North South University
Department of ECE

Name: Muhammad Zawad Mahmud
ID No: 1931401642

Course: CSE323
Date of Submission: 10.09.2021

Section: 01

Write an individual review paper about any existing operating system. A brief history and design principles of the operating system, present a detailed understanding of the technology used in process and memory management, its file systems, security used etc.

A program that acts as an intermediary between a user of a computer and the computer hardware. Operating system goals: ● Execute user programs and make solving user problems easier ● Make the computer system convenient to use ● Use the computer hardware in an efficient manner. When it comes to best OS to use, Linux, Traditional Windows and MAC OS are great options. Here I'm going to talk about Microsoft Windows which I use for almost 18 hours in a day.

Microsoft Windows Operating System

Abstract: Windows is the most famous and recognizable working framework. From Windows 95, right to the Windows 10, it has been the go-to working programming that is activating the computing systems around the world. It is easy to use, and fires up and continues tasks quick. Windows is working framework from Microsoft Inc. It is a framework program that controls, arranges and controls in general work area distributing tasks. It fills in as an interface between the equipment and other programming on the PC framework. It assigns memory areas to information and other PC programs. It controls other PC assets like documents and projects, input/yield gadgets, PC recollections and so on during information preparing activities.

I. Introduction:

Windows additionally plays out the equipment and programming finding to recognize issues and mistakes. The most recent versions have more implicit security to guard and the information safe.

II. Versions of windows:

Windows provides a graphical user interface (GUI), and support many peripheral devices. In addition to windows operating system for personal computers, Microsoft also offers operating system for servers and personal computers.

1. Windows 1.0- 2.0 (1985-1992). Window 1.0 allowed users to point and click to access the windows. Window 2.0 added icons, keyboard shortcuts and improved graphics.

2. Windows 3.0-3.1 (1990-1994). Support better icons and program manager (1st version of "look and feel" screen of Microsoft windows).

3. Windows 95 (August 1995). It runs faster and has ability to automatically delete and configure installed hardware (plug and play).

4. Windows 98 (June 1998). It offers supports for new technology FAT32, AGP, MMX, USB, DVD. It is an active desktop which integrates the web browser (internet Explorer).

5. Windows ME- Millennium Edition (September 2000). Booting is in Dos option.

6. Windows NT 3.1-4.0 (1993-1996). It supports pre-emptive multi tasking. They are Windows NT and Windows NT SERVER.

7. Windows 2000 or W2k. (February 2000). It is an operating system for business desktop and laptop systems to run software applications.

8. Windows XP (October 2001). It has a better look and feel. There are two versions Home and professional.

9. Windows Vista (November 2006). It offered an advancement in reliability, security and ease of deployment.

10. Windows 7 (October 2009). Improved performance and start-up-time and window media center.

11. Windows 8 (August 2012). It was developed with touch screen use in mind. Better start-up. Start screen replaced look and feel screen made up of "live Tiles"

12. Windows 10 (2015). Fast start-up, Microsoft edge, Microsoft new browser.

13. Window server (2003). Designed for corporate networking, internet/intranet, hosting, data bases and similar functions.

14. Window Home Server (January 2007). This is a "consumer Server" designed to use with multiple computers connected in the home.

OS Name	Computer Architecture Supported	Target System Default	Security Threat	Best For	Price	Website
Windows	X86, x86-64,	Workstation, Personal Computer	Huge	Apps, Gaming, Browsing	$119 - $199	**Windows**

III. Design Goals:

- Windows networking operating systems -Influenced by several operating system models 1. Employed already-existing frameworks. 2.Introduced new features. -Object model: manage and allocate resources. -Symmetric multiprocessing (SMP): achieve maximum multiprocessor performance
- Needs: 1. Accommodate user needs. 2.Optimize resources
- Response where Five design goals: 1. Extensibility. 2.Portability. 3.Reliability. 4.Compatibility. 5.Performance

IV. Design principle of Windows:

Aside from window operating system designed for personal computers (PCs) and laptops, Microsoft has also developed operating system for services, handheld devices and mobile phones.

