1. Windows

Windows Architecture

- Kernel: It's the heart of the operating system that manages critical functions such as memory management, process scheduling, and device access. Unlike other systems, the Windows kernel is hybrid, combining characteristics of a monolithic kernel and a microkernel, to improve efficiency and modularity. This hybrid design balances performance and flexibility.
- HAL (Hardware Abstraction Layer): It's the hardware abstraction layer. This layer allows the kernel to operate independently of the underlying hardware. In other words, it isolates hardware peculiarities, enabling more consistent and efficient software development and maintenance.
- WIN32 and UWP: It's the Universal Windows Platform. Specifically, they are user-level subsystems that provide APIs for applications to interact with hardware through the kernel. There are differences between these two:
  - Win32: It's the set of APIs that are part of the user environment in Windows. It does not interact directly with the hardware, but rather makes calls to the kernel through the Windows API.
  - UWP (Universal Windows Platform): It's a more modern development platform, designed to create universal applications that work on multiple devices, such as PCs, tablets, Xbox, and IoT devices. These applications run in a sandbox environment, which improves security and portability. In itself, it is not a subsystem, but a development framework.
- Windows Manager: Window managers control the visual presentation of the graphical interface, including window management, user interaction, and graphic effects.
- Network subsystem: This manages all network communications, including internet access and local network connectivity. It includes support for protocols such as TCP/IP and enables communication between applications and local or remote networks.
- Device Drivers: They provide the necessary interface for the kernel and hardware to interact efficiently. Windows Driver Model (WDM) and the Windows Driver Framework (WDF) are the modern architectures for driver development in Windows.
- Windows Registry: It's a centralized database that stores configurations and options, where users can customize the system. It stores configurations for both the system and applications and drivers.
- Windows Services: It's the part of the software that Windows uses to perform specific tasks at the operating system and application level, usually transparently to the user. Service

Control Manager (SCM) is the component responsible for managing the lifecycle of these services.

- CMD: It's a terminal emulator that allows interaction with the operating system through commands. It's a legacy component. Currently, PowerShell is the more modern and powerful command-line interface in Windows.
- Security Subsystem: Here, the system can manage access control and auditing. Its main purpose is to maintain the integrity and security of the operating system and user data. Key components include: Local Security Authority Subsystem Service (LSASS), Security Account Manager (SAM), and Active Directory (in domain environments). In addition to them, it is responsible for managing security policies, authentication, and authorization.

Software Types

In Windows, there are 3 types of licenses:

- OEM (Original Equipment Manufacturer): These types of licenses usually come pre-installed on new devices and are linked to specific hardware. They are not transferable from one computer to another. Technical support in this case is provided by the manufacturer, not Microsoft.
- RETAIL (Full Packaged Product - FPP): These licenses can be transferred between devices, as long as the license is deactivated from the previous device before being activated on the new device. They are purchased independently.
- VOLUME (Volume Licensing): Volume licenses are aimed at companies, organizations, or educational environments, as they allow activation on multiple devices within a corporate network.

Differences Between PowerShell and CMD

In both, you can execute code and give commands to the system to do what you want, but there are differences between them.

- CMD or Command Prompt: It is based on the MS-DOS command-line interface. It is an interface that over time has not had visual changes and is not as comfortable to use today. Although it is a useful tool for basic tasks such as file navigation, batch script execution, and system command execution, it is quite limited compared to PowerShell.
- PowerShell: It is an object-oriented Shell with a complete scripting language. Its integration with the .NET Framework (in Windows PowerShell) and .NET Core/5+ (in PowerShell Core and PowerShell 7+) allows you to manipulate not only text, but complete objects with properties and methods, which makes it extremely powerful for system administration and automation. It is a complete environment that combines an interactive shell and a robust programming language.

The most notable differences are:

- **Power and flexibility**: PowerShell allows you to work with objects, which facilitates the manipulation of complex data without the need to parse text as in CMD.
- **.NET Based**: This gives you access to .NET libraries and classes, greatly expanding scripting and administration capabilities.
- **Advanced capabilities**: PowerShell supports cmdlets, scripts, functions, modules, WMI (Windows Management Instrumentation) access, REST APIs, and remote management tools like WinRM.

