and when we reach our desired location, then we can drop the object to that location. It is one of the most used features of windows as users have to move files from one location to another.

File Extensions:

File extensions are used to define the type of the file. For example, an image file will have an extension of .jpg, .jpeg and a Word document will have an extension .docx, .xls, .txt etc. Users could have different types of extensions and these extensions help you decide the type of software that will be used to access these files.

Multitasking:

The term Multitasking means to run more than one file or application on Windows at the same time. It is a very important component of Windows which saves our time as well as allows us to perform more tasks at the same time.

Virtual Keyboard:

A virtual keyboard is a software through which we can see a keyboard on our screen and use it with our Mouse. It is mostly used in the cases when the keyboard is not working properly, or one is using windows on a touch device.

Disk Drives:

Disk Drives are drives used to store applications and files. Hard Drives and Floppy Drives are used for this purpose. They are very important for the user’s instructions as well as your hardware to work properly.

Defragmenting Hard Drives:

Defragmenting a Drive means to erase all the data from that drive. It is also another important component of windows as users need to clean up their hard drives from time to time and it also comes as a built-in utility.”

Shell of operating system

The Windows shell is the graphical user interface for the Microsoft Windows operating system.

Its readily identifiable elements consist of the desktop, the taskbar, the Start menu, the task switcher, and the AutoPlay feature. On some versions of Windows, it also includes Flip 3D and the charms. In Windows 7, the Windows Shell Experience Host interface drives visuals like the Start Menu, Action Center, Taskbar, and Task View/Timeline. However, the Windows shell also implements a shell namespace that enables computer programs running on Windows to access the computer's resources via the hierarchy of shell objects.

Desktop

Windows Desktop is a full-screen window rendered behind all other windows. It hosts the user's wallpaper and an array of computer icons representing:

1. Files and folders: Users and software may store computer files and folders on Windows desktop. Naturally, on a newly installed version of Windows, such items do not exist.

Software installers commonly place files known as shortcuts on the desktop, allowing users to launch installed software. Users may store personal documents on the desktop.

2. Special folders: Apart from ordinary files and folders, special folders (also known as "shell folders") may appear on the desktop. Unlike ordinary folders, special folders do not point to an absolute location on a hard disk drive. Rather, they may open a folder whose location differs from computer to computer (e.g. Documents), a virtual folder whose contents is an aggregate of several folders on disk (e.g. Recycle Bin or Libraries) or a folder window whose content is not files, but rather user interface elements rendered as icons for convenience (e.g. Network). They may even open windows that do not resemble a folder at all (e.g. Control Panel).

Taskbar

Windows taskbar is a toolbar-like element that by default, appears as a horizontal bar at the bottom of the desktop. It may be relocated to the top, left or right edges of the screen. Starting with Windows 98, its size can be changed. The taskbar of Windows 7 is as follows:

1. Start button: Provides access to the Start menu.

2. Quick Links menu: Invoked by right-clicking on the Start button. Grants access to several frequently used features of Windows, such as accessing the desktop, Settings, Windows

Command Processor, Windows Power Shell, and File Explorer.

3. List of open windows: Along the length of the taskbar, open windows are represented by their corresponding program icons. And once pinned, they will remain even after their respective windows are closed.

4. Shortcuts: An update to Windows 95 and Windows NT 4 added a Quick Launch Bar that can hold file, program, and action shortcuts, including by default the "show desktop" command.

5. Desk bands: Toolbars provided by Windows or other programs for easier access to that program's functions.

6. Notification area: Allows programs to display icons representing their status as well as pop-up notifications associated with those icons. By default, Windows volume control, network status, Action Center, date, and time are displayed in this area.

7. "Show desktop" button: Allows users to access their desktops. It is moved from the left of the Taskbar as a Quick Launch shortcut to the rightmost side as its own dedicated hover button in Windows 7.

8. Task View: A function in Windows 7 allowing the user to view and manage open windows and virtual desktops.

9. Action Center: Introduced in Windows 7, the Action Center gave notifications and tips on boosting computer performance and security.

Task switching

Windows 7 has a unified task switcher called Task View, which manages not only application windows but virtual desktops as well.

Start menu

Starting with Windows 95, all versions of Windows feature a form of Start menu, usually by this very same name.

Process Creation

The fundamental Windows process management function is Create Process, which creates a process with a single thread. It is necessary to specify the name of an executable program file as part of the Create Process call.

Due to Windows’s layered architecture and the presence of environment subsystems, process creation is quite complex in windows 7. A detailed description is given below:

● The AWin32 application calls Create Process().

● A number of parameter conversions and behavioral conversions are done from the Win32 World to the NT world.

• Create Process () then calls the NtCreateUserProcess() API in the process manager of the NT executive to actually create the process and its initial thread.

• The process manager calls the object manager to create a process object and returns the object handle toWin32. It then calls the memory manager to initialize the address space of the new process, its handle table, and other key data structures, such as the process environment block (PEBL) (which contains internal process management data).

• The process manager calls the object manager again to create a thread object and returns the handle to Win32. It then calls the memory manager to create the thread environment block (TEB) and the dispatcher to initialize the scheduling attributes of the thread, setting its state to initializing.

• The process manager creates the initial thread startup context (which will eventually point to the main () routine of the application), asks the scheduler to mark the thread as ready, and then immediately suspends it, putting it into a waiting state.

• A message is sent to the Win32 subsystem to notify it that the process is being created. The subsystem performs additionalWin32-specific work to initialize the process, such as computing its shutdown level and drawing the animated hourglass or “donut” mouse cursor.

• Back in CreateProcess(), inside the parent process, the ResumeThread() API is called to wake up the process’s initial thread. Control returns to the parent.