1. Window CE (November 2006) it is designed for small devices such as PDAs for handheld computing devices. 2. Windows mobile (April 2000) designed for smart phones and mobile devices. 3. Windows phone 7-10 (November 2010) or win phone 7 designed for smart phones and mobile devices but targeted more to the consumer market than enterprises market.

V. Major Features of Windows OS:

All in all, feature wise it stands out in a few places, and lags behind in others.

A. Memory Management:

- Every operating system 1. Has own physical memory view. 2. Makes application programs access memory in specified ways
- Full physical memory 1. Virtual Memory Manager pages some memory contents to disk
- Challenge for all Windows operating systems 1. Run application programs (Windows or POSIX) and Without programs crashing into each other's memory
- Memory layout (recent Windows versions) 1. Operating system: high virtual memory 2. User code and data: low virtual memory
- User process 1. Cannot read or write system memory directly
- Memory paged to disk 1. User-accessible memory 2. System memory segment labeled paged pool
- Memory never paged to disk 1. System memory segment labeled nonpaged pool

User-Mode Feature:

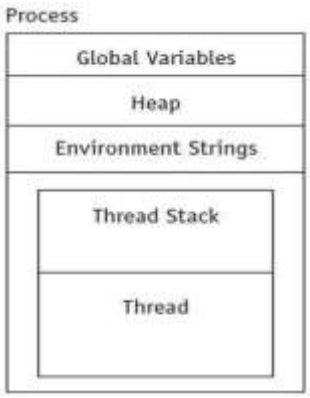
- VM Manager (virtual machine manager) 1. User-mode subsystems share memory efficiently 2. Provides native services: allows process to manage virtual memory -Allocate memory in two stages. -Read and/or write protection for virtual memory. -Lock virtual pages in physical memory. -Retrieve information about virtual pages. -Protect virtual pages. -Rewrite virtual pages to disk

B. Virtual Memory Implementation:

- VM manager reliance 1. Address space management 2. Paging techniques
- Address space management 1. Upper half of virtual address space (Accessible only to kernel- mode processes) 2. Code in lower part (kernel code and data) (Never paged out of memory)
- Paging 1. VM manager part: transfers pages (Between memory page frames and disk storage) 2. Complex combination: -Software policies: when to bring a page into memory and where to put it. -Hardware mechanisms: exact manner VM Manager translates virtual addresses into physical addresses. 3. Pager not portable: modified for each new hardware platform (Windows: small code and well-isolated) 4. Processor chip translates program generated address into a physical address (Page with address not in memory: hardware generates a page fault and calls the pager) 5. Processor uses translation look-aside buffer (TLB) (Speeds memory access)
- Paging policies 1. Dictate how and when paging is done 2. Composition -Fetch policy: determines when pager copies a page from disk to memory. -Placement policy: determines where virtual page is loaded in memory. -Replacement policy: determines which virtual page is removed from memory to make room for a new page

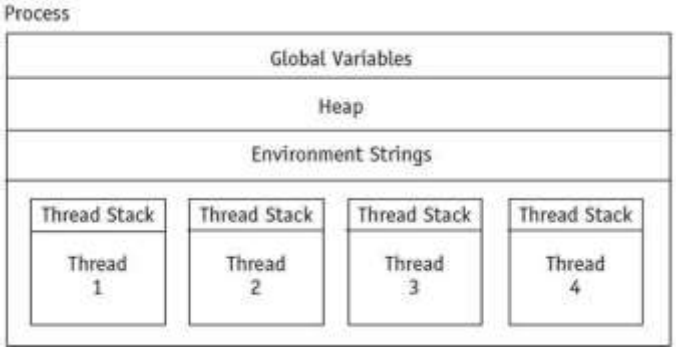
C. Process Management:

- Process: 1. Combination of executable program, private memory area, and operating system allocated resources as the program executes 2. Requires at least one execution thread
- Thread 1. Process entity: roughly equated to a task
- Windows 1. Pre-emptive-multitasking, multithreaded operating system 2. Process contains one thread composed of: -A unique identifier. -Volatile set of registers: contents indicate processor's state. -Two stacks used during thread's execution. -Private storage area: used by subsystems and dynamic-link libraries
- Threads 1. Thread components called thread's context 2. Actual data forming context varies from one processor to another 3. Kernel (Schedules threads for execution on a processor) 4. Thread executes code 5. Overhead incurred by thread is minimal 6. Unit asking (Process with single thread)



Unit asking in Windows. Here's how a process with a single thread is scheduled for execution on a system with a single processor.

