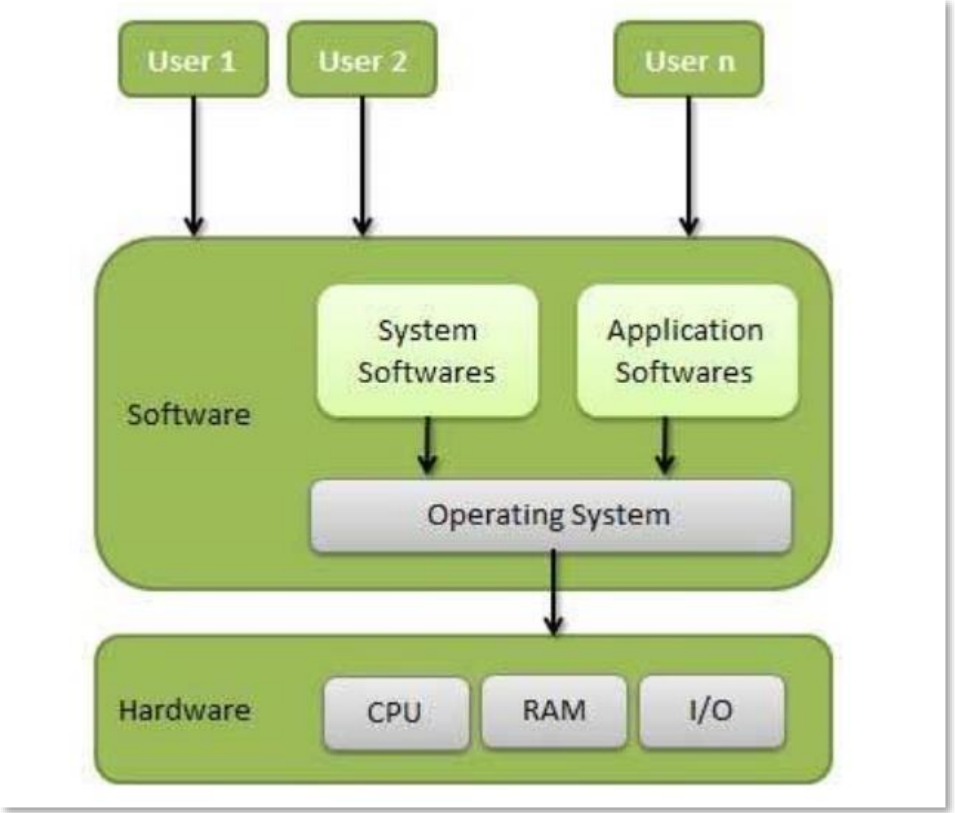
UT2: OPERATING SYSTEMS. ELEMENTS, STRUCTURE AND GENERAL FUNCTIONS

TABLE OF CONTENTS

1. OPERATING SYSTEM DEFINITION

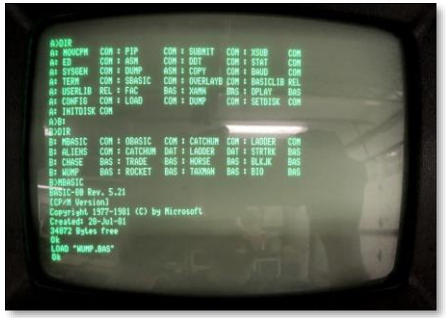
An **operating system** is a program that **acts as an interface between the user and the computer hardware** and **controls the execution of all kinds of programs**.



How does the user communicate with the hardware?

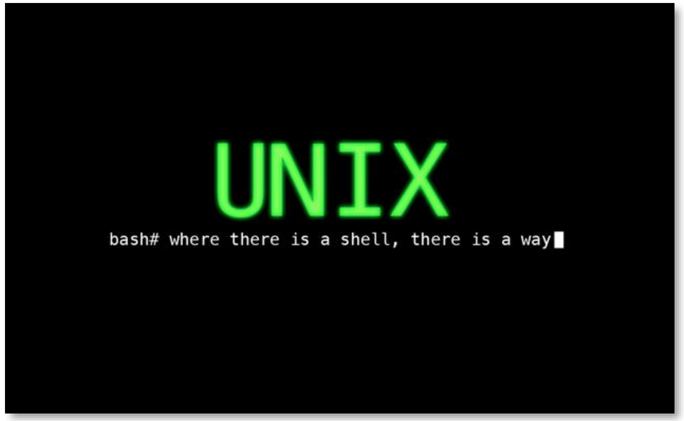
A **user interface (UI)** refers to the part of an operating system, program, or device that allows a user to enter and receive information.

- **TEXT USER INTERFACE:** In this case, the commands are entered as text (e.g., “cat story.txt”).



```
A>DIR
A: MOVCPM   COM : PIP      COM : SUBMIT   COM : XSUB     COM
A: ED       COM : ASM      COM : DDT      COM : STAT     COM
A: SYSGEN   COM : DUMP     ASM : COPY     COM : BAUD     COM
A: TERM     COM : SBASIC   COM : OVERLAYB COM : BASICLIB REL
A: USERLIB  REL : FAC      BAS : XAMN     BAS : DPLAY    BAS
A: CONFIG   COM : LOAD     COM : DUMP     COM : SETDISK  COM
A: INITDISK COM
A>B:
B>DIR
B: MBASIC   COM : OBASIC   COM : CATCHUM  COM : LADDER   COM
B: ALIENS   COM : CATCHUM  DAT : LADDER   DAT : STRTRK   BAS
B: CHASE    BAS : TRADE    BAS : HORSE    BAS : BLKJK    BAS
B: WUMP     BAS : ROCKET   BAS : TAXMAN   BAS : BIO      BAS
B>MBASIC
BASIC-80 Rev. 5.21
[CP/M Version]
Copyright 1977-1981 (C) by Microsoft
Created: 28-Jul-81
34872 Bytes free
Ok
LOAD "WUMP.BAS"
Ok
```

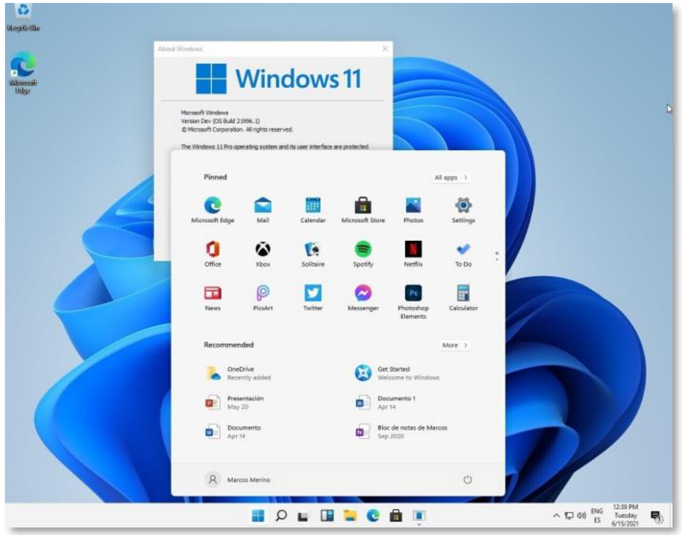
```
Starting MS-DOS...

C:\>_
```



```
bash# where there is a shell, there is a way█
```

To display the text-based Command Prompt in Windows, open the **Start** menu and type **cmd**. Press **Enter** on the keyboard to launch the command prompt in a separate window. With the command prompt, you can type your commands from the keyboard instead of using the mouse.

- **GRAPHICAL USER INTERFACE:** In most operating systems, the primary user interface is graphical, i.e. instead of typing the commands you manipulate various graphical objects (such as icons) with a pointing device.



Microsoft Windows
Version Dev (OS Build 21996.1)
© Microsoft Corporation. All rights reserved.

The Windows 11 Pro operating system and its user interface are protected

2. HISTORICAL BACKGROUND OF OPERATING SYSTEM

- Operating Systems have evolved over the years. So, their evolution through the years can be mapped using generations of computers.
- There are five generations of operating systems. These can be described as follows.
  - **First Generation of Computer (1940 – 1959):** Vacuum tubes and boards
  - **Second Generation of Computer (1960 – 1965):** transistors
  - **Third Generation of Computer (1966 – 1971):** integrated circuits
  - **Fourth Generation of Computer (1971 – 1981)**
  - **Fifth Generation of Computer (1981 – Current)**

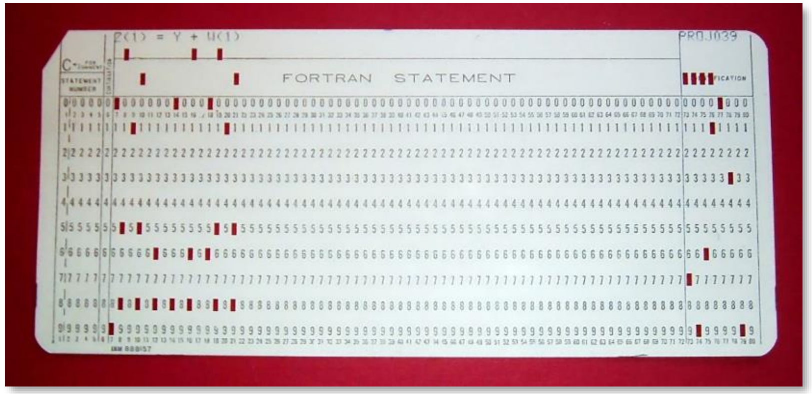
First Generation of Computer (1940 – 1959): Vacuum tubes and boards

- Machines like ENIAC, MARK I → had **no** operating system.

- When UNIVAC I appears → the **concept of an operating system was born** → **EXEC I**

Second Generation of Computer (1960 – 1965): transistors

- **Type of operating systems: Batch operating systems** (sistemas operativos por lotes). They executed the jobs one by one in batch. When one job from the batch executed, then the second job has taken from it and so on.
- Programs were on plugboards *(tableros de conexiones)*, later, punch cards *(tarjetas perforadas)*.