• Now, inside the initial thread of the new process, the user-mode link loader takes control (inside ntdll.dll, which is automatically mapped into all processes). It loads the entire library dependencies (DLLs) of the application, creates its initial heap, sets up exception handling and application compatibility options, and eventually calls the main() function of the application.

Process Termination

When you start an app, the operating system creates a process for an executable file of the app. It contains the program code and its current activity. Windows assigns a special number known as Process Identifier (PID) which is unique for every process. There are a number of reasons you might want to kill a process, and different methods can be used to terminate it. Here is how it can be done.

Task Manager

1. Open Task Manager.

2. Click on "More details" in the bottom right corner to enter Full view mode.

3. Select the desired app in the app list.

4. Click on the End task button or hit the Del key on the keyboard.

This is Task Manager's most well known method. The same can be done from the Details tab. It is a special tab which lists process names instead of app names. Select a process in the list and either Click on the End process button or hit the Del key. Using the End Task button means Windows first tries to see for a certain timeout if the process has really stopped responding, and attempts to collect a crash or memory dump of the process. It then terminates the app.

Taskkill

Another classic method to close a process is the console tool taskill. TaskKill is a Windows console tool which can shut down processes from the command line, batch files, scripts and shortcuts. The program uses the same techniques as Task Manager, so won’t work in every situation, but it’s still Handy for automating shutdowns. Some processes are running as Administrator (elevated). In order to kill them, i need to open an elevated command prompt instance.

1. Open the command prompt as the current user or as Administrator.

2. Type task list to see the list of running processes and their PIDs. Since the list might be very long, a pipe character with the more command can be used.

3. To kill a process by its PID, type the command: taskkill /F /PID pid\_number

4. To kill a process by its name, type the command: taskkill /IM "process name" /F Taskkill supports many useful options which can be used to terminate apps.

Power shell

To kill a process which runs elevated, you need to open Power Shell as Administrator.

1. Open Power Shell. If required, run it as Administrator.

2. Type the command Get-Process to see the list of running processes.

3. To kill a process by its name, execute the following cmdlet: Stop-Process –Name "Process Name" -Force

4. To kill a process by its PID, run the command: Stop-Process -ID PID –Force

Process Communication

The Windows operating system provides mechanisms for facilitating communications and data sharing between applications. Collectively, the activities enabled by these mechanisms are called interposes communications (IPC). Some forms of IPC facilitate the division of labor among several specialized processes. Other forms of IPC facilitate the division of labor among computers on a network. After it is decided whether application would benefit from IPC, it should be decided which of the available IPC methods to use. It is likely that an application will use several IPC mechanisms. The answers to these questions determine whether an application can benefit by using one or more IPC mechanisms. The following IPC mechanisms are supported by Windows:

● Clipboard: A loosely coupled data sharing method. When a user uses copy or cut command in any application/windows, the copied data is saved in clipboard by windows (temporary storage). The other application can access the data from the clipboard.

● COM:Component Object Model offers a platform to interact in Server and Client patterns between processes. COM server can be a local server or In-Process server. There can also be multiple COM clients who interact with the COM server and exchange data.

● Data Copy:Windows provides a message i.e. WM\_COPYDATA which enables a process to share data with another process. It can be used with Send Message API of win32 and

COPYDATASTRUCT is used as a parameter. This message is used in case of local computers.

● DDE:Dynamic Data Exchange is a protocol that contains a set of guidelines and rules to send data across processes. A process can use Send Message API with

WM\_DDE\_INITIATE or WM\_DDE\_ACK message sent in response to WM\_DDE\_INITIATE message. It uses shared memory to exchange data.

● File Mapping: File Mapping, a fast communication mechanism between processes and Gives an efficient way to use the file content in the virtual memory or by accessing the memory sharing. In this IPC the data of the file is treated as a part of the address space of the process so that process can easily access the address of the content. Any other process with access to the shared memory should implement synchronization to mitigate the risks of data getting corrupted. File Mapping is done on the same system/machine and is not available for network processes.

● Mailslots:Mailslot provides only one way communication until users create multiple mailslots. One process can create a mail slot as a server and the other process creates the client mailslot. Mailslot client sends the message to the maillot server by writing a message and the messages are appended to the mailslot server until the server has read them. A process can have both server and client mailslot and this helps in multi directional communication. Mailslot provides the facility to broadcast the message over the network.

● Pipes:Pipes can be used as both single and bi-directional data sharing mechanisms.

Windows supports two types of pipes i.e. pipe and anonymous pipe. Anonymous pipes can be used in the same network or between the related processes only, while named pipes can be used over a network within different processes. Pipe can be considered as a FIFO queue Where one end acts as a server and other as the client.

● RPC:Remote Procedure Call provides a way to communicate over the network so that a process can invoke a function in the other process. RPC maintains a tightly coupled relationship between the client and server with high performance.

● Windows Sockets: Socket is an efficient way to send and receive data over the network and on the local computer. It uses multiple protocols like TCP/IP and UDP. It is used with the combination of machine IP and Port address where the data can be transported.