- Multithreading 1. Systems with multiple processors. -Process has as many threads as CPUs available. -All threads belonging to one process: share global variables, heap, and environment strings
- Windows operating systems 1. Include some synchronization mechanisms (Give exclusive access to global variables as multithreaded processes execute)



Multitasking using multithreading. Here's how a process with four threads can be scheduled for execution on a system with four processors.

- Multithreading example: modifying a database application 1. Entering records: one thread writes the last record to disk while another thread accepts new data 2. Database searching: several threads search an array simultaneously
- Client/server applications: CPU-intensive for server 1. Client makes query requests; server's processor manages the query 2. Windows handles requests allocating additional CPU resources

D. Multitasking:

In figuring, performing multiple tasks is the simultaneous execution of different assignments (otherwise called measures) over a specific timeframe. New assignments can interfere with as of now began ones preceding, they finish, rather than hanging tight for them to end. Subsequently, a PC executes sections of numerous errands in an interleaved way, while the assignments share regular handling assets, for example, focal preparing units (CPUs) and principle memory. Performing various tasks naturally interferes with the running project, sparing its state (halfway outcomes, memory substance and PC register substance) and stacking the spared condition of another program and moving control to it. This "setting switch" might be started at fixed time stretches (pre-emptive performing multiple tasks), or the running project might be coded to motion toward the administrative programming when it tends to be interfered with (helpful performing multiple tasks).

E. Compatibility:

An incredible inheritance emotionally supportive network with a stunning history for in reverse similarity. Anyone can install any types of software and games into windows. Progressed coordinated systems administration. **Gaming Compatibility** - best in the OS world for windows. **The Windows Registry**-A bound together, incorporated, key-esteem store that permits joining and acknowledgment of framework factors and introduced programming and information base factors. **EDIT** - more features to be added after.

F. Device Management:

- Windows I/O system provides: 1. Multiple installable file systems (FAT, CDFS, and NTFS) 2. Services making device-driver development easy (Workable on multiprocessor systems) 3. Drivers: added to or removed from system dynamically System administrators 4. Fast I/O processing (Drivers written in high-level language) 5. Mapped file I/O capabilities (Image activation, file caching, and application use)
- I/O system 1. Packet driven (I/O request represented by I/O request packet (IRP)) 2. IRP (Data structure controlling how I/O operation processed at each step)
- I/O manager: 1. Creates an IRP representing each I/O operation 2. Passes IRP to appropriate driver 3. Disposes packet when operation complete
- Driver IRP receipt 1. Performs specified operation 2. Passes it back to I/O manager *or 3.* Passes it through I/O manager to another driver for further processing
- I/O manager tasks 1. Supplies code, common to different drivers 2. Manages buffers for I/O requests 3. Provides time-out support for drivers 4. Records installable file systems loaded into operating system 5. Provides flexible I/O facilities (Subsystems (POSIX) implement their respective I/O application programming interfaces) 6. Allows dynamic loading: device drivers and file systems based on users' needs
- Windows I/O services 1. Device-independent model (Multilayered device driver concept)
- Device driver made up of standard set of routines 1. Initialization routine 2. Dispatch routine 3. Start I/O routine 4. Completion routine 5. Unload routine 6. Error logging routine
- I/O manager 1. Determines driver called to process request (Based on file object's name) 2. Driver object -Represents individual driver in system. -Created by I/O manager when driver loaded into system. -May have multiple device objects connected to it. 3. Device object - Physical, logical, or virtual device on the system. -Describes device characteristics