- Plugboard

GE EXEC II

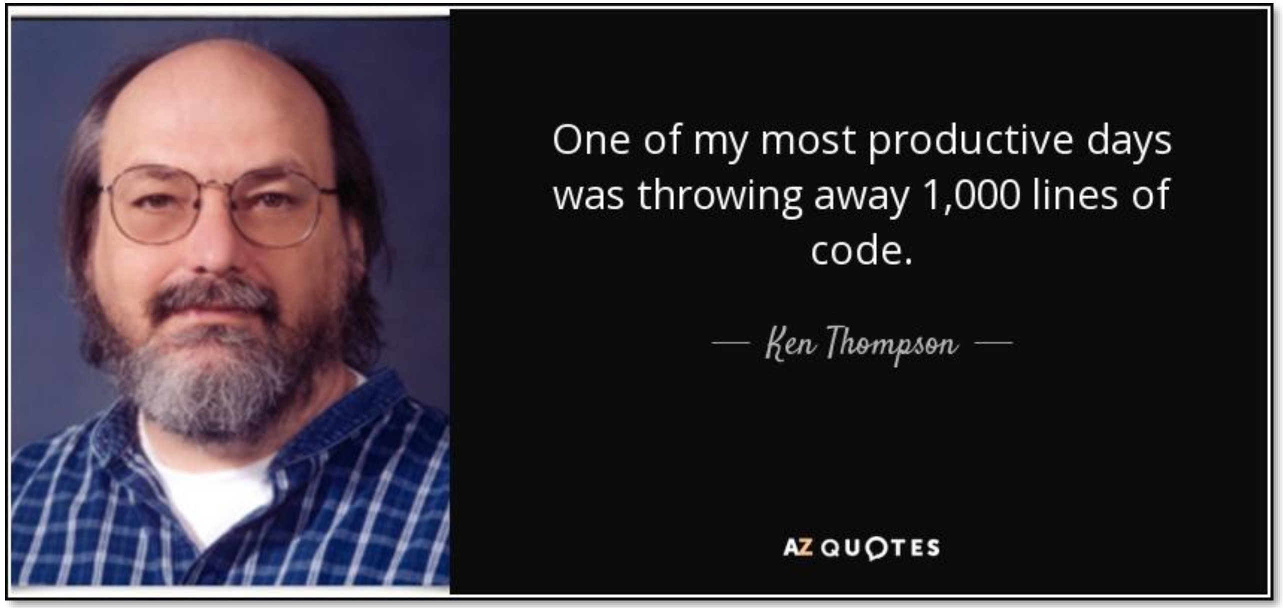
- Whith UNIVAC 7 appears → **EXEC II**
- **Other type of operating systems: Time-Sharing Operating Systems** (sistemas operativos de tiempo compartido). Each task is given some time to execute so that all the tasks work. Each user gets the time of CPU. These systems are also known as Multitasking Systems. → **CTSS**
- The Compatible Time-Sharing System (CTSS) (1962) was designed at MIT as an experimental time-sharing system.

IBM 7094 at MIT

Third Generation of Computer (1966 – 1971): integrated circuits

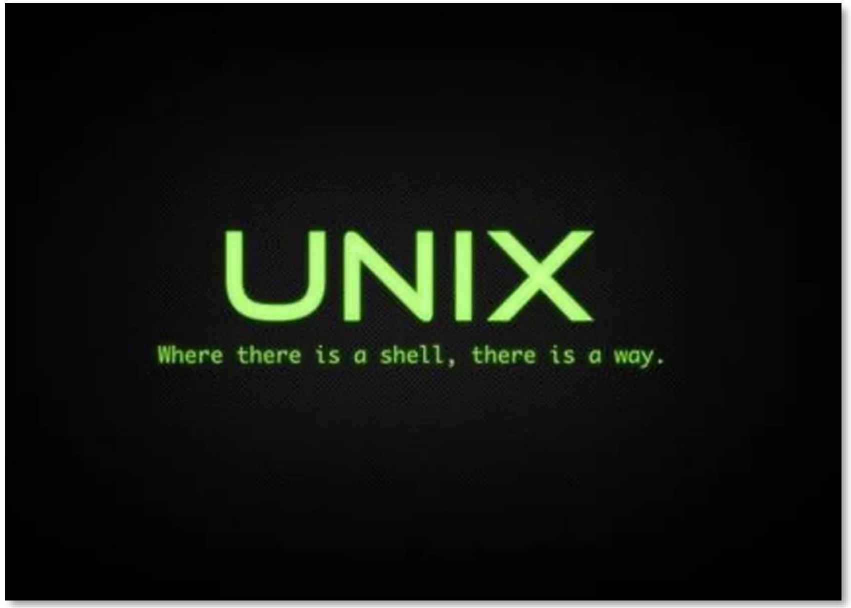
- In the early 1960s, each computer had its own operating system, which made them incompatible with each other, even for computers from the same company.
- IBM attempted to solve both problems with the introduction of the System / 360, a family of computers. The whole family shared the same operating system → **OS / 360**
- After the success of CTSS, MIT, Bell Labs and General Electric decided to undertake *(emprender)* the development of a "computer service", a machine that would serve hundreds of timeshare users. This system was called **MULTICS** (MULTiplexed Information and Computing System).

- Despite the limited commercial success of MULTICS, some of its characteristics influenced.
- Ken Thompson found a small PDP-7 minicomputer that no one was using and started writing a simple version of MULTICS for a single user. From this came → the **UNIX** operating system.



One of my most productive days was throwing away 1,000 lines of code.

— Ken Thompson —



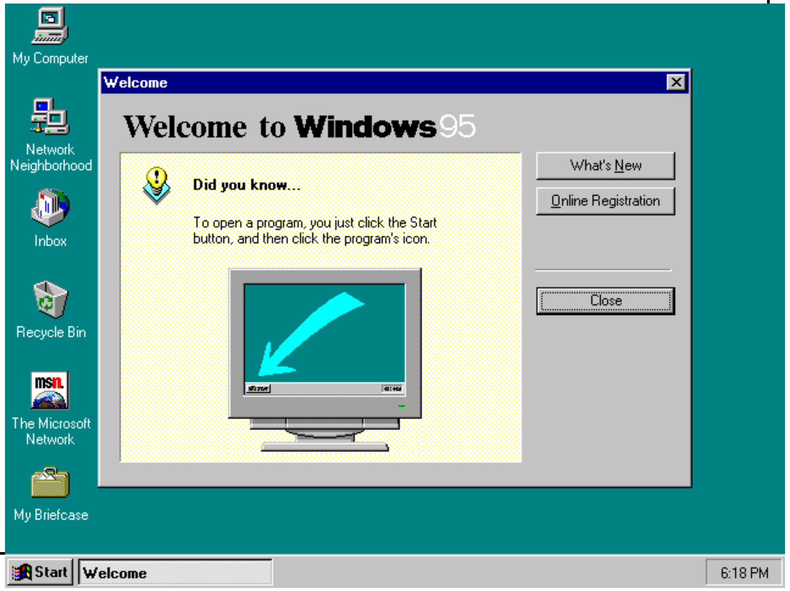
Fourth Generation of Computer (1971 – 1981)

- The first operating system for microcomputers was **CP / M** (Control Program for Microcomputers), developed by Gary Kildall for the Intel 8080 processor.
- The other great operating system of the time was **MS-DOS**. This system has its origin in the DOS system (Disk Operating System) of Seatle Computer Products. This was acquired by Bill Gates and it soon dominated the PC market.

```
Starting MS-DOS...

C:\>_
```

Fifth Generation of Computer (1981 – …)

- Microsoft looked for a successor to MS-DOS, so it produced a graphical interface system which called **Windows** and was originally running on top of MS-DOS, that is, it was a shell and not a real operating system.
- MAC OS, the operating system was introduced in 1984 to run the company's Macintosh line of personal computers (PCs).
- **Solaris** is an operating system based on Unix developed in 1992 by Sun Microsystems.
- In 1995 he released the first version of Windows that could be considered a true operating system, **Windows 95**.



Welcome to Windows95

Did you know...

To open a program, you just click the Start button, and then click the program's icon.

Activity I

At the address https://www.computerhistory.org/timeline/computers/ we can find information about the history of computers. From the images you find there, build a timeline showing some of the machines. Find out their corresponding operating system.

Indications: the minimum information on each machine will be:

- Machine name
- Machine image
- Year of appearance
- Developers
- Use for which it was designed
- **Operating system running**

3. TYPES OF OPERATING SYSTEM

- **Server operating systems** *(sistemas operativos para servidores)***:** they are systems designed to interact with other systems (clients) offering services.

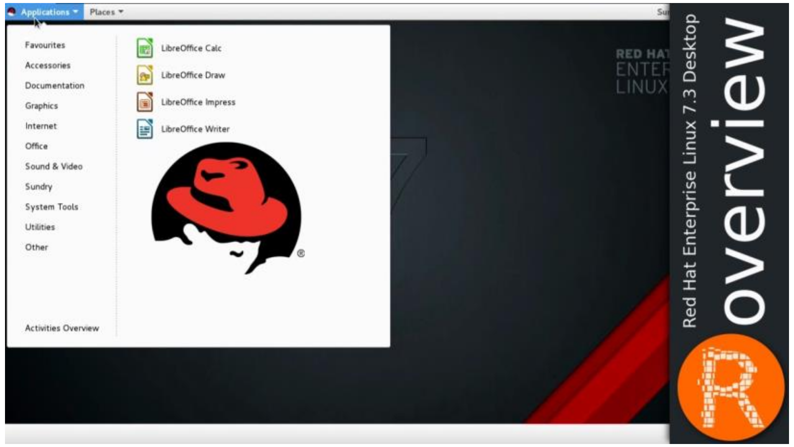
  For this reason, they do not usually have Graphical User Interface (GUI, Graphic User Interface) or have one with functionalities limited, it is because the GUI consumes resources and is potentially causing of system errors. They usually have CLI (Command Line Interface).

  UNIX; Windows server, 2008, 2012, 2016, 2019; Red Hat Enterprise Linux Server, Ubuntu server, Solaris, etc.

- **Desktop Operating Systems** *(sistemas operativos de escritorio)*: They work directly to offer their services to users, so their interfaces are usually graphical.

  Almost all of these systems are distributed among the domestic consumer market.

  Mac OS X; Windows 7, 8, 10, 11…; Ubuntu; Linux Mint; Red Hat Enterprise, Linux Desktop, etc.;



- **Embedded Operating Systems Operating Systems** *(sistemas operativos embebidos)***:** are a specialized operating system designed to perform a specific task for a device that is not a computer work directly to offer their services to users, so their interfaces are usually graphical.

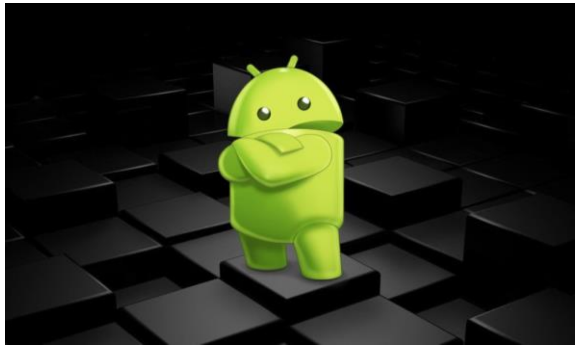
Cisco IOS for routers Cisco**; OSEK operating system** for cars.

- **Device Operating Systems** *(sistemas operativos de dispositivo)*: They are systems designed to run on a very particular type of computer: handheld devices.

  This type of device can be current smartphones and tablets.
  They are systems that are halfway between systems desktop and embedded operating systems.

  They have to work on more hardware limited than desktop computers, but increasingly they are demanding more functions and user interfaces more advanced.

  Android, iOS, HarmonyOS,
  Firefox OS (last version 2015) , etc.



- **Network Operating Systems** *(sistemas operativos de internet)*: They are systems conceiving as operating systems in which both applications and user data reside in the cloud.

  In this way it is possible to reduce the hardware needs lowering the price of equipment. In exchange they demand permanent connection to the Internet to obtain all its functionalities.

Chrome OS, EyeOS

eyeos

chromeOS

- **Real Time Operating Systems** *(sistemas operativos en tiempo real)*: a Real-time operating system is an operating system intended to serve real time application that process data as it comes in, mostly without delay.