Commands vs. Cmdlets

- **Commands**: They are external programs or executables independent of the shell environment, which perform specific tasks. Common examples include `ping`, `ipconfig`, and batch scripts (`.bat`, `.cmd`). These commands run in both CMD and PowerShell because PowerShell has compatibility with external Windows commands.
- **Cmdlets**: They are internal commands exclusive to PowerShell, designed as instances of .NET classes. They operate with objects instead of text, which allows direct access to properties and methods without the need for parsing. Cmdlets follow the Verb-Noun convention (e.g., `Get-Process`, `Stop-Service`, `New-Item`).

Important Directories in Windows

C:
├── Windows │
├── System32 - Critical system files and drivers.
│ ├── SysWOW64 - Compatibility for 32-bit applications on 64-bit systems.
│ ├── Temp - System temporary files.
│ ├── Prefetch - Prefetch files to optimize loading times.
│ ├── SoftwareDistribution
│ │ └── Download - Temporary files downloaded by Windows Update.
│ ├── Tasks - System scheduled task files.
│ ├── Logs - System log files.
│ └── Resources - Themes and visual elements of the operating system.
├── Boot - Files related to the boot process.
├── Program Files - Installed 64-bit applications.
├── Program Files (x86) - Installed 32-bit applications on 64-bit systems.
├── ProgramData - Shared application configurations and data.
├── Users
│ ├── user_name
│ │ ├── AppData

```
│ │ │ ├── Local - User-specific local configurations.
│ │ │ ├── LocalLow - Local configurations with lower privilege level.
│ │ │ └── Roaming - Synchronizable configurations between devices.
│ │ ├── Documents - User's personal documents.
│ │ ├── Desktop - Items on the user's desktop.
│ │ └── Downloads - Files downloaded by the user.
│ └── Default - Default profile for new users.
├── Recovery - Tools and data for system recovery.
├── System Volume Information - Volume metadata and restore points.
├── hiberfil.sys - Hidden file for system hibernation.
├── pagefile.sys - Paging file for virtual memory.
├── Swapfile.sys - Swap file for modern applications.
└── Others
├── Windows.old - Remains of previous Windows versions after an update.
└── Temporary files - Depending on specific configurations.
```

Important Files

- Ntldr and bootmgr: Ntldr is present in versions prior to Windows Vista, after which it was replaced by bootmgr, which is currently used. They help load the operating system when the PC is turned on, more specifically, they are boot managers.
- Ntoskrnl: It is the Windows kernel and is responsible for memory management, process management, and hardware interaction. It is fundamental for system operation and provides essential services such as communication between hardware and applications.
- Winload: Responsible for loading Windows during the startup process.
- Explorer: Windows Explorer process, responsible for providing the graphical user interface, including: desktop, taskbar, start menu, etc. Consequently, it consumes many hardware resources.
- Svchost: Hosts Windows services, as a result, several services can share a single svchost process, this to reduce resource consumption.
- C:\Windows\System32\Config: Windows registry database files are stored, which contain crucial configurations for the operating system, applications, and hardware.
- C:\Windows\System32\Drivers: Contains device drivers that allow interaction between the operating system and hardware. Drivers are programs that allow the operating system to recognize and communicate with devices.
- Pagefile: It is a file used for virtual memory. When physical RAM is full, Windows uses `pagefile.sys` as additional temporary storage space, allowing applications to continue running even when there is not enough RAM available.
- Hiberfil: This file is used when the system enters hibernation. It stores the contents of RAM on the hard drive, which allows, when the computer is turned on again, the system to

recover exactly in the same state as when it was turned off.

- C:\Windows\System32\services.exe: It is the process responsible for managing the execution of system services, such as network services, drivers, and other processes fundamental for Windows operation.
- C:\Windows\System32\drivers\etc\hosts: The `hosts` file allows manual assignment of domain names to IP addresses. It is a static way to manage addresses and is used by Windows before performing DNS queries.
- C:\Windows\System32\userinit.exe: It is responsible for initializing the user environment when Windows starts. After user authentication, this process configures the desktop environment, network settings, and other aspects of the user session.
- C:\Windows\System32\lsass.exe: The `lsass.exe` (Local Security Authority Subsystem Service) manages the system's security policy, including user authentication checks, password security, and implementation of security policies.
- C:\Windows\System32\smss.exe: Responsible for starting a user session and managing the subsystem environment.
- C:\Windows\memory.dmp: This file is generated when the system experiences a "blue screen" (BSOD - Blue Screen of Death). The memory dump file contains a copy of the RAM contents at the time of the error, which allows analyzing the cause of the failure.