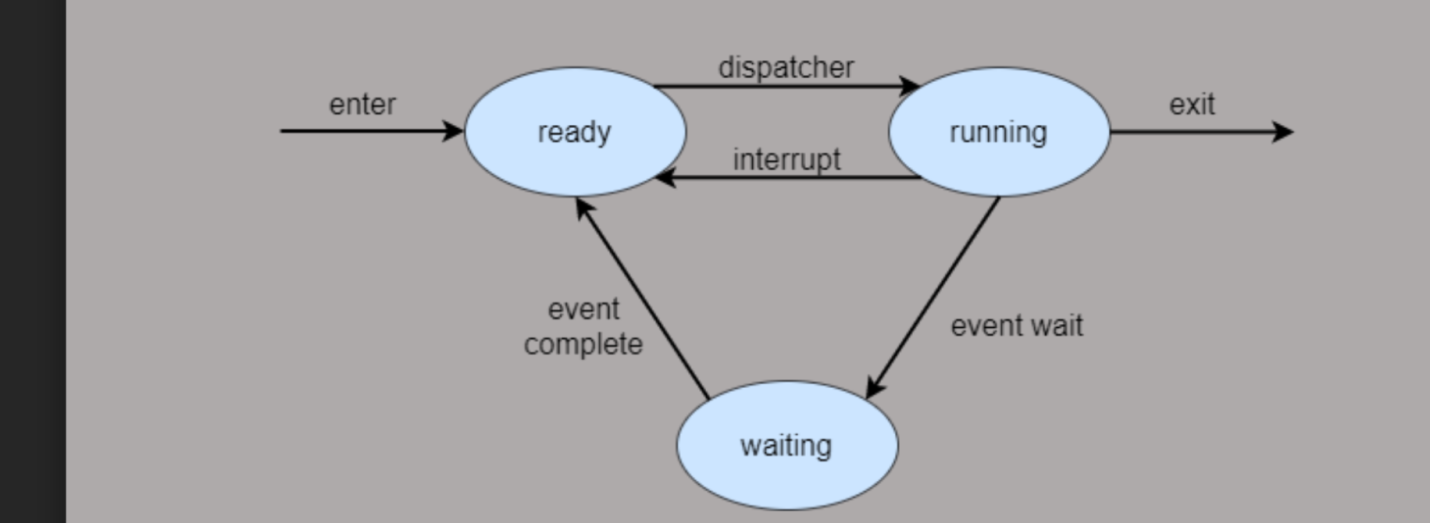
Process State Diagram

Windows applications can exist in three states at the basic level as shown below.

● Ready

● Running

● Waiting



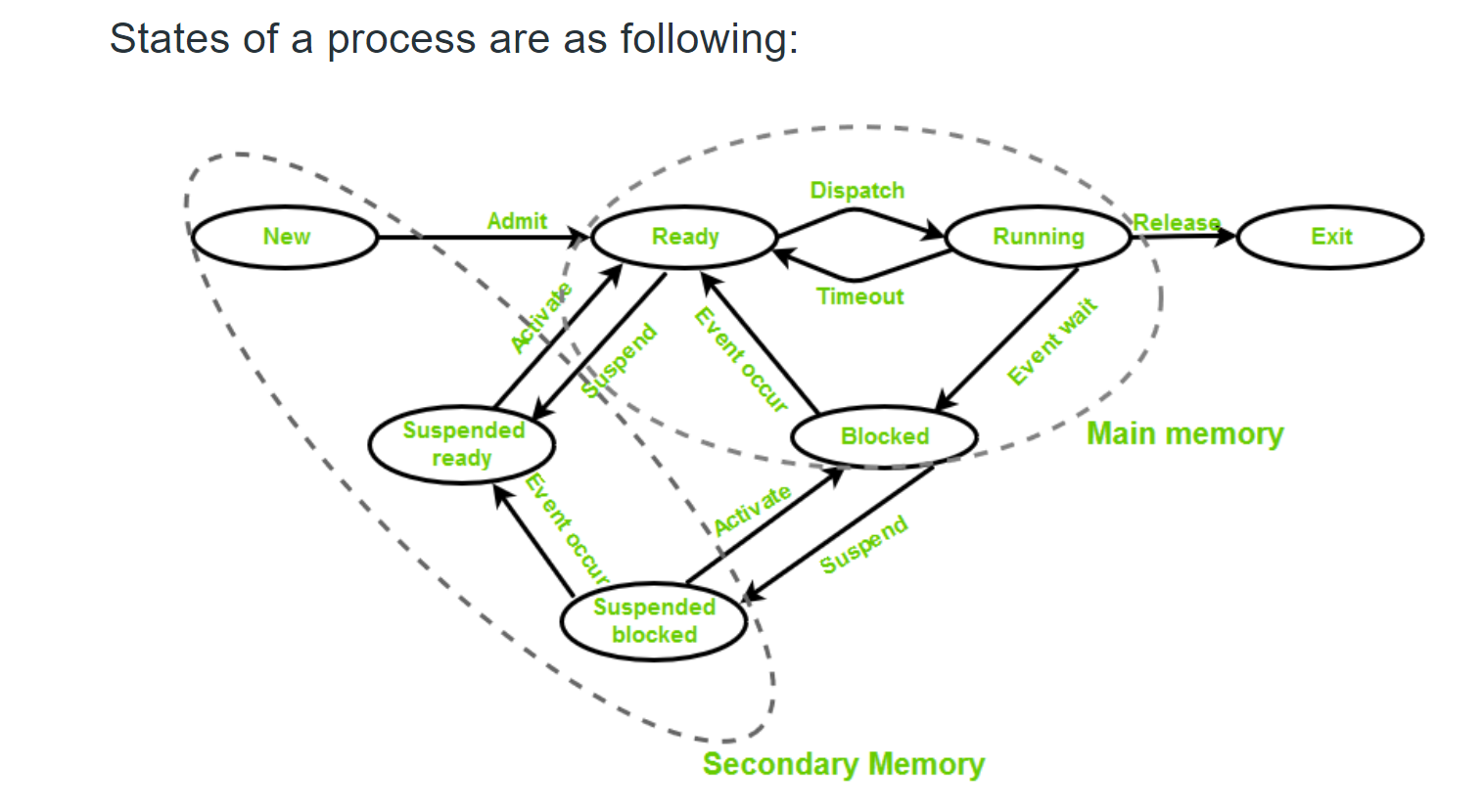
● When a user launches/activates any application, then it goes in the running state.