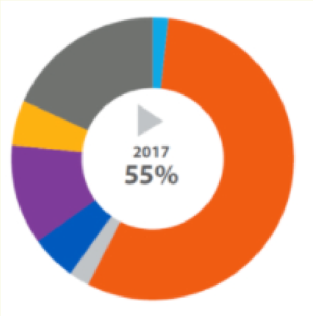
  Are used in aeronautics, industrial production, motoring...

  RTLinux, Lynx OS

Many systems can be part of several types simultaneously: there are a multitude of Linux distributions that can be configured as server or desktop operating system; there are systems device operating instructions that can be found in embedded systems; time operating systems real are usually found as embedded systems; etc.

Activity II

Look for information about which are the main smartphone operating systems and which is its market share (cuota de mercado).



After this, make a table indicating:

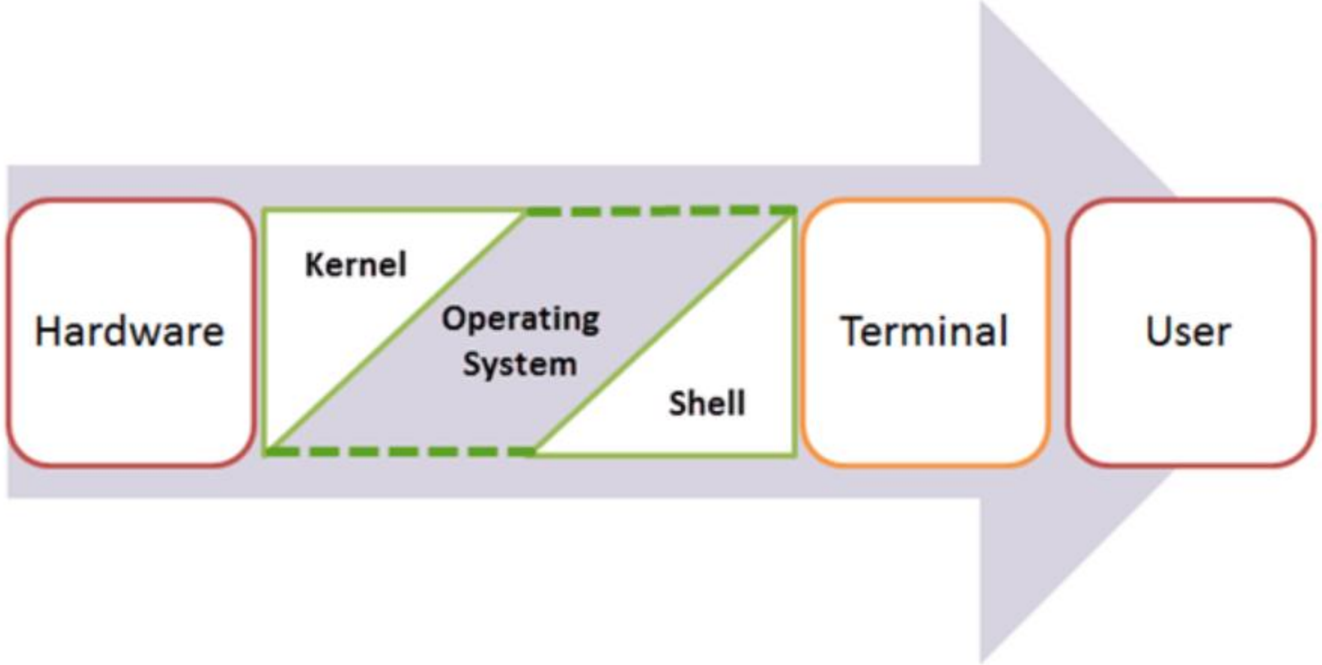
- Operating system
- Company
- Characteristics

4. STRUCTURE OF OPERATING SYSTEMS

- Each operating system has its own structure that is transparent to the user and applications.
- Normally the operating system is made up of **a series of utilities** for its administration and use; **an interface of user** that can be graphical (GUI Graphical User Interface) or not (CLI Command Line Interface); and a **kernel** that **is the base of the system**.
- If we focus on the kernel, what it is really distinctive from one system to another (they can share utilities and interface, but what differentiates systems is your kernel) we can find:
  - **Monolithic cores** (núcleos monolíticos)
  - **Layered systems** (sistemas por capas)
  - **Virtual machine systems** (sistemas de máquina virtual)

What is Kernel in Operating System?

- The kernel is the central component of a computer operating systems. It makes the communication between the hardware and software possible. While the Kernel is the innermost part of an operating system, a shell is the outermost one.



4.1 Monolithic cores (núcleos monolíticos)

- A monolithic kernel is an operating system architecture where the entire operating system is working in kernel space. All functions of the system are in a series of procedures in the kernel.
- This kernel must be fully loaded in memory and run to function correctly.

Microsoft Windows 9x series (95, 98).

Linux.

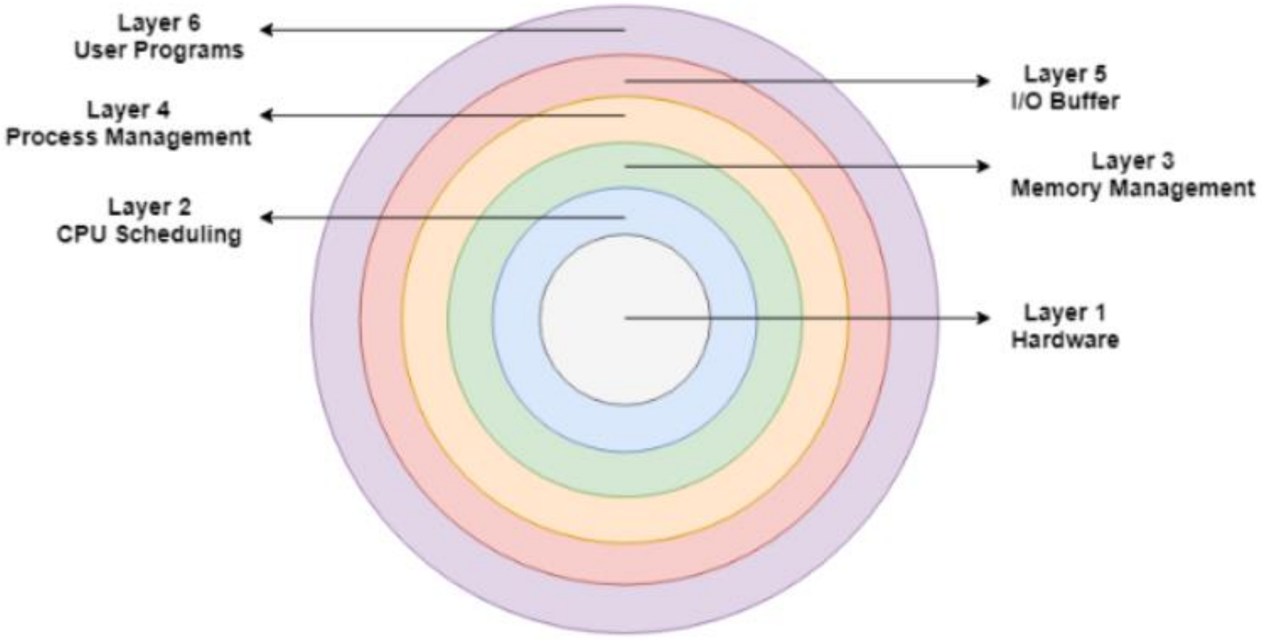
Unix.

4.2 Layered systems (sistemas por capas)

- The operating system is split into various layers In the layered operating system and each of the layers have different functionalities. This type of operating system was created as an improvement over the early monolithic systems.
- The kernel is divided into a series of layers.

Why Layering in Operating System?

Layering provides a distinct advantage in an operating system. All the layers can be defined separately and interact with each other as required. Also, it is easier to create, maintain and update the system if it is done in the form of layers. Change in one layer specification does not affect the rest of the layers.



LAYERED OPERATING SYSTEM

Monolithics lead to modular systems. Kernel can be divided into separate modules, that is, a modular structure. That can be loaded and dynamically download when needed - Linux tends to be a modular system.

4.3 Virtual machine systems (sistemas de máquina virtual)

- On this system, the virtual machine monitor, runs on hardware only and performs multiprogramming, providing multiple virtual machines identical to the next layer up. Virtual machines are identical copies of the hardware of the real machine that can even run different operating systems.

KVM

VirtualBox

Xen

Wmware

5. FUNCTIONS OF OPERATING SYSTEM

- An operating system **manages the computer hardware**.
- **Provide the user with a suitable interface** to work.
- **Process Management/Program Execution:** The Operating System is responsible for the execution of all types of programs.
- **Manipulation of File System:** The Operating System is responsible for making decisions regarding the storage of all types of data or files, i.e, floppy disk/hard disk/pen drive, etc. The Operating System decides how the data should be manipulated and stored.
- **Error Detection and Handling:** The Operating System is responsible for the detection of any type of error or bugs that can occur while any task.

5.1 Process Management (Gestión de procesos)

- The operating system decides which tasks should be executed at any given time, creating and destroying processes; and facilitating communication between them. This management must be done making the most of the processor.
- We will learn more about this …

Program vs Process

A process is a program in execution. For example, when we write a program in C or C++ and compile it, the compiler creates binary code. The original code and binary code are both programs. When we actually run the binary code, it becomes a process.

A process is an 'active' entity, instead of a program, which is considered a 'passive' entity. A single program can create many processes when run multiple times; for example, when we open a .exe or binary file multiple times, multiple instances begin (multiple processes are created).

Attributes or Characteristics of a Process

All of the attributes of a process are also known as **the context of the process**. Every process has its own process control block (PCB) (Bloque de control de proceso BCP). All of the above attributes are part of the PCB.

Attributes:

1. **Process Id:** A unique identifier assigned by the operating system.

2. **Process State:** Can be ready, running, etc.

3. **CPU registers:** Like the Program Counter (CPU registers must be saved and restored when a process is swapped in and out of CPU).

