

# Tema 3. Sistemas Operativos

---

Jesús María Aransay Azofra

Sistemas Informáticos

Universidad de La Rioja 2010/2011

# Índice

---

- 3.1 Misión y necesidad del sistema operativo
  - 3.2 Algunos ejemplos de sistemas operativos
  - 3.3 Gestión de usuarios y grupos
  - 3.4 Árbol de directorios
  - 3.5 El sistema de archivos - Gestión de permisos
  - 3.6 Tareas o procesos y servicios
-

## 3.1 Misión y necesidad del SO

---

1. Qué entendemos por sistema operativo
  2. Qué tipo de tareas nos evita el sistema operativo (programación de bajo nivel, ensamblador)
  3. Qué facilidades nos suele aportar un sistema operativo
  4. Estructura de un sistema operativo
  5. Gestión de memoria y recursos por parte del SO
-

## 3.1.1 Definición de SO

---

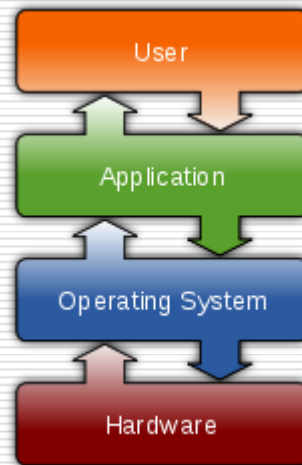
Existen muy diversas definiciones de Sistema Operativo. Sin embargo, todas ellas tienen características comunes, que nos servirán para crear una idea concreta de qué es o para qué sirve un SO

---

## 3.1.1 Definición de SO

---

- An **operating system (OS)** is software, consisting of programs and data, that runs on computers and manages the computer hardware and provides common services for efficient execution of various application software:



## 3.1.1 Definición de SO

---

- The operating system (OS) is specialized computer software that allocates memory and manages system resources. When a computer is turned on, the OS is loaded into memory and works as an abstraction layer between the physical hardware and the software. While the OS doesn't perform a specific function it helps other programs run smoothly and efficiently.

## 3.1.1 Definición de SO

---

- "... the operating system is like a (orchestra's) conductor: It is responsible for coordinating all of the computer's individual components so that they work according to a single plan..... Similarly, the OS allocates the computer's components to different programs, synchronizes their individual activities, and provides the mechanisms that are needed so that the programs execute in perfect harmony."

## 3.1.1 Definición de SO

---

Un SO es un programa que tiene encomendadas una serie de funciones cuyo objetivo es **simplificar** el manejo y la utilización del computador, haciéndolo seguro y eficiente. Las funciones clásicas del SO se pueden resumir en:

- ❑ **Gestión de los recursos** del computador
- ❑ **Ejecución de servicios** para los programas en ejecución
- ❑ **Ejecución de los mandatos** de los usuarios



## 3.1.1 Definición de SO

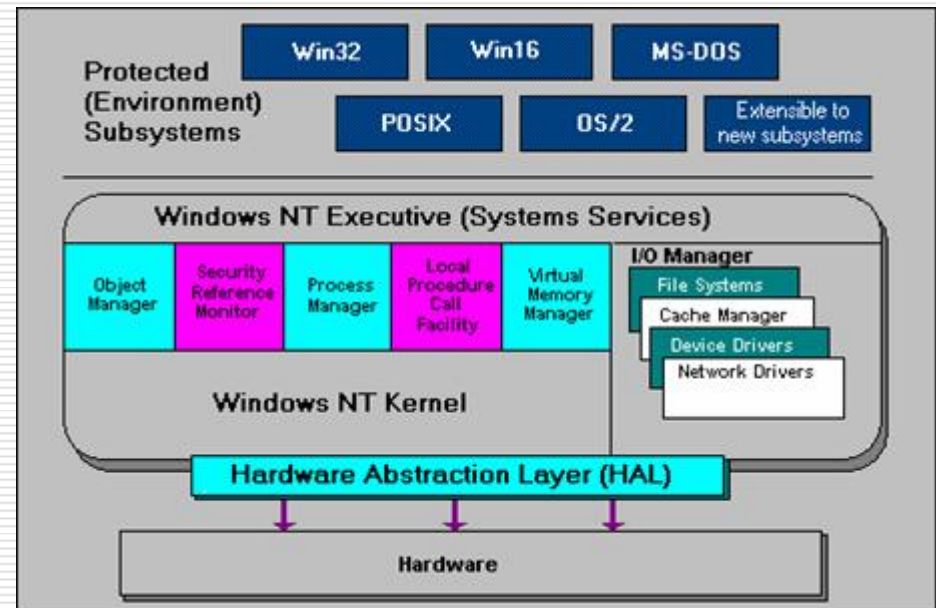