Introduction to permissions

In Windows, permissions define what type of access is granted to a user or group on an object, such as a file or folder. These permissions can be assigned to groups, users, or other objects with security identifiers in the domain; at the business level. And at the home use level, they can be assigned to both local groups and users (who are on the computer where the same object resides).
There are 5 types of permissions:

- Read: Allows viewing files and folders.
- Write: Allows modification of files or folders.
- Execute: Allows executing executable files.
- Modify: It is a combination of the permissions of: read, write, execute and the ability to delete files and folders.
- Full control: Grant all of the above permissions, plus the ability to change permissions and take ownership of these files and folders.

Important Concepts

- Ownership: Each object in Windows has an owner, usually the user who created that object. Therefore, the owner has the ability to change the permissions of that object, regardless of

the established permissions.

- Inheritance: It is a mechanism that allows objects within a folder to automatically inherit the permissions of that folder. Only permissions that are marked to be inherited are effectively inherited. Inheritance can be disabled, allowing specific permissions to be set for certain objects without being affected by inherited permissions.
- User rights: In Windows, they are specific privileges granted to user accounts or groups, which allow specific actions to be performed on the system. *Permissions are at the object level and rights at the action level.*
- Integrity labels: They are part of the Windows security model and are used to classify processes and objects according to their level of trust. Its main function is to prevent lower integrity objects from interacting with or affecting higher integrity processes or objects. It has 4 levels: System, High, Medium, and Low.
- Access Control Lists (ACLs): They are more commonly used at the business level. Their operation is based on an Access Control Entry (ACE), which defines permissions for users and groups on a system object. It allows centralizing all permissions management in one place. There are 2 main types:
    - DACL (Discretionary Access Control List): It is the list that defines which users and groups have access and what type of access they have to objects.
        - SACL (System Access Control List): It is used for auditing, it can be configured to record object access events and generate audit entries when the object is accessed or modified.

Attributes

Attributes in Windows are markers that define special properties and behaviors in files and folders, which prevent a file from being accidentally modified. The attributes are:

- Read-Only: Prevents a file from being accidentally modified, since if the file has this attribute, it can only be read, not modified. This attribute does not prevent the file from being deleted, it only prevents its modification.
- Hidden: If a file has this attribute, it will not appear in normal searches. Hidden files can be seen by enabling the "Show hidden files" option in the folder options of Windows Explorer.
- System: Critical assets for Windows operation are identified, generally they should not be modified by users.
- Archive: It is automatically marked by Windows when a file is modified, indicating that it needs to be backed up.
- Directory: It is specific to folders and indicates that the object is a directory.
- Temporary: It is used by files that are used temporarily and that can be deleted after use. It is not manually applied Windows assigns this attribute automatically to the folders.
- Offline: Indicates that the file data is not available immediately.}

Attributes can be modified with the Attrib command

Wildcards

Wildcards are special characters used to represent one or more characters in a text string. They are widely used in file searches or to perform operations with commands. The following are the most common wildcards:

- Asterisk (*): Represents any number of characters, including zero.
- Question mark (?): Represents a single character.
- Square brackets ([ ]): Matches any single character within the brackets.
- Exclamation mark (!): Excludes the characters included between the square brackets.
- Hyphen (-): Matches any range of characters. Remember that you must specify the characters in ascending order.
- Hashtags (#): Matches any numerical character.

Variables

Environment variables are a set of dynamic values within an operating system. In Windows, they are used to determine specific information about the operating system environment. These variables include information such as file paths, directory names, configuration data, and other information that allows the system to work more optimally. There are 2 types of variables:

- User variables: These are specific to each user on the system and store configurations that only affect the current user's environment.
- System variables: These affect all users on the machine, as they are global and are mainly used to configure operating system and installed software information.