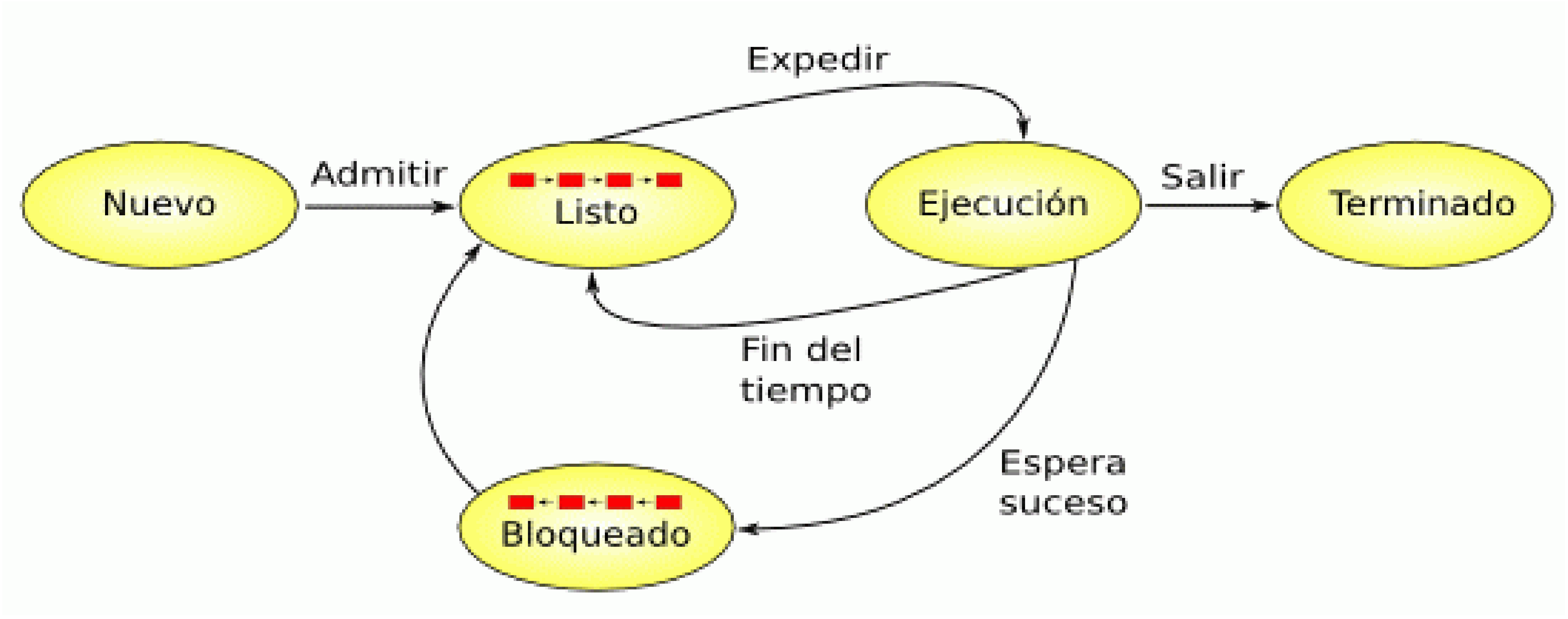
4. **I/O status information:** For example, devices allocated to the process, open files, etc

5. **CPU scheduling information:** For example, Priority (Different processes may have different priorities)

States of Process:

A process is in one of the following states:

1. **New:** Newly Created Process (or) being-created process.
2. **Ready:** After creation process moves to Ready state, i.e. the process is ready for execution.
3. **Run:** Currently running process in CPU (only one process at a time can be under execution in a single processor).
4. **Wait (or Block):** When a process requests I/O access.
5. **Complete (or Terminated):** The process completed its execution.
6. **Suspended:** When the ready queue becomes full, some processes are moved to suspended ready state.


Expedir
Nuevo
Admitir
Listo
Ejecución
Salir
Terminado
Fin del tiempo
Espera suceso
Bloqueado

5.1.1 CPU Scheduling in Operating Systems (Planificador de procesos)

- A multiprogrammed computer usually has several processes competing for the CPU in the same time. This situation occurs every time two or more processes are in the state ready simultaneously and the operating system has to decide which of the processes should run first.
- The part of the operating system that makes the decision is called **CPU Scheduler** (planificador de procesos).
- To make that decision, different algorithms are used, called **scheduling algorithms** (algoritmos de planificación).

5.1.2 Different Scheduling Algorithms (Algoritmos de planificación)

Objectives of Process Scheduling Algorithm

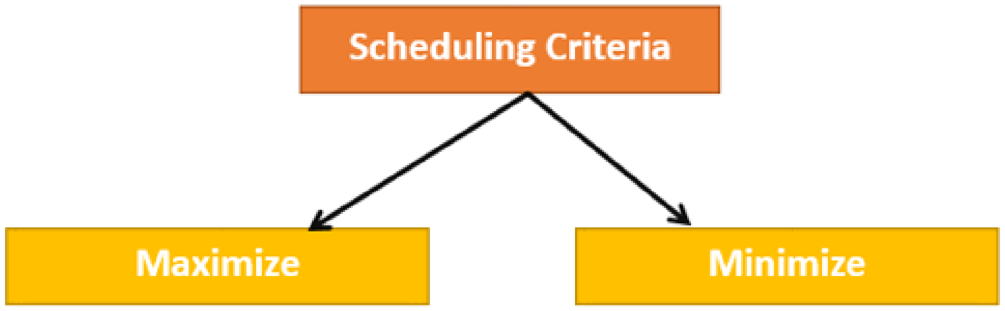
Max CPU utilization [Keep CPU as busy as possible]

Max throughput (rendimiento máximo) [Number of processes that complete their execution per time unit]

Min turnaround time (tiempo de respuesta mínimo) [Time taken by a process to finish execution]

Min waiting time [Time a process waits in ready queue]

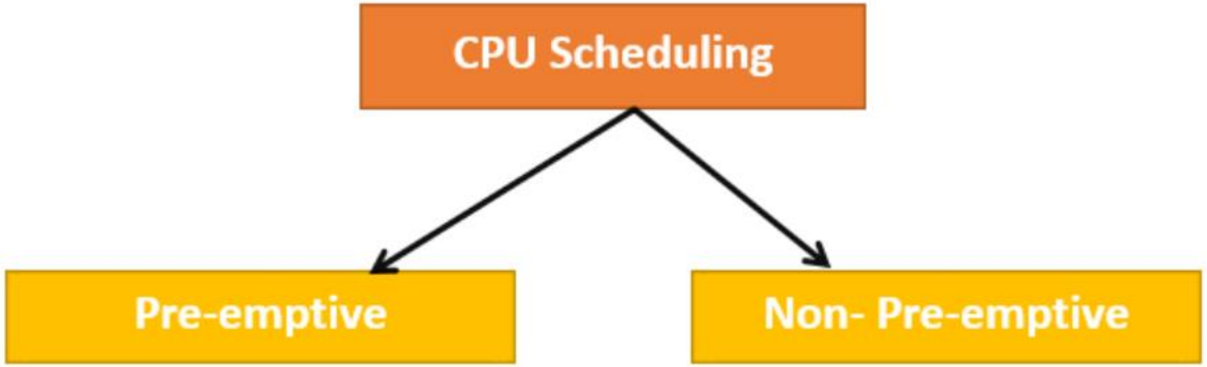
Min response time [Time when a process produces first response]



Maximize:
CPU Utilization
Throughput

Minimize:
Turnaround time
Waiting time
Response time

Types of CPU Scheduling (tipos de algoritmos de planificación)



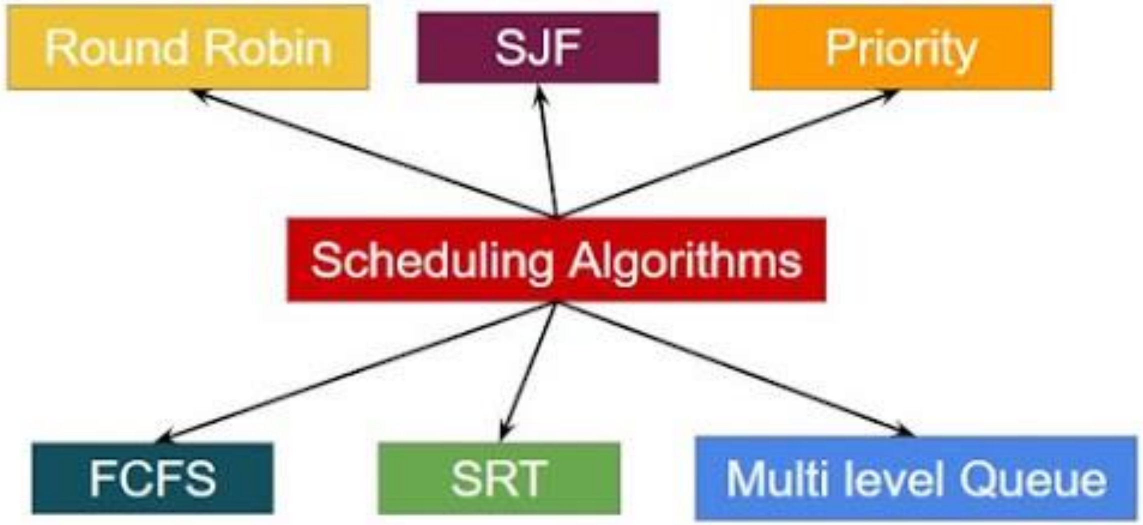
Preemptive Scheduling (Algoritmos Expulsivos/Expropiativos)
Processes can only occupy the CPU for one time determined, called quantum. When that time expires the planner forces the process to leave the CPU, even if the execution has not finished, in which case it will go to the ready state.

Non-Preemptive Scheduling (Algoritmos No expulsivos/No expropiativos)
When a process occupies the CPU, it occupies the time it needs. It is it who voluntarily cedes the CPU to others.

It has the disadvantage that one process can monopolize the processor (Infinite loop).

Names of CPU scheduling Algorithm (Nombres de algoritmos de planificación)

- ✓ **First Come First Serve (FCFS) or First Input First Output (FIFO)** (PRIMERO EN LLEGAR, PRIMERO EN SER SERVIDO) (PRIMERO EN LLEGAR, PRIMERO EN SALIR)
- ✓ **Shortest-Job-First (SJF)** (PRIMERO EL PROCESO MÁS CORTO)
- ✓ **Shortest Remaining Time (SRT)** (PRIMERO EL QUE TENGA TIEMPO RESTANTE MÁS CORTO)
- ✓ **Priority Scheduling** (PLANIFICACIÓN POR PRIORIDADES)
- ✓ **Round Robin Scheduling (RR)** (PLANIFICACIÓN POR RONDAS)



+ Examples in the notebook

Algorithm parameters:

Of each algorithm we know:

- **Entry time** (T_I) (Tiempo de entrada o de llegada al sistema) → The moment when the process enters in the system.
- **Runtime** (Tiempo de ejecución) (T_X) → Time that the process needs for its execution.

We can calculate the following for each algorithm:

- **Response time** (T_R) (Tiempo de respuesta o tiempo de retorno) → The time when the process arrives until it ends.
- **Waiting time** (T_E) (Tiempo de espera) → It is the time that the process spends waiting within the system.

$$(T_E) = (T_R) - (T_X)$$

We can calculate the average times:

- We can calculate the **average response** and **average wait times**.

First come, First served (FCFS) (Primero en llegar, primero en salir)

It is the easiest and most simple CPU scheduling algorithm. In this type of algorithm, the process which requests the CPU gets the CPU allocation first. This scheduling method can be managed with a FIFO queue.

FCFS is a non-preemptive scheduling algorithm. → Algoritmo no expulsivo.

Shortest-Job-First (SJF) (Primero el proceso más corto)

Process which have the shortest burst time at the execution time are scheduled first. If two processes have the same bust time then FCFS is used to break the tie. It favors processes that take less time to run.

It is a non-preemptive scheduling algorithm.

Shortest Remaining Time (SRT)

It is preemptive mode (el modo expulsivo) of SJF algorithm in which jobs are schedule according to shortest remaining time.

It is a preemptive scheduling algorithm.