● Applications can be waited if a user does not use it and it is no longer in the foreground.

● From the Suspended state, applications can get ready for either resume that application or terminate the OS in order to reclaim system resources.

It is important to understand the process st ate transitions in a running application.

When the user first launches the application, the splash screen is shown and then the application starts running.



The process can be explained as follows −

● When the application is suspended, the app gets five seconds to handle that suspended

event.

● When the application is suspended, absolutely no code runs and no resources are allocated.

● When it resumes, the app is notified that it has resumed. If it is coming from a suspended

state, you need to take no action.

● Under memory pressure, it is possible for the application to be terminated.

● Remember that the user will not be notified at that point, and so any saving user does, hasto do when the user enters into the suspended application state.When the application transits back and forth between Running and Suspended states, firesuspending and resuming events respectively.

Process Management System-Calls

System calls in Windows 17 are used for file system control, process control, interprocess communication, main memory management, I/O device handling, security etc. The programs interact with the Windows operating system using the system calls. Since system calls are the only way to access the kernel, all the programs requiring resources must use system calls.

Details about some of the important system calls in Windows are given as follows –

|  |  |
| --- | --- |
| System Call | Description |
| CreateProcess() | A new process is created using this command |
| ExitProcess() | This system call is used to exit a process. |
| CreateFile() | A file is created or opened using this system  call. |
| ReadFile() | Data is read from the file using this system  call. |
| WriteFile() | Data is written into the file using this system  call. |
| CloseHandle() | This system call closes the file currently in  use. |
| SetTimer() | This system call sets the alarm or the timer of  a process |
| CreatePipe() | A pipe is created using this system call |
| SetFileSecurity() | This system call sets the security for a  particular process |
| SetConsoleMode() | This sets the input mode or output mode of the console’s  input buffer or output screen buffer respectively. |
| ReadConsole() | This reads the characters from the console  input buffer. |
| WriteConsole() | This writes the characters into the console  output buffer. |

Process versus Thread

Each process provides the resources needed to execute a program. A process has a virtual address space, executable code, open handles to system objects, a security context, a unique process identifier, environment variables, a priority class, minimum and maximum working set sizes, and at least one thread of execution. Each process is started with a single thread, often called the primary thread, but can create additional threads from any of its threads.

A thread is the entity within a process that can be scheduled for execution. All threads of a process share its virtual address space and system resources. In addition, each thread maintains exception handlers, a scheduling priority, thread local storage, a unique thread identifier, and a set of structures the system will use to save the thread context until it is scheduled. The thread context includes the thread's set of machine registers, the kernel stack, a thread environment block, and a user stack in the address space of the thread's process. Threads can also have their own security context, which can be used for impersonating clients.

Microsoft Windows supports preemptive multitasking, which creates the effect of simultaneous execution of multiple threads from multiple processes. On a multiprocessor computer, the system can simultaneously execute as many threads as there are processors on the computer.

A job object allows groups of processes to be managed as a unit. Job objects are namable, securable, sharable objects that control attributes of the processes associated with them. Operations performed on the job object affect all processes associated with the job object.

An application can use the thread pool to reduce the number of application threads and provide management of the worker threads. Applications can queue work items, associate work with waitable handles, automatically queue based on a timer, and bind with I/O.

User-mode scheduling (UMS) is a lightweight mechanism that applications can use to schedule their own threads. An application can switch between UMS threads in user mode without involving the system scheduler and regain control of the processor if a UMS thread blocks in the kernel.

Each UMS thread has its own thread context instead of sharing the thread context of a single thread. The ability to switch between threads in user mode makes UMS more efficient than thread pools for short-duration work items that require few system calls.

A fiber is a unit of execution that must be manually scheduled by the application. Fibers run in the context of the threads that schedule them. Each thread can schedule multiple fibers. In general, fibers do not provide advantages over a well-designed multithreaded application. However, using fibers can make it easier to port applications that were designed to schedule their own threads.

User level thread Model

The user-level threads are implemented by users and the kernel is not aware of the existence of these threads. It handles them as if they were single-threaded processes. User-level threads are small and much faster than kernel level threads. They are represented by a program counter(PC), stack, registers and a small process control block. Also, there is no kernel involvement in synchronization for user-level threads.

Advantages of User-Level Threads

Some of the advantages of user-level threads are as follows −

● User-level threads are easier and faster to create than kernel-level threads. They can also be more easily managed.

● User-level threads can be run on any operating system.

● There are no kernel mode privileges required for thread switching in user-level threads.

Disadvantages of User-Level Threads

Some of the disadvantages of user-level threads are as follows −

● Multithreaded applications in user-level threads cannot use multiprocessing to their advantage.

● The entire process is blocked if one user-level thread performs blocking operation.

Kernel-Level Threads

Kernel-level threads are handled by the operating system directly and the thread management is done by the kernel. The context information for the process as well as the process threads is all managed by the kernel. Because of this, kernel-level threads are slower than user-level threads.

Advantages of Kernel-Level Threads

Some of the advantages of kernel-level threads are as follows −

● Multiple threads of the same process can be scheduled on different processors in kernel- level threads.

● The kernel routines can also be multithreaded.

● If a kernel-level thread is blocked, another thread of the same process can be scheduled by

the kernel.

Disadvantages of Kernel-Level Threads

Some of the disadvantages of kernel-level threads are as follows −

● A mode switch to kernel mode is required to transfer control from one thread to another in

a process.

● Kernel-level threads are slower to create as well as manage as compared to user-level threads.

