



UT 14. WINDOWS

Computer Systems
CFGS DAW

Aarón Martín Bermejo
a.martinbermejo@edu.gva.es

2022/2023

Version:230405.1032


License



Attribution - NonCommercial - ShareAlike (by-nc-sa): No commercial use of the original work or any derivative works is permitted, distribution of which must be under a license equal to that governing the original work.

Nomenclature

Throughout this unit different symbols will be used to distinguish important elements within the content. These symbols are:

 Important

 Attention

 Interesting

TABLE OF CONTENTS

1. INTRODUCTION.....	4
2. FILE MANAGEMENT.....	5
2.1 Filesystems.....	5
2.2 File attributes.....	6
3. USERS, GROUPS, PERMISSIONS.....	8
3.1 Users.....	8
3.2 Groups.....	9
3.3 Permissions.....	9
4. WINDOWS COMMAND LINE.....	11
5. RESOURCES.....	13

UT 14. WINDOWS

1. INTRODUCTION

Windows is a privative operating system developed by Microsoft around the mid 1980s widely used in personal computers (PCs). Most recent version right now it's Windows 11 released in October 2021.

Its more acknowledgeable feature is its user-friendly interface which includes a Graphical User Interface (GUI) that allows users to interact with the system through icons, menus, **windows** and visual elements. Although like almost nothing in computers the windows GUI-based for operating systems was already developed, over the years it was Microsoft Windows the one extremely succeeding in the PC world.

In windows there are different “families” that target different sectors of the computers market:

- Windows NT family for consumers: windows XP, Vista, Windows 7/8/10/11
- Windows Server for servers: latest version is Windows Server 2022 and its main competitor is Linux as a whole but specifically certain distributions like RedHat.
- Windows IoT, for embedded devices.

There have been other families that nowadays are no longer under development like:

- Windows 9x: targeted for consumers, was the base of Windows operating systems until Windows NT achieved stability and performance. Windows 95 or Windows 98 are based on this family.
- Windows Mobile and Windows Phone: targeted for smartphones, there were many versions of Windows Mobile and Windows Phone specially after buying Nokia. Although, the expansion in the smartphone segment was not very good and Microsoft finally discontinued that department.

2. FILE MANAGEMENT

2.1 Filesystems

Microsoft developed several filesystems for their different versions of windows. Most relevant ones are:

1. **FAT filesystems:** FAT or File Allocation Table is a filesystem that was the default one for MS-DOS and Windows 9x family. It was developed originally to be used in floppy disks and adapted to work on hard drives. With the increasing capacity FAT12, FAT16 and FAT32 were developed to adapt the filesystem to it.

It works using an index table (the file allocation table) where each entry of a linked list indicates which cluster/s (a contiguous region in the disk storage) are used by each file. Each entry in this list is limited by a fixed number of bits, based on the FAT version: 8, 12, 16 or 32 bits. This limitation on the entry list also limitates the number of clusters each file can use being calculated that number as the product of the largest number that can be stored in each entry per the size of the cluster.

Boot sector	File allocation table 1	File allocation table 2 (duplicate)	Root directory	Other directories and all files
-------------	-------------------------	-------------------------------------	----------------	---------------------------------

FAT file system volume organization¹.

Even though FAT is pretty old, it's still being used in many systems, specially in embedded systems or in solid-state memory cards because of its simplicity in the implementation.

2. **NTFS:** NTFS or New Technology Filesystem is a proprietary journaling filesystem developed by Microsoft. It became the default filesystem on Windows after FAT and has many interesting features, like:
 - a) **Journaling:** it means that any change it's made by the system it's written into a journal or log **before** the actual change is written in the disk. That way you can reverse or recover changes in case failure happen or even recover the files up to a certain stable point.
 - b) **Shadow copies:** NTFS allows to make back ups of file under usage, by making shadow copies of them based on the journal.
 - c) **ACLs or Access Control Lists:** administrators can define access control lists on the filesystem level for certain files/directories.

¹ <https://social.technet.microsoft.com/wiki/contents/articles/6771.the-fat-file-system.aspx>

- d) File-level encryption: each file can be encrypted individually, instead of the whole drive. This will be of extremely important for tools like Bitlocker.
- e) Transparent compression: files can be stored compressed and be used without any extra action of the user to decompress them. Although, performance can be hit because compression/decompression steps need to be performed to write and read files.

Windows needs to be installed on an NTFS filesystem in order to work.

Besides from the filesystems, Windows has certain specific features in the file management:

- It has storage units and each one usually corresponds to a partition of a device and, therefore, to a filesystem. Each storage unit has a name ranging from A to Z and the main one storing the operating system usually it's called C:
- It has an inverted tree structure to organize the files, since there's a root folder where all other files and folders will hang which is the storage unit. Although, as there are multiple storage units, there's not a single root folder.
- Instead of the regular slash used in UNIX filesystem to navigate the files, in windows it's used the backslash to do it: "C:\users\myuser\myfiles"
- Usually the operating system files are stored into C:\Windows and, specially, in C:\Windows\system32
- Usually, all the program files are installed in "C:\Windows\Program Files" for the 64 bit programs and in "C:\Program Files(x86)" for the 32 bit programs.