Round Robin Scheduling (planificación por rondas)

It is similar to FCFS scheduling, the processes enter the CPU following a FIFO scheme but with a a limited amount of CPU time called Quantum. If the active process is still in execution at the end of its quantum will be pulled from the CPU by the scheduler passing to the end of the tail.
A very long Quantum degenerates into an FCFS system.

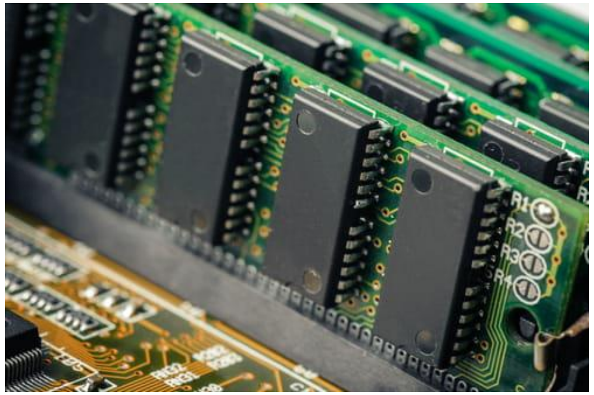
Round Robin is a preemptive scheduling algorithm. → Algoritmo expulsivo

Priority Scheduling (planificación por prioridades)

In this scheduling, processes are scheduled according to their priorities, i.e., highest priority process is scheduled first. If priorities of two processes match, then schedule according to arrival time.

5.2 Memory Management (Gestión de memoria)

- As we mentioned, one of the functions that the operating system has to do is the efficient management of the main memory of the equipment (RAM memory).
- Note that, despite the constant increase in amount of available memory that has been lived for the last years; **memory needs on the part of systems and applications have grown** in the same or greater measure.
- In this way we can understand that **the appropriate management it is essential to have a good performance**.



Main memory - RAM

RAM gives applications a place to store and access data. **It stores the information your computer is actively using** so **that it can be accessed quickly.**

The more programs your system is running, the more you'll need.

And the operating systems are multitasking, it means that it **allowing a user to perform more than one computer task** (such as the operation of an application program) **at a time**.

Virtual memory

Many years ago, the first programs appeared were too big to fit in available memory. The solution adopted was to divide the program into fragments, called overlays (superposiciones) so they only need to be in memory the one that was running at that time. The division into fragments was a tedious task to the programmer. The solution to this problem is virtual memory.

The main idea is that if a computer running an operating system needs more memory or RAM than the memory installed in the system then it uses a small portion of the hard drive for this purpose.

Techniques of virtual memory

Paging

Paginación

Segmentation

Segmentación

5.2.1 What is paging? (paginación)

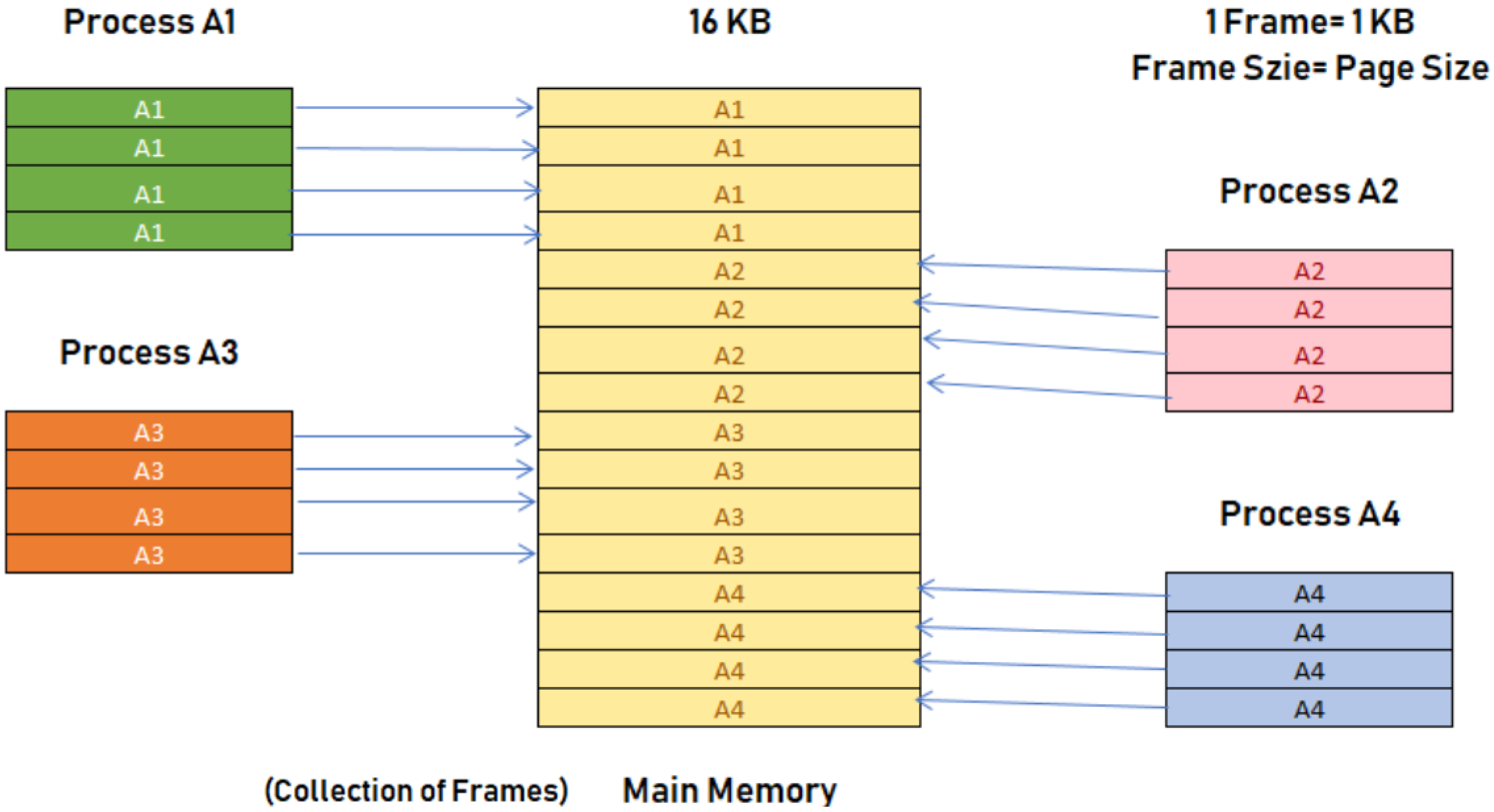
In the Paging method, the **main memory is divided into small fixed-size blocks** of physical memory, which is called **frames**. **The programs are divided into blocks of the same size called pages**. Then you just have to locate the pages of a process in the free memory frames.

Example of paging

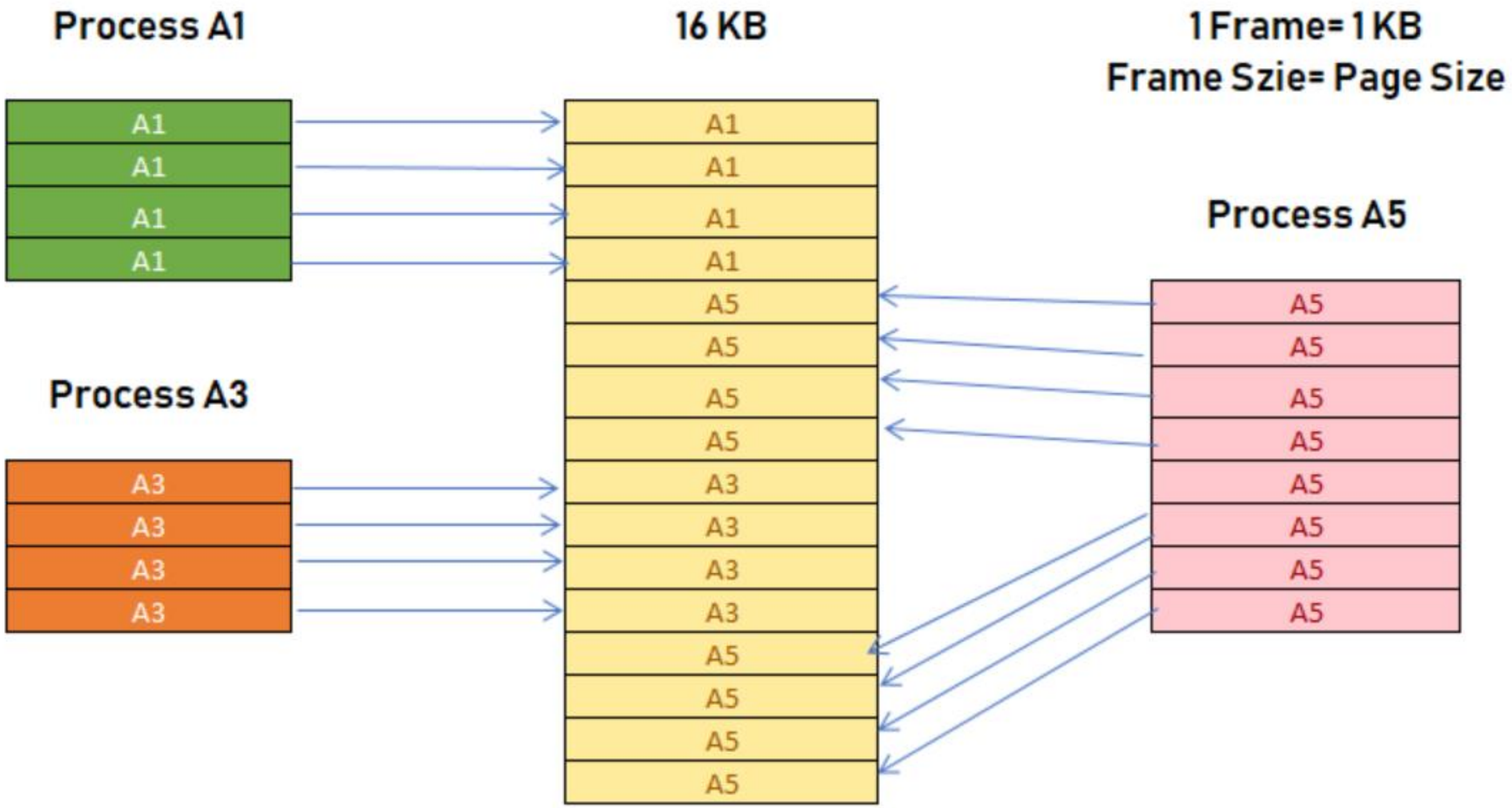
For example, if the main memory size is 16 KB and Frame size is 1 KB. Here, the main memory will be divided into the collection of 16 frames of 1 KB each.

There are 4 separate processes in the system that is A1, A2, A3, and A4 of 4 KB each. Here, all the processes are divided into pages of 1 KB each so that operating system can store one page in one frame.

At the beginning of the process, all the frames remain empty so that all the pages of the processes will get stored in a contiguous way.