Process Scheduling Parameters

Parameters influence the scheduling of a process. Windows 7 scheduling parameters are dependent on the maintenance plan type. Focusing on Time Based Maintenance Plans and Performance Based Maintenance Plans. The parameters are set to optimize the criteria listed below:

● Maximum CPU utilization

● Maximum throughput

● Minimum turnaround time

● Minimum waiting time

● Minimum response time

There are several different criteria to consider when trying selecting the "best" scheduling algorithm for a particular situation and environment, including:

● CPU utilization- Ideally the CPU would be busy 100% of the time, so as to waste

0 CPU cycles. On a real system CPU usage should range from 40% (lightly loaded) to 90% (heavily loaded).

● Throughput - Number of processes completed per unit time. May range from 10

/ second to 1 / hour depending on the specific processes.

● Turnaround time - Time required for a particular process to complete, from submission time to completion.

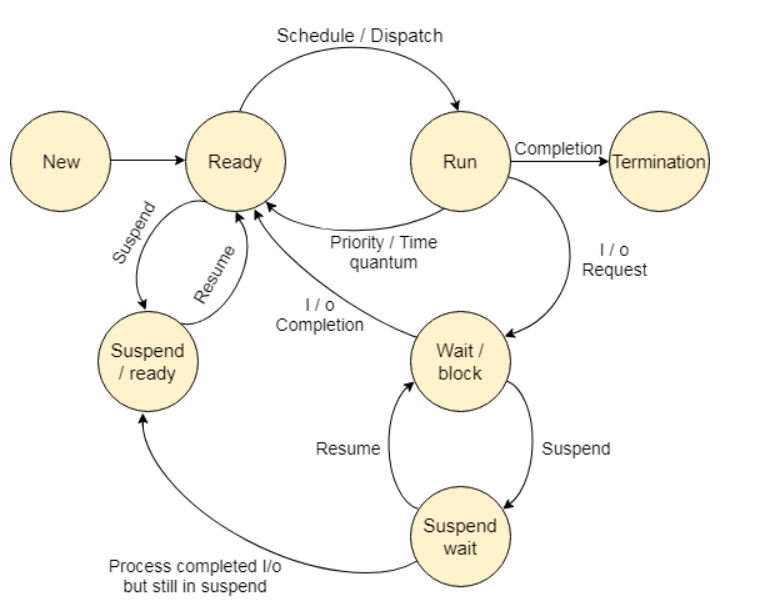
● Waiting time - How much time processes spend in the ready queue waiting their turn to get on the CPU.

● Load average - The average number of processes sitting in the ready queue waiting their turn to get into the CPU. Reported in 1-minute, 5-minute, and 15- minute averages by "uptime" and "who".

● Response time - The time taken in an interactive program from the issuance of a command to the commence of a response to that command.

● In general one wants to optimize the average value of a criteria (Maximize CPU utilization and throughput, and minimize all the others.) However sometimes one wants to do something different, such as to minimize the maximum response time.

● sometimes it is more desirable to minimize the variance of a criteria than the actual value. I.e. users are more accepting of a consistent predictable system than an inconsistent one, even if it is a little bit slower.