⚡ There's a huge difference on capacities on architectures based on x86 (32 bits) and x64 (64 bits), starting by the available memory that can be assigned to programs targeted for one architecture or the other.

<https://phoenixnap.com/kb/x64-vs-x86>

2.2 File attributes

Windows files and folders have attributes that can be checked or not (not confuse attributes with permissions). The attributes are:

- Common attributes
 - **Hidden:** the file won't be listed unless the user has the "show hidden files" option selected
 - **Read only:** it's used to avoid modification or deletion of a file but works only as a warning, files can be written anyways.

- Advanced attributes
 - **Indexed:** it states whether the indexation should take this file or not to be used in searching².
 - **Compress:** the file will automatically be compressed, based on the NTFS filesystem transparent compression feature.
 - **Cypher:** the file will be individually cyphered, based on NTFS file-level encryption. Both compress and cyphering cannot be checked at the same time since both processes imply similar processes to be made upon the file but with different purposes.

² <https://support.microsoft.com/es-es/windows/indexaci%C3%B3n-de-b%C3%BAsquedas-en-windows-10-preguntas-frecuentes-da061c83-af6b-095c-0f7a-4dfecda4d15a>

3. USERS, GROUPS, PERMISSIONS

The concept of users, groups and permissions are pretty common in the software world and specifically in the operating systems and they share the same background concepts, the implementation of them and the features associated can differ a lot.

3.1 Users

In windows, there are two classifications of types of user accounts:

- **Local accounts:** a local account is an account created in the computer that it's being accessed and it will work offline. There are different types of local accounts:
 - **Guest account:** is a special account to let people without user to use the computer. It has limited access to the system, like for installing new software or writing files. By default, it is disabled. You have to enable it if your system needs it (for example, a computer in public library to find books).
 - **Standard accounts:** they are limited accounts, and they only can do basic actions (execute programs without administrator rights, modify their own files, etc).
 - **Administrator accounts:** this accounts have administrator rights. They can do several actions like install programs, install hardware, create/delete/modify user accounts, etc.
- **Domain accounts:** when you need to manage users over different machines or over large networks, domain accounts are the ones that target this kind of requirements.

Domain accounts make use of Windows domains, which are a specific type of network where user accounts and all the resources of the network are managed by a domain controller which is the one that authenticates and gives access to the network resources (like printers, storage, computers, etc).

Active Directory is the directory service that manages Windows domains. Is composed of many different services but most important one is AD DS or Active Directory Domain Services which is the one responsible of authenticating, managing access to the resources and storing all that information.

Although managing the access with Active Directory can be lead to a high level of specification based on the organization requirements, usually the types of accounts managed rely on the types of local accounts available.

- **Microsoft accounts:** MSA or Microsoft Accounts is a single sign-on (SSO) to allow user to authenticate in multiple Microsoft products like Outlook, Office 365 or even Windows. Over the last few years Microsoft has been encouraging this way of authentication in Windows for regular users over the local accounts.

3.2 Groups

Windows user groups are collections of user accounts that are used to manage the access to resources in windows, either locally or in a Windows network. There are two types of windows user groups:

- **Built-in:** these are the groups that come along with the operating system and cannot be modified. Example of this kind of groups are "Administrators", "Users", "Guests" and "Power users" which have pre-defined access permissions to the computer.
- **Custom groups:** are groups created by users and/or administrator from the ground.

When a user is added to a group, the permissions and privileges set up for that group are automatically inherited by the user.

Although usually groups are usually relative to permissions to access resources, in windows you can also specify certain policies or settings for groups like the password complexity policy or the account lockout policy.

3.3 Permissions

In Windows there are many different permission types. There are 6 standard permissions for files and folders:

- **List folder contents:** allows the user to view the file names and subfolder names, navigate to subfolders. view folders but does not permit access to the folder's files.
- **Full Control:** grants the user complete control over a file or folder, including the ability to modify, delete, and change permissions.
- **Modify:** allows the user to modify files or folders, but not delete them.
- **Read and Execute:** allows the user to view files and execute programs, but not modify or delete them.
- **Read:** allows the user to view files, but not modify or delete them.
- **Write:** allows the user to create and modify files, but not delete them.

As you can see, they are not as clear as in Linux and they can be pretty subtle like the difference between write and modify permissions, which relies on the creation of files.

On top of that, these are the simplest permissions, since there's a whole list of advanced permissions like the next ones:

1. Take Ownership: allows a user to take ownership of a file or folder, which can be useful if the current owner is no longer available or has changed.
2. Traverse Folder/Execute File: allows a user to navigate through a folder to reach files and folders within it, even if they do not have permissions to view the contents of those files and folders.
3. List Folder/Read Data: allows a user to view the contents of a folder and read the data in files within that folder.
4. Read attributes: allows user to read the file attributes
5. Write attributes: allows user to modify the file attributes
6. Change permissions: allows a user to change the permission on a file or folder.