Process A1
16 KB
1 Frame= 1 KB
Frame Szie= Page Size
A1
A1
A1
A1
Process A2
A2
A2
A2
A2
Process A3
A3
A3
A3
A3
Process A4
A4
A4
A4
A4
A1
A1
A1
A1
A2
A2
A2
A2
A3
A3
A3
A3
A4
A4
A4
A4
(Collection of Frames)
Main Memory

In this example you can see that A2 and A4 are moved to the waiting state after some time. Therefore, eight frames become empty, and so other pages can be loaded in that empty blocks. The process A5 of size 8 pages (8 KB) are waiting in the ready queue.



Process A1

A1
A1
A1
A1

Process A3

A3
A3
A3
A3

16 KB

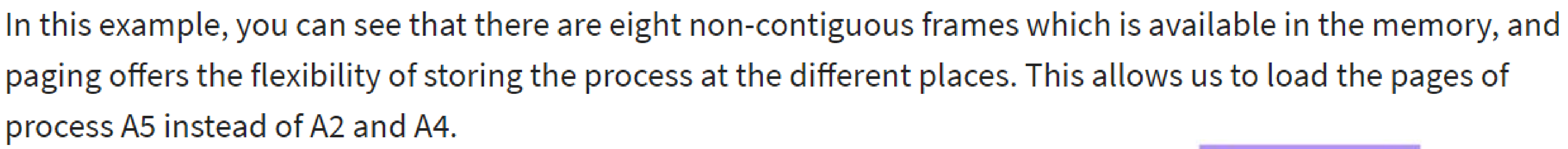
A1
A1
A1
A1
A5
A5
A5
A5
A3
A3
A3
A3
A5
A5
A5
A5

1 Frame= 1 KB
Frame Szie= Page Size

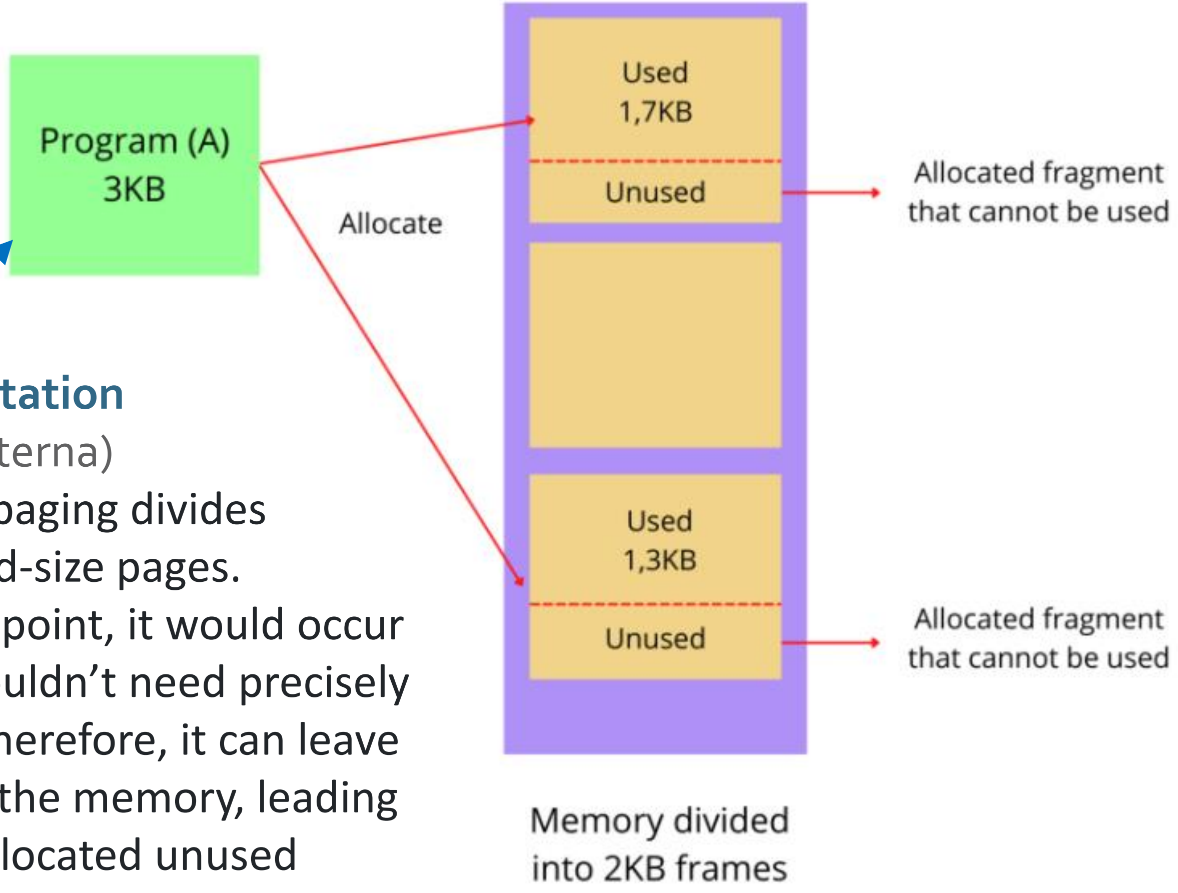
Process A5

A5
A5
A5
A5
A5
A5
A5
A5

Main Memory (Collection of Frames)



In this example, you can see that there are eight non-contiguous frames which is available in the memory, and paging offers the flexibility of storing the process at the different places. This allows us to load the pages of process A5 instead of A2 and A4.



Internal Fragmentation

(fragmentación interna)

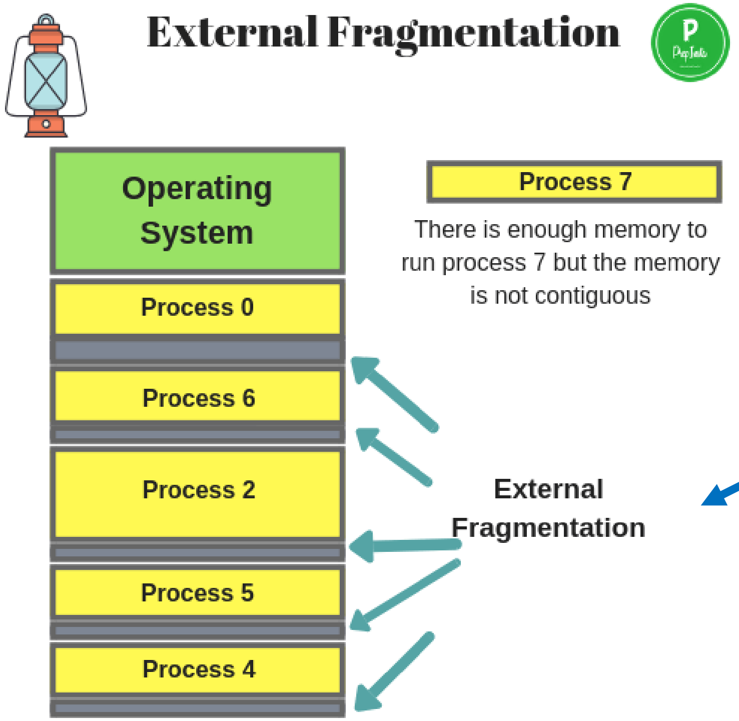
We already know paging divides programs into fixed-size pages. However, at some point, it would occur that a program wouldn’t need precisely the whole page. Therefore, it can leave a free partition of the memory, leading to memory with allocated unused spaces.

Paging may cause **Internal fragmentation**

5.2.2 What is segmentation? (segmentación)

Segmentation method works almost similarly to paging, only difference between the two is that segments are of variable-length whereas, in the paging method, pages are always of fixed size.

A program segment includes the program's main function, data structures, utility functions, etc. The OS maintains a segment map table for all the processes. It also includes a list of free memory blocks along with its size, segment numbers, and it's memory locations in the main memory or virtual memory.



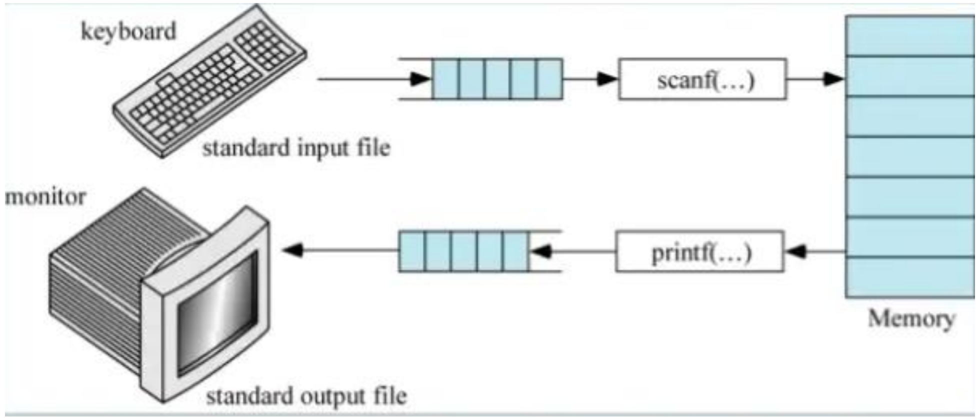
External Fragmentation

There is enough memory to run process 7 but the memory is not contiguous

In segmentation method, processes are loaded/ removed from the main memory. Therefore, the free memory space is separated into small pieces which may create a problem of **external fragmentation**.