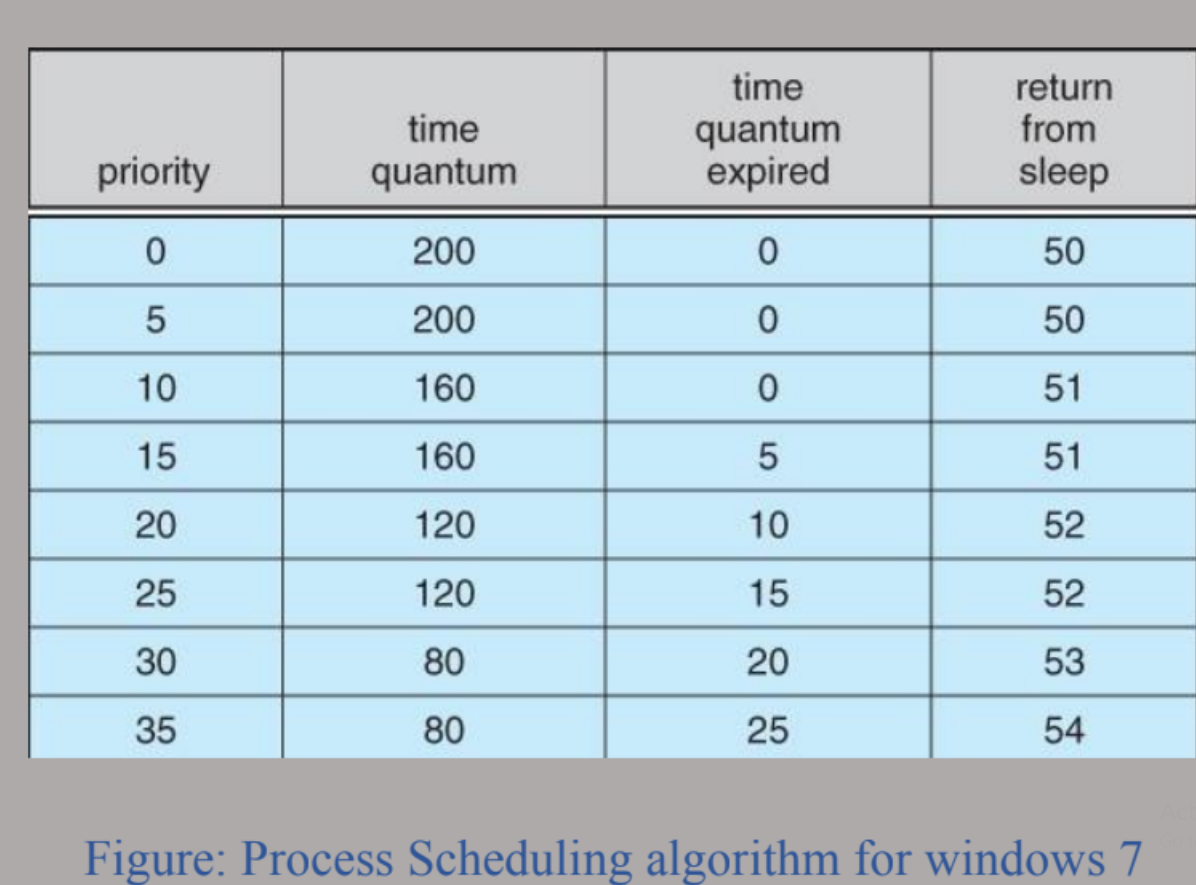


Process Scheduling Algorithm

Windows 10 uses Round Robin and different levels of priority for process scheduling.

Multitasking is controlled by the system using the priority values, as processes compete with each other for processor time.

The priority level ranges from 0 (lowest priority) to 35 (highest priority). The highest priority processes are separated from the lowest priority processes, using queues. Higher priority queues have more time slices than other queues. This allows for more processing time of processes that need to be executed faster.



In round robin, when a process has been executed and the processor is free, Windows looks for the ready queue with the highest priority to be executed next. This is known as Context Switch.

The steps of context switch are given below:

1. Save the context of the process that has just been executed.

2. Place the process executed, at the end of the queue.

3. Find the highest priority queue that contains the ready queue.

4. Remove the process at the head of the queue, load its context, and execute.

Lower priority queues can cease to execute, if a higher priority queue becomes available to run.

These executions are halted with the Suspend Thread or Switch to Thread functions. The scheduler does not allocate any processor time to the suspended or blocked queues, regardless of their priority. When the processes are ready to be executed, it is transferred to the ready queue.

Common reasons for context switching are given below:

● The time slice has expired.

● A queue with a higher priority has become ready to be executed.

● A running thread needs to wait.

Priority Boosts

The scheduler in Windows 7 uses priority values to determine the next executable. Each thread has a dynamic priority and a base priority. Initially all the threads have the same priority value.

The system can boost or lower the dynamic priority value, depending on the circumstances. This allows the system to be responsive and ensures process starvation from processor time.

Reasons for dynamic priority boosts are given below:

● When a process is brought to the foreground.

● When a window receives input, such as keyboard inputs, mouse input, timer messages.

● When a blocked thread’s wait condition is satisfied.

Windows uses multiple threads to access a resource. This is achieved with the help of a synchronization object and the wait functions. Wait functions can be specified to include the handle of the synchronization object. This means that processes can share the same handle, making it possible for inter process synchronization.

There are four exclusive types of synchronization objects:

1. Event: Notifies when an event has occurred.

2. Mutex: Can only be owned by one thread at a time, allowing mutually exclusive access to

a shared resource.

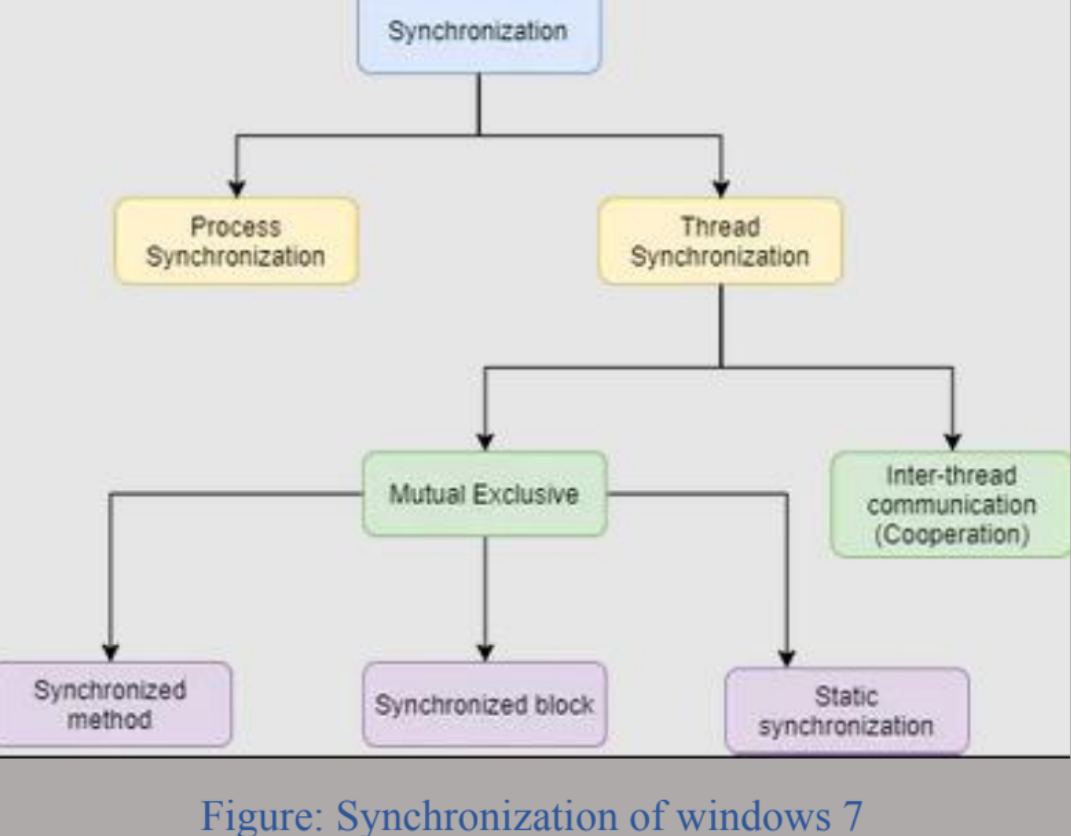
3. Semaphore: Maintains a count, limiting the number of threads able to access a resource simultaneously.