Examples of environment variables:

- PATH: Stores the paths of executable files.
- TEMP and TMP: Their value is the location where temporary files are stored.
- USERPROFILE: Its value is the path of the current user's profile.
- SYSTEMROOT: Its value is the Windows installation directory.
- COMSPEC: Its value is the location of the Windows interpreter.
- HOMEPATH: It is the relative part of the user's profile, such as `\Users\Username`, but without including the drive (`C:`).
- PROGRAMFILES: Its value is the installation directory for programs.
- WINDIR: Its value is the directory where Windows is installed.
- APPDATA: Its value is the location for application data for the current user.
- LOCALAPPDATA: Used for storing specific application data.

- PUBLIC: Its value is the path of the directory used for shared files between users.
- COMPUTERNAME: Stores the computer name assigned in the system configuration.
- USERNAME: Stores the current user's name.
- USERDOMAIN: Stores the domain where the current user is located.
- ALLUSERSPROFILE: Location of the common profile for all users.
- NUMBER_OF_PROCESSORS: Number of available CPU cores.
- PROCESSOR_ARCHITECTURE: Processor architecture.

To get the output of any of these environment variables, you can do it in the following ways:

For CMD:

- To see a specific variable (`echo %<variable_name>%`).
- To see all environment variables (`set`).
  For PowerShell:
- To see a specific variable (`echo $env:<variable_name>`).
- To see all environment variables (`Get-ChildItem Env:`).

Redirections in PowerShell

Redirections allow you to manipulate the information returned by a command or cmdlet, to send it to another command, store it, or manage it in different ways.
Redirection operators:

- (>): Redirects standard output to a file, overwriting the content if the file already exists.
- (>>): Redirects all standard output by default, but does not overwrite the information, only appends it.
- (2>): Redirects error output to a file, overwriting it.
- (2>>): Redirects error output to a file, appending it.

Windows Commands for CMD

Variable Management

- set: Displays, creates, or modifies temporary environment variables.
- setx: Creates or modifies permanent environment variables.
- echo: Displays the value of a variable.

Navigation, File, and Directory Management

- dir: Lists the contents of a directory.
- cd: Changes the current directory.

    - Double dot (..): Moves up one level in the directory tree.
- md or mkdir: Creates a directory.
- rd or rmdir: Removes an empty directory.
- del: Deletes a file.
- copy: Copies files to another directory.
- move: Moves files or renames them.
- ren: Renames a file.
- attrib: Changes file attributes.

Help and Support

- help: Displays the list of available commands.
- help [command]: Displays help about a specific command.
- [command] /?: Alternative to get help about a command.

User Management

(Requires CMD as administrator)

- net user: Lists system users.
- net user [user]: Displays user information.
    - /add: Creates a new user.
    - /delete: Deletes a user.

Group Management

(Requires CMD as administrator)

- net localgroup: Lists local groups.
- net localgroup [group]: Displays group members.
    - /add: Adds a user to a group.
    - /delete: Removes a user from a group.

Process Management

- tasklist: Lists running processes.
- taskkill /IM [process_name] /F: Terminates a process by name.
- taskkill /PID [process_ID] /F: Terminates a process by its ID.

Network Management

- ipconfig: Displays network configuration.

    - /all: Detailed network information.
    - /release: Releases the current IP address.
    - /renew: Requests a new IP address.
- ping [address]: Verifies connectivity with an IP or domain.
- tracert [address]: Displays the route to a server.
- netstat: Displays active network connections.
- nslookup [domain]: Resolves a domain name to an IP.
- arp -a: Displays the ARP cache (IP addresses ↔ MAC addresses).
- route print: Displays the routing table.

Windows Commands for PowerShell

Variable Management

- $env:[variable_name]: Accesses environment variables.
- $Variable = "Value": Creates or modifies PowerShell variables (only in the current session).
- Remove-Variable: Deletes a variable.
- Get-Variable: Lists PowerShell variables.