Event	Result
Instruction: READ "MYFILE.TXT"	READ = FUNCTION CODE 1 "MYFILE.TXT" = DISK SECTOR 10
Actions:	1. Access DRIVER OBJECT (1)
	2. Activate READ routine
	3. Access DISK SECTOR 10

Example showing how a device object is created from an instruction to read a file,

- Device objects list 1. Represents physical, logical, virtual devices. 2. Controlled by the driver
- Advantages of using different objects 1. Portability -Frees I/O manager from knowing details about drivers. -Follows pointer to locate driver 2. Easy loading of new drivers 3. Easy assigning drivers to control additional or different devices (System configuration changes)
- I/O manager knows nothing about file system
- Overhead 1. I/O manager passes information requests back and forth 2. Uses single-layer device driver approach (Simple devices (serial and parallel printer ports) 3. Uses multilayered approach (More complicated devices (hard drives))
- Asynchronous I/O operations 1. Nearly all low-level operations

G. File Management:

- Windows current versions 1. Designed to be independent of file system on which they operate
- Virtual file 1. Primary file handling concept (current Windows versions) 2. Programs perform I/O on virtual files (File handles manipulate them) 3. Executive file object representing all I/O sources and destinations
- Processes call native file object services to read from or write to file
- I/O manager directs virtual file requests 1. Real files, file directories, physical devices, or other system-supported destinations
- File objects 1. Hierarchical names 2. Protected by object-based security 3. Support synchronization 4. Handled by object services
- Opening file 1. Process supplies file's name and type of access required

- File objects bridge gap 1. Between physical devices' characteristics and directory structures, file system structures, and data formats
- File object 1. Memory-based representation of shareable physical resources (Contains only data unique to an object handle)
- File Contains data to be shared 1. New file object created with new set of handle-specific attributes 2. Each time process opens a handle

Attribute	Purpose
Filename	Identifies the physical file to which the file object refers
Device type	Indicates the type of device on which the file resides
Byte offset	Identifies the current location in the file (valid only for synchronous I/O)
Share mode	Indicates whether other callers can open the file for read, write, or delete operations while this caller is using it
Open mode	Indicates whether I/O is synchronous or asynchronous, cached or not cached, sequential or random, and so on
File disposition	Indicates whether to delete the file after closing it

- File management system 1. Supports long filenames (Include spaces and special character) 2. Automatically shortens filenames when required

H. Network Management:

- Networking 1. Integral part of Windows operating system executive 2. Provides services: user accounts and resource security 3. Implements communication between computers 4. Named pipes: provide high-level interface for passing data between two processes (regardless of locations) 5. Mail slots: provide one-to-many and many-to-one communication mechanisms
- Redirector: Network can incorporate multiple redirectors 1. Coded in C programming language 2. Implemented as loadable file system driver 3. Not dependent on system's hardware architecture 4. Function (Direct I/O request from user or application to remote server with appropriate file or resource)
- Windows Server operating systems 1. Written in C (Complete compatibility with existing MS-NET and LAN Manager SMB protocols) 2. Implemented as loadable file system drivers 3. No dependency on hardware architecture (Where operating system running)

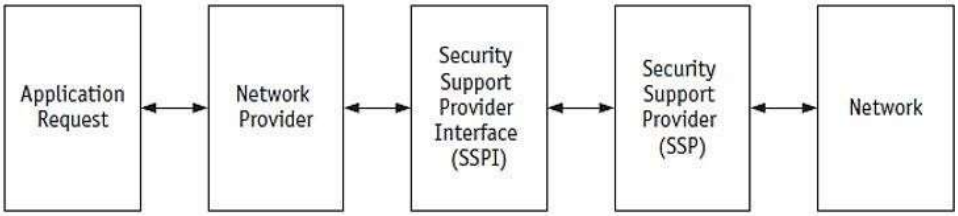
I. Safety and Security Management:

- Windows network operating systems 1. Provide object-based security model 2. Security object (Represents any resource in system (file, device, process, program, or user)) 3. Allows administrators to give precise security access (Monitor and record how objects used)
- Windows biggest concern 1. Aggressive patch management needed (Combat many viruses and worms)
- U.S. Department of Defense (Identified and categorized seven levels of security features)
- Class C2 security level compliance 1. Features in Windows -A secure logon facility. - Discretionary access control. -Auditing ability. -Memory protection
- Multilayered security system 1. First security layer: password management 2. Second security layer: file access security -Distinguishes between owners and groups. -Users decide operation types person can perform on a file. -Gives user auditing capabilities: automatically tracks who uses files and how files used

J. Security Terminology:

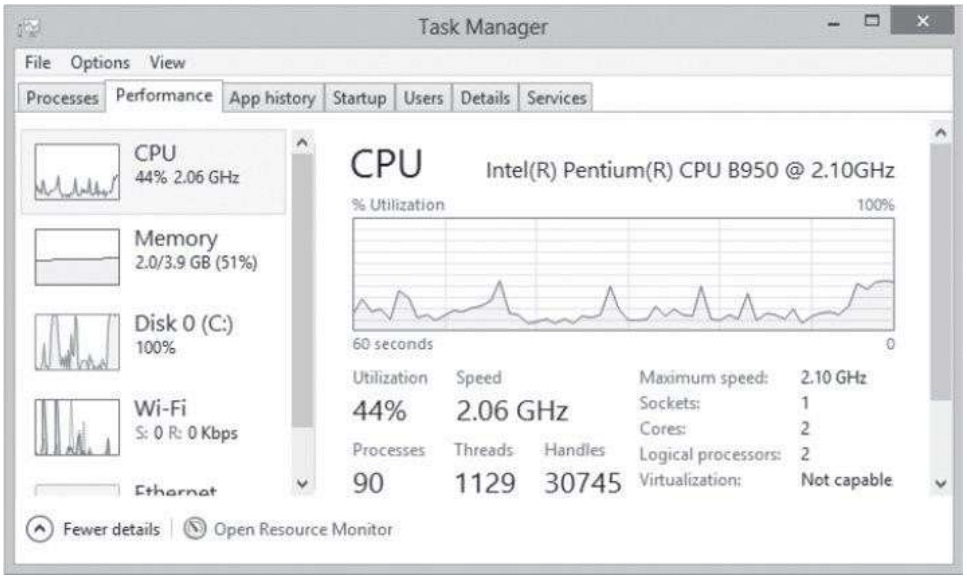
- Built-in security 1. Necessary element for managers of Web servers and networks 2. Requires authentication mechanism allowing client to prove identity to server. -Client supplies authorization information. -Server uses information: determines specific access rights given to client 3. Provides data integrity using various methods

- Windows operating systems feature Kerberos security
- Kerberos security 1. Authentication (mutual), data integrity, and data privacy 2. Each domain has own Kerberos server 3. Microsoft implemented standard Kerberos protocol 4. Microsoft separates distributed security services users from their providers (Supports many options without unusable complexity)



Requests from an application flow through these security stops on the way to and from the network.

K. Task Manager:



Using the Task Manager, users can view the system status, assign priorities, and more.

VI. Conclusion:

Last but not the least Windows Started as a microcomputer operating system and now includes complex multiplatform software also Run computing systems of all sizes. Regardless of whether these weaknesses are because of their huge portion of the market (making them gigantically alluring) or coding blunders with respect to Microsoft, the outcome is the equivalent. There is a steady requirement for each framework overseer and PC owner to proactively keep all Windows frameworks as secure as conceivable through cautious access control and fix the executives. So it is a powerful operating system and got Significant market force.

References:

- https://www.researchgate.net/publication/317182647_MICROSOFT_WINDOWS_OPERATING_SYSTEM
- https://www.computerhope.com/issues/ch001967.htm
- https://www.softwaretestinghelp.com/best-operating-systems/
- http://cybercomputing.blogspot.com/2014/08/Features-of-Windows-OS-and-Advantages-and-Disadvantages-of-Windows-OS.html
- https://slideplayer.com/slide/6199323/
- https://quizlet.com/209985329/itec-274-final-exam-flash-cards/
- https://www.scribd.com/presentation/212101397/Chapter-15