4. Waitable timer: Notifies waiting threads when a specified time has arrived.

Booting process

Besides this, windows 7 has an excellent boot process

• During the BIOS Initialization phase, the platform firmware identifies and initializes hardware devices, and then runs a power-on self-test (POST). The POST process ends when the BIOS detects a valid system disk, reads the master boot record (MBR), and starts



Bootmgr.exe. Bootmgr.exe finds and starts Winload.exe on the Windows boot partition, which begins the OSLoader phase.

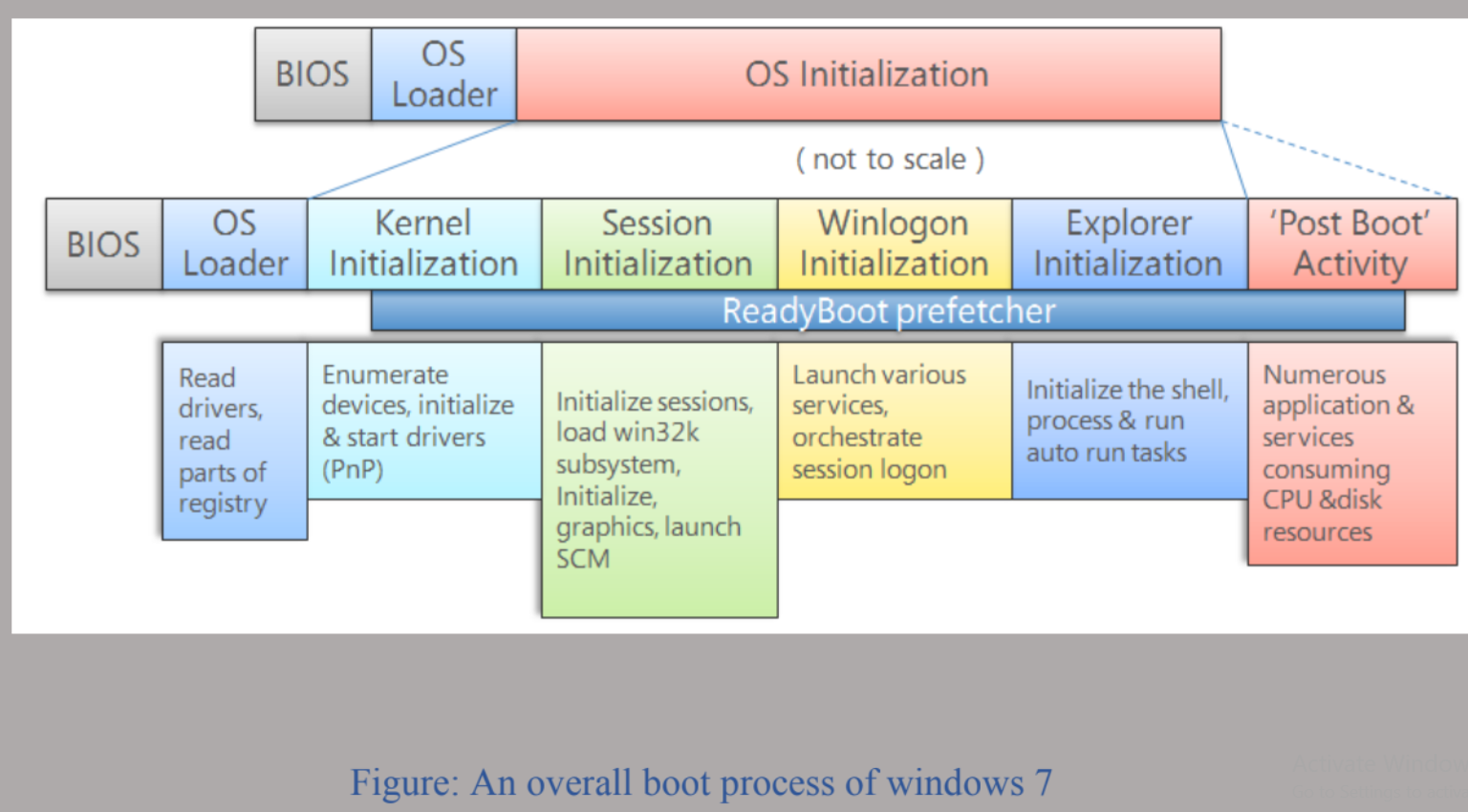
• During the OSLoader phase, the Windows loader binary (Winload.exe) loads essential system drivers that are required to read minimal data from the disk and initializes the system to the point where the Windows kernel can begin execution. When the kernel starts to run, the OSloader loads the system registry hive and additional drivers that are marked as BOOT\_START into memory.

• During the OS Initialization phase, most of the operating system work occurs. This phase involves kernel initialization, Plug and Play activity, service start, logon, and Explorer (desktop) initialization. The OS Initialization can be divided into four sub phases. Each

Sub phase has unique characteristics and performance vulnerabilities

• The Post Boot phase includes all background activity that occurs after the desktop is ready.

The user can interact with the desktop, but the system might still be starting services, tray icons, and application code in the background, potentially having an impact on how the user perceives system responsiveness.



Virtual Memory Management

In Windows 7, the memory manager implements virtual memory to provide services such as mapping files, large memory support, cache memory support and so on.

Virtual Address Space

Each process has its own virtual address space. Threads cannot access memory space of another process, protecting a process from being corrupted by other processes. Virtual address space is an internal data structure, which corresponds to their actual physical address.