Navigation, File, and Directory Management

- Get-Location: Displays the current location.
- Set-Location: Changes the current directory (equivalent to `cd`).
- New-Item: Creates files or folders.
- Remove-Item: Deletes files or folders.
- Get-ChildItem: Lists the contents of a directory (equivalent to `dir`).
- Copy-Item: Copies files or folders.
- Move-Item: Moves files or folders.
- Rename-Item: Renames files or folders.

Help and Support

- Get-Help: Displays help about a command.
- Get-Command: Lists all available commands in PowerShell.
- Get-Alias: Displays command aliases (like abbreviations).

User Management

(Requires administrator permissions)

- Get-LocalUser: Lists local users.

- New-LocalUser: Creates a new user.
- Remove-LocalUser: Deletes a user.
- Set-LocalUser: Modifies user properties.

Group Management

(Requires administrator permissions)

- Get-LocalGroup: Lists local groups.
- New-LocalGroup: Creates a local group.
- Remove-LocalGroup: Deletes a local group.
- Add-LocalGroupMember: Adds a user to a group.
- Remove-LocalGroupMember: Removes a user from a group.

Process Management

- Get-Process: Lists running processes.
- Stop-Process: Terminates a process by name or ID.
- Start-Process: Starts a new process.

Network Management

- Get-NetIPAddress: Displays configured IP addresses.
- Test-Connection: Verifies connectivity (equivalent to `ping`).
- Get-NetAdapter: Lists network adapters.
- Get-DnsClient: Displays DNS configuration.
- Get-NetRoute: Displays the routing table.

Aliases in PowerShell

An alias is a keyword or alternative name that simplifies the execution of commands or cmdlets in PowerShell, allowing the use of custom shortcuts. Aliases are temporary and only exist during the current session, unless they are saved in the PowerShell profile.

Alias Management Cmdlets

- Get-Alias: Displays the aliases defined for commands in PS.
- New-Alias: Allows you to create new aliases in the current session.
- Remove-Alias: Deletes existing aliases in the current session.
- Export-Alias: Exports the aliases of the current session to a file.
- Import-Alias: Imports aliases from a file to the current session.

Introduction to Security in Windows

- Core isolation: It is a Windows security feature that uses hardware-based virtualization to create a secure and isolated environment where critical operating system processes run. Its goal is to protect the Windows kernel against advanced attacks such as kernel-mode malware or malicious code injection techniques.
- Memory integrity: It is an extension of core isolation, also known as Hypervisor-Protected Code Integrity (HVCI). This function ensures that only digitally signed and trusted drivers and binaries can interact with the isolated kernel.
- DEP (Data Execution Prevention): It is a Windows security function that prevents the execution of code from specific areas of memory marked only for data storage. In this way, it prevents attacks such as buffer overflows or shellcode execution.
- Security policies: They are a set of rules and configurations that control the security and behavior of the system. They include:
  - Account policies: Password rules, account lockout, etc.
  - User rights assignment: Determines what actions a user or group can perform (for example, log in locally or remotely).
  - Local security options: UAC (User Account Control), Firewall, encryption, etc. settings.

Audit Policies:Audit policies are a subcategory of security policies, focused on the logging and monitoring of security-related events. They allow auditing actions such as:

- Access to system resources.
- Use of privileges.
- Changes in security configuration.
- Failed or successful login attempts.

Event Viewer: It is an integrated tool in Windows that logs all system actions and events, grouped into categories such as errors, warnings, operational and security information. It is essential for:

- Security auditing: Tracking access attempts, policy changes, etc.
- Problem diagnosis: Identifying software and hardware errors.
- Forensic analysis: Collecting evidence by reviewing historical logs.

Group Policies: They are centralized configurations that manage the behavior and security policies on computers connected to an Active Directory domain. They allow applying user and system configurations to multiple computers, so they are mainly used in business environments.
BitLocker: It is a full disk encryption tool integrated into Windows. It uses the AES (Advanced Encryption Standard) algorithm with key lengths of 128 bits or 256 bits. BitLocker encrypts the entire storage volume and protects data in case of loss or theft of the device.

Windows Defender: It is the integrated security solution of Windows. It provides real-time protection against threats such as viruses, malware, ransomware, spyware and web-based attacks. Its main advantage is being fully integrated with the operating system, offering optimized performance without the need for additional software.