5.3 I/O Management (Gestión de entrada/salida E/S)

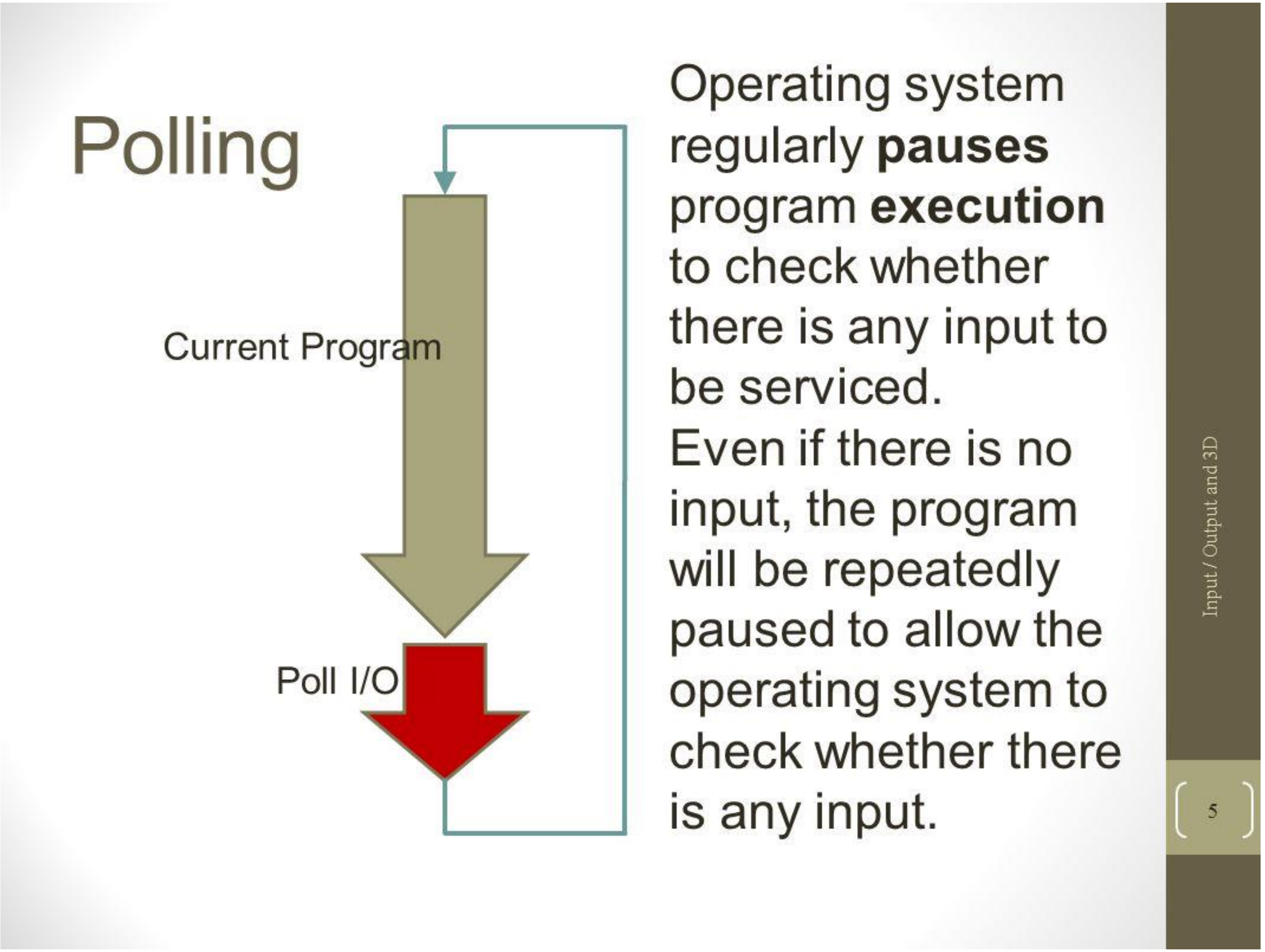
- The primary role of the operating system in computer Input / Output is to manage and organize I/O operations and all I/O devices.
- The controlling of various devices that are connected to the computer is a key concern of operating-system designers. This is because **I/O devices vary so widely in their functionality and speed** (for example a mouse, a hard disk and a CD-ROM), **varied methods are required for controlling them**. These methods form the **I/O unit of the kernel** of OS that separates the rest of the kernel from the complications of managing I/O devices.



5.3.1 I/O methods (Métodos E/S)

- **POLLING** (E/S controlada por programa)
- **INTERRUPT CONTROLLED I/O** (E/S controlada mediante interrupciones)
- **DIRECT MEMORY ACCESS (DMA)** (E/S mediante acceso directo a memoria)

Polling (E/S controlada por programa)



Polling

Operating system regularly **pauses** program **execution** to check whether there is any input to be serviced.

Even if there is no input, the program will be repeatedly paused to allow the operating system to check whether there is any input.

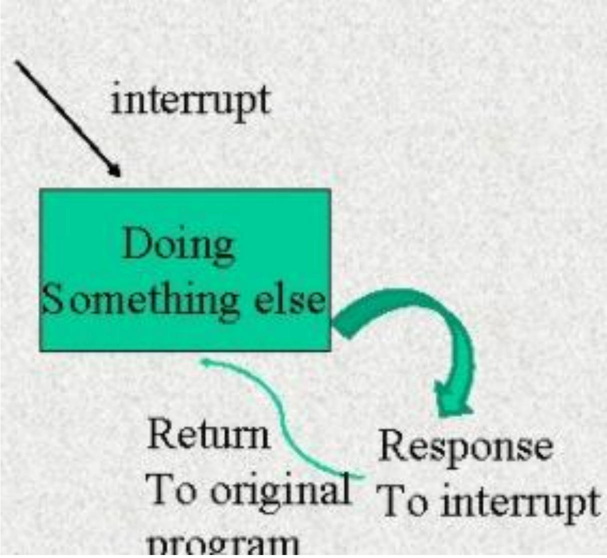
Input / Output and 3D

5

CPU loses time doing these queries.

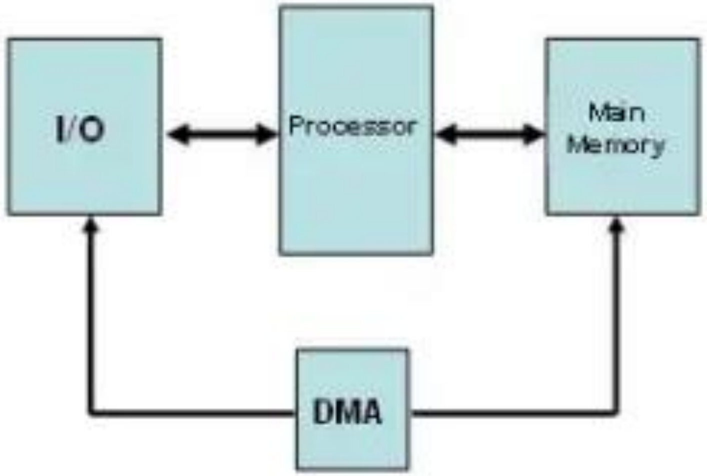
Interrupt controlled I/O (E/S controlada por programa)

Interrupts are signals sent to the CPU by external devices, normally I/O devices. They tell the CPU to stop its current activities and execute the appropriate task that the I/O devices need.



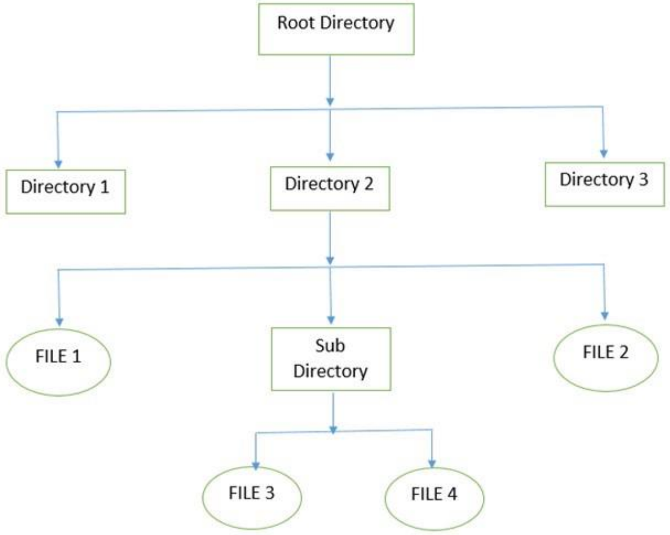
Direct Memory Access (DMA) (E/S mediante acceso directo a memoria)

- A defined portion of memory is used to send data directly from a peripheral to the motherboard without involving the microprocessor, so that the process does not interfere with overall computer operation.
- Multiple hardware systems adopt Direct Memory Access, such as hard disk controllers, graphics cards, network cards, and sound cards.

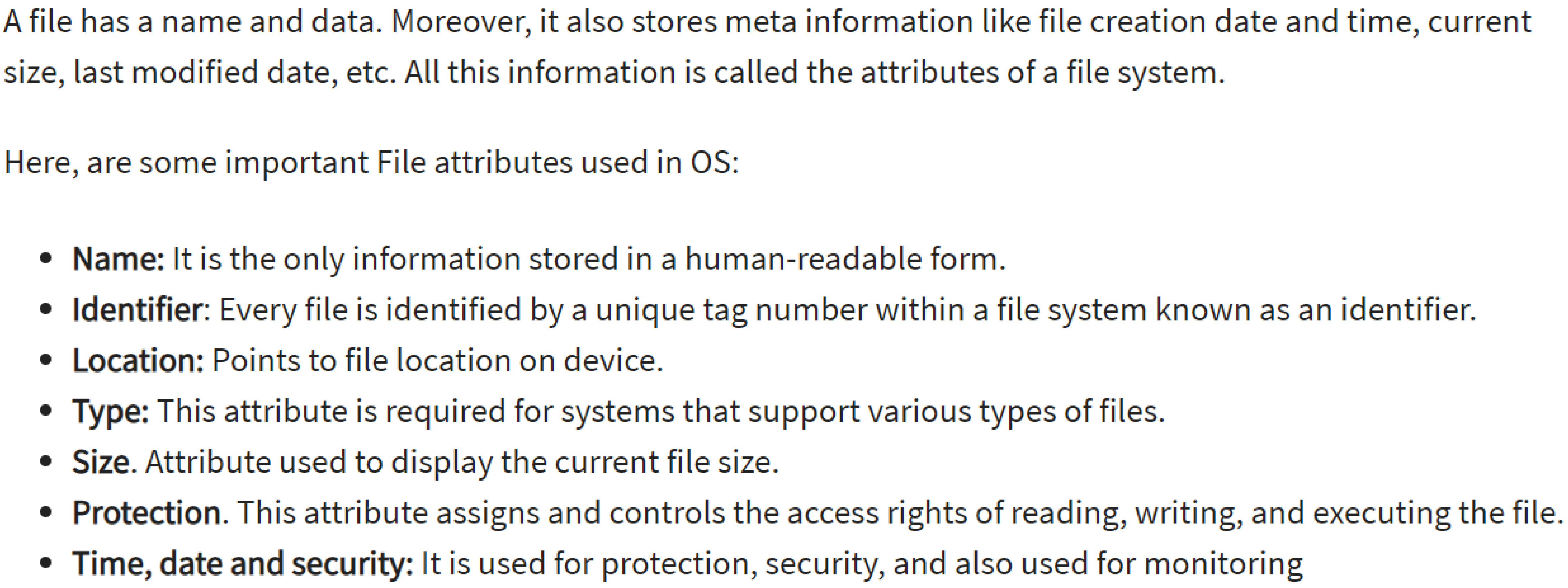


5.4 File Management (Gestión de archivos)

- File management is one of the basic and important features of operating system. **Operating system is used to manage files of computer system**. All the files with different extensions are managed by operating system.
- A **file** is collection of specific information stored in the memory of computer system**. File management is defined as the process of manipulating files in computer system**, its management includes the process of creating, modifying and deleting the files.



File Attributes



A file has a name and data. Moreover, it also stores meta information like file creation date and time, current size, last modified date, etc. All this information is called the attributes of a file system.

Here, are some important File attributes used in OS:

- **Name:** It is the only information stored in a human-readable form.
- **Identifier**: Every file is identified by a unique tag number within a file system known as an identifier.
- **Location:** Points to file location on device.
- **Type:** This attribute is required for systems that support various types of files.
- **Size**. Attribute used to display the current file size.
- **Protection**. This attribute assigns and controls the access rights of reading, writing, and executing the file.
- **Time, date and security:** It is used for protection, security, and also used for monitoring

5.4.1 File systems (Sistemas de archivos)

- File systems are responsible for managing persistent data, and it is therefore essential to ensure that they function correctly.

FAT

FAT32 is one of the oldest of the three file systems available to Windows.

It is introduced this system in Windows 95 to replace the FAT16 file system used with older OS systems like DOS and Windows 3. **Individual files on a FAT32 drive cannot excessed 4 GB in size, which is maximum**.

FAT32 partition should be less than 8 TeraByte (TB).

EXFAT

The exFAT file system was introduced in 2006 and was added to older versions of Windows with updates to Windows XP and Windows Vista operating systems.