The processes in the physical memory have a set of pages in the virtual memory, known as the working set. The pages in the working set are shared between these processes. Removing a page from a working set of a process will not affect other pages. Removing a page from the working set of all processes, the page becomes a transition page. The transition page is then cached in the memory until it is referred to again, or repurposed.

The memory manager creates memory pools that the system uses to allocate memory. There are two types of memory pools, nonage pool and paged pool. These memory pools are mapped into the virtual address space of each process, and are reserved for the system. Kernel objects allocated are used in the nonage pool. Paged pools can be paged in and out of the system, depending on the circumstances. Single processors have 3 paged pools, while multiprocessors have 5 paged pools in Windows 7.

Virtual Memory Functions

Virtual Memory functions allow processes to manipulate pages in the virtual address space.

Windows 7 can perform 8 operations.

1. Reserve a range of a process’s virtual address space.

2. Commit a range of reserved pages.

3. Specify read/write, read only or no access for a range of committed pages.

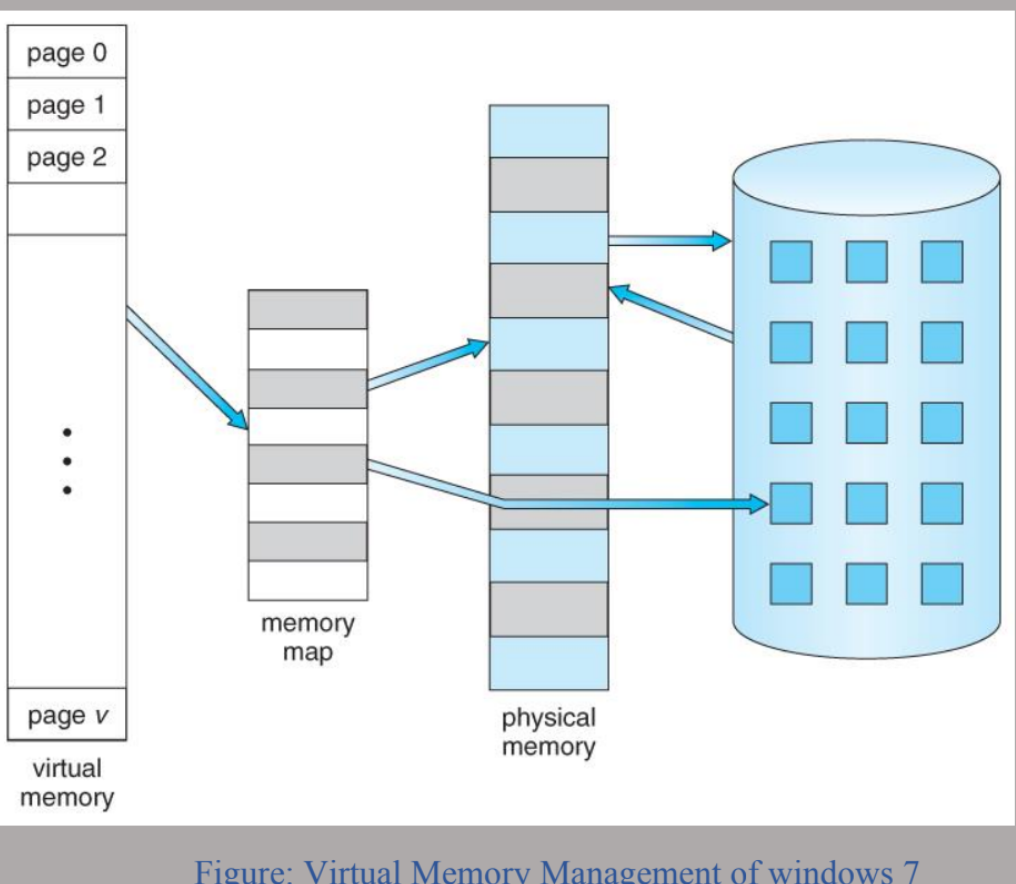
4. Free range of reserved pages.

5. Recommit a range of committed pages.

6. Lock pages of committed pages into physical memory.

7. Obtain information about pages.

8. Change the access protection for a specified range of committed pages.



Dynamic Device Support

1. Early in the history of the PC industry, computer configurations were fairly static, although new devices might occasionally be plugged into the serial, printer, or game ports on the back of a computer. The next steps toward dynamic configuration of PCs are laptop docks and PCMCIAcards. Using such a device, a PC could quickly be connected to or disconnected from a full set of peripherals. Contemporary PCs are designed to enable users to plug and unplug a huge host of peripherals frequently.

2. Support for dynamic configuration of devices is continually evolving in Windows. The system can automatically recognize devices when they are plugged in and can find, install, and load the appropriate drivers often without user intervention. When devices are unplugged, the drivers automatically unload, and system execution continues without disrupting other software. Additionally, Windows Update permits downloading of third- party drivers directly through Microsoft, avoiding the usage of installation DVDs or having the user scour the manufacturer’s website.

3. Beyond peripherals, Windows Server also supports dynamic hot-add and hot-replace of CPUs and RAM, as well as dynamic hot-remove of RAM. These features allow the components to be added, replaced, or removed without system interruption. While of limited use in physical servers, this technology is key to dynamic scalability in cloud computing, especially in Infrastructure As- a-Service (IaaS) and cloud computing environments. In these scenarios, a physical machine can be configured to support a limited number of its processors based on a service fee, which can then be dynamically upgraded, without requiring a reboot, through a compatible hypervisor such as Hyper-V and a simple slider in the owner’s user interface.

Conclusion:

With the implementation of user-friendly operating system, our day-to-day life activities have become much easier. Operating system like Windows 7 plays a great role both in computing and also in monitoring our daily tasks. It’s important for a user to buy the original product instead of piracy to support the developers.