⚡ The list of advanced permissions may differ under different windows versions

4. WINDOWS COMMAND LINE

In windows there are two command lines: **cmd** and **powershell**. Both are shells that allow the user to communicate with the operating system or an application through the usage of commands instead of a graphical interface. Due to the high importance and weight of the graphical interface on windows, the shells don't have such a high usage as in the UNIX universe. Even so, they exist and they are extremely important specially for IT management.

CMD³ or command shell was the first shell built into Windows to execute commands. You can either execute standalone commands or with batch files with the extension ".bat".

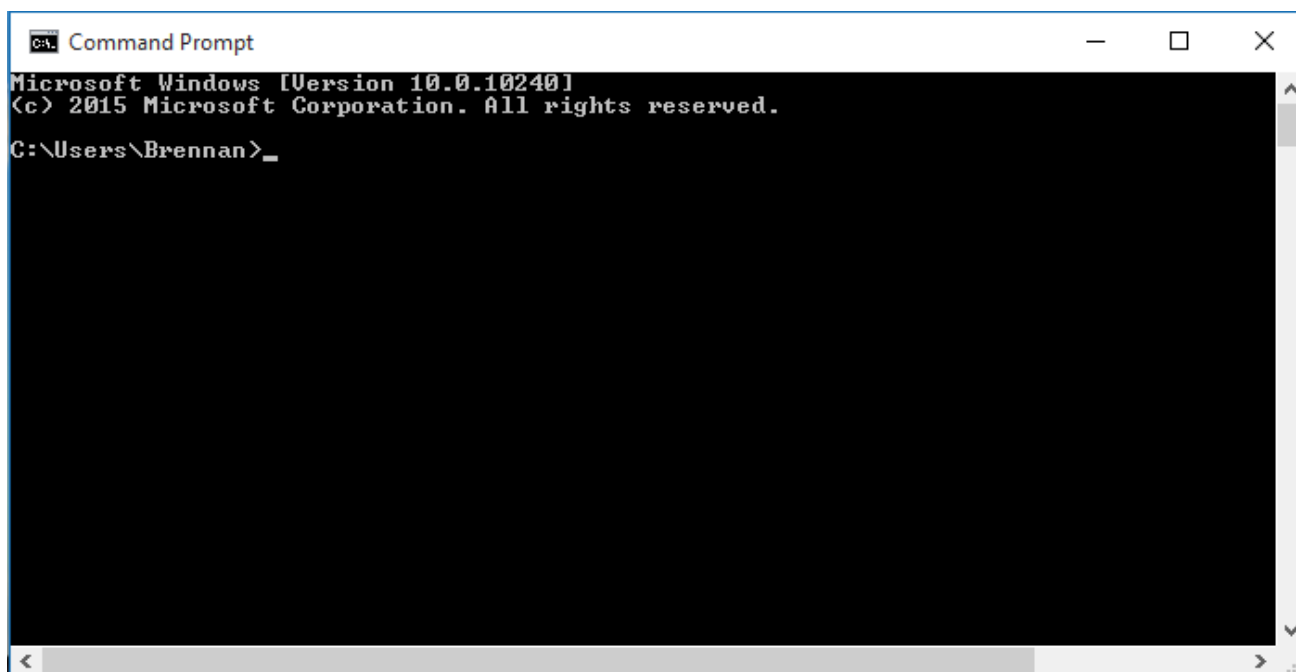



Figure: CMD

There's also an engine to execute scripts inside the CMD called Windows Script Host which can execute scripts based on Javascript and VisualBasicScript. That can be performed by using wscript or cscript.

 To execute commands in Windows Script Host you need to use cscript or wscript

<https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/cscript>

³ <https://learn.microsoft.com/es-es/windows-server/administration/windows-commands/cmd>

<https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/wscript>

After CMD, Powershell was developed and designed to be a more powerful, flexible and robust shell in order to be able to do more advanced tasks, specially in scripting. To do so, Powershell executes cmdlets which are similar to Windows Commands that were executed by CMD but they use a different and more extensible scripting language. Also, Powershell can run any Windows Command run by CMD but not in the opposite way.

Here there are some example commands in CMD and Powershell:

```
# This is a comment in PowerShell

Write-Host "Hello, world!" # This prints "Hello, world!" to the console in
PowerShell

echo "Hello, world!" // This prints "Hello, world!" to the console in CMD

$number = 42 # This assigns the value 42 to a variable named $number in
PowerShell

set number=42 // This assigns the value 42 to an environment variable named
"number" in CMD

Get-ChildItem C:\ # This lists the contents of the C:\ directory in
PowerShell

dir C:\ # This does the same thing in CMD, but the output formatting is
different

ls C:\ # This is a shorter alias for Get-ChildItem that also works in
PowerShell

cd C:\Users # This changes the current directory to C:\Users in PowerShell

cd C:\Users // This does the same thing in CMD

Get-Process | Where-Object {$_.Name -eq "chrome"} # This lists all processes
named "chrome" in PowerShell

tasklist | find "chrome" // This does the same thing in CMD
```

As you can see, the syntax and the “commands” being executed differ. These differences are specially important for advanced scripting.

5. RESOURCES

1. https://en.wikipedia.org/wiki/Design_of_the_FAT_file_system