NTFS

NTFS is a modern-day file system that is used by default used by Windows. When you install Windows 10 into your PC or laptop, it formats your system drive with the NTFS file system.

This file system has the file size and partition size limits, which are so huge.

NTFS file system made it's first debut with Windows XP. It supports file permissions for security, shadow copies of your backup, etc.

EXT2, EXT3, EXT4

Ext2, Ext3, and Ext4 are all extended file systems created for Linux. But they have differences in many aspects like file size limit.

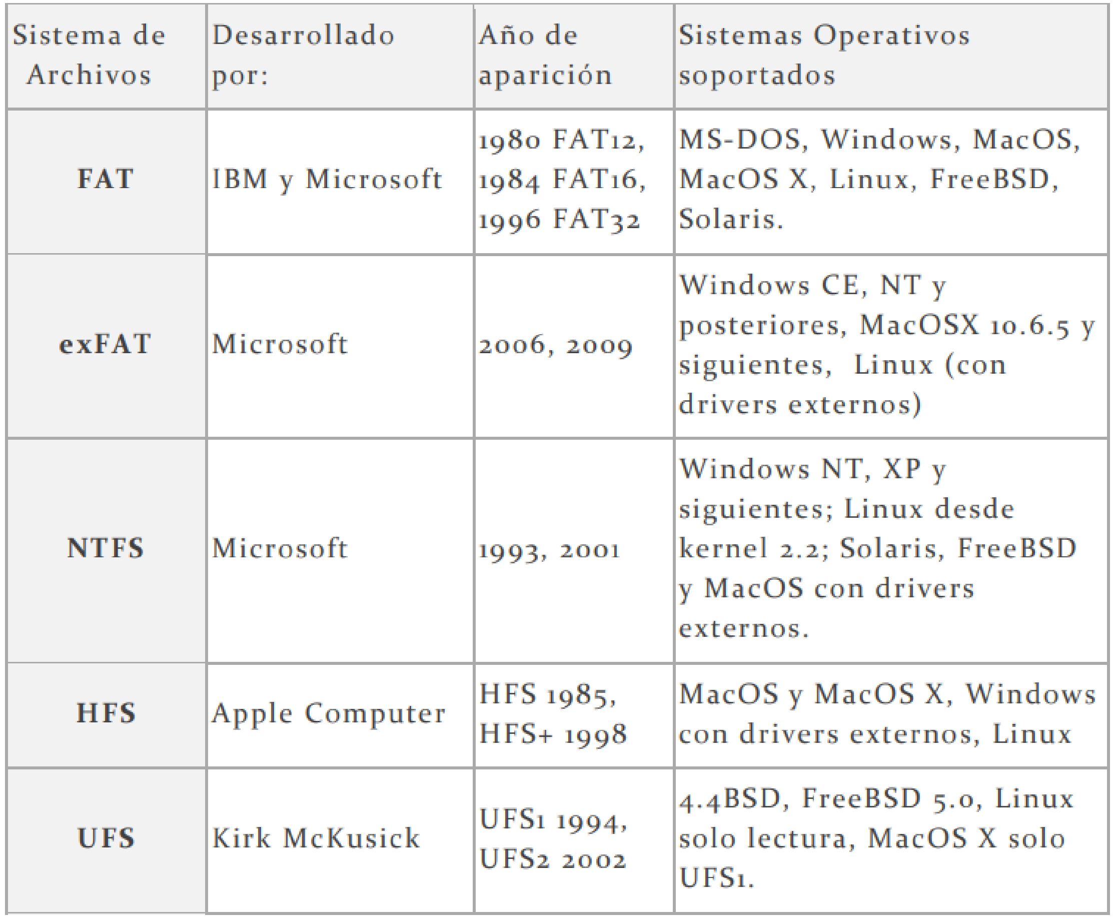
ReiserFS

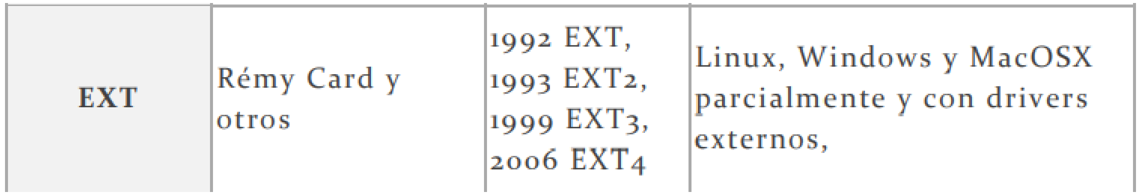
ReiserFS (Reiser File System) is a journaling file system for Linux. ReiserFS is known to be particulary good at handling the large numbers of small files. It is the default file system on a number of distributions, including: Elive, Xandros, Linspire, and GoboLinux.

HFS y HFS+

HFS Plus or HFS+ (also known as Mac OS Extended or HFS Extended) is a journaling file system developed by Apple Inc.

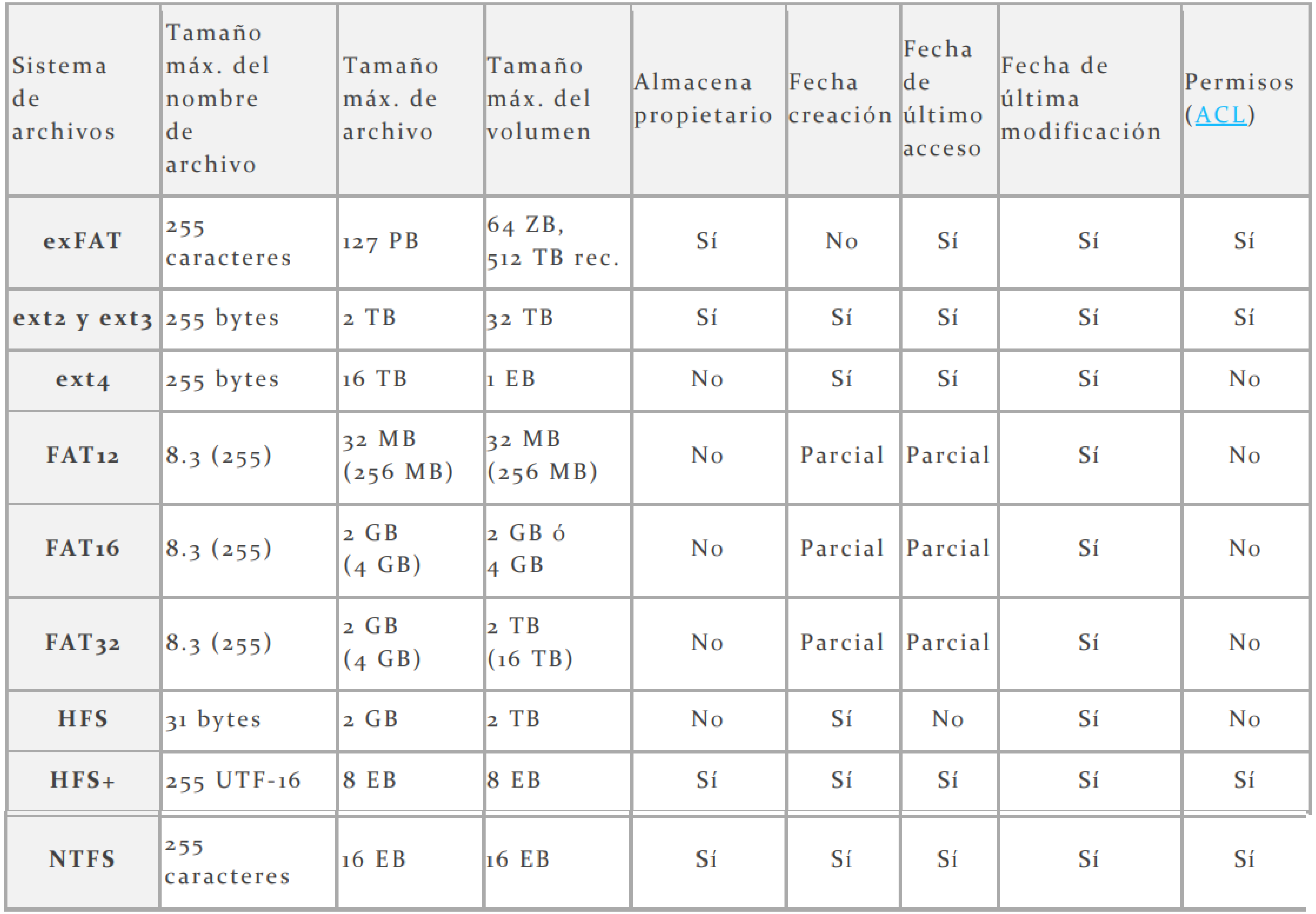
Tabla 1: Principales sistemas de archivos





Sistema de Archivos	Desarrollado por:	Año de aparición	Sistemas Operativos soportados
FAT	IBM y Microsoft	1980 FAT12, 1984 FAT16, 1996 FAT32	MS-DOS, Windows, MacOS, MacOS X, Linux, FreeBSD, Solaris.
exFAT	Microsoft	2006, 2009	Windows CE, NT y posteriores, MacOSX 10.6.5 y siguientes, Linux (con drivers externos)
NTFS	Microsoft	1993, 2001	Windows NT, XP y siguientes; Linux desde kernel 2.2; Solaris, FreeBSD y MacOS con drivers externos.
HFS	Apple Computer	HFS 1985, HFS+ 1998	MacOS y MacOS X, Windows con drivers externos, Linux
UFS	Kirk McKusick	UFS1 1994, UFS2 2002	4.4BSD, FreeBSD 5.0, Linux solo lectura, MacOS X solo UFS1.
EXT	Rémy Card y otros	1992 EXT, 1993 EXT2, 1999 EXT3, 2006 EXT4	Linux, Windows y MacOSX parcialmente y con drivers externos,

Tabla 2: Principales características de los sistemas de archivos



Sistema de archivos	Tamaño máx. del nombre de archivo	Tamaño máx. de archivo	Tamaño máx. del volumen	Almacena propietario	Fecha creación	Fecha de último acceso	Fecha de última modificación	Permisos (ACL)
exFAT	255 caracteres	127 PB	64 ZB, 512 TB rec.	Sí	No	Sí	Sí	Sí
ext2 y ext3	255 bytes	2 TB	32 TB	Sí	Sí	Sí	Sí	Sí
ext4	255 bytes	16 TB	1 EB	No	Sí	Sí	Sí	No
FAT12	8.3 (255)	32 MB (256 MB)	32 MB (256 MB)	No	Parcial	Parcial	Sí	No
FAT16	8.3 (255)	2 GB (4 GB)	2 GB ó 4 GB	No	Parcial	Parcial	Sí	No
FAT32	8.3 (255)	2 GB (4 GB)	2 TB (16 TB)	No	Parcial	Parcial	Sí	No
HFS	31 bytes	2 GB	2 TB	No	Sí	No	Sí	No
HFS+	255 UTF-16	8 EB	8 EB	Sí	Sí	Sí	Sí	Sí
NTFS	255 caracteres	16 EB	16 EB	Sí	Sí	Sí	Sí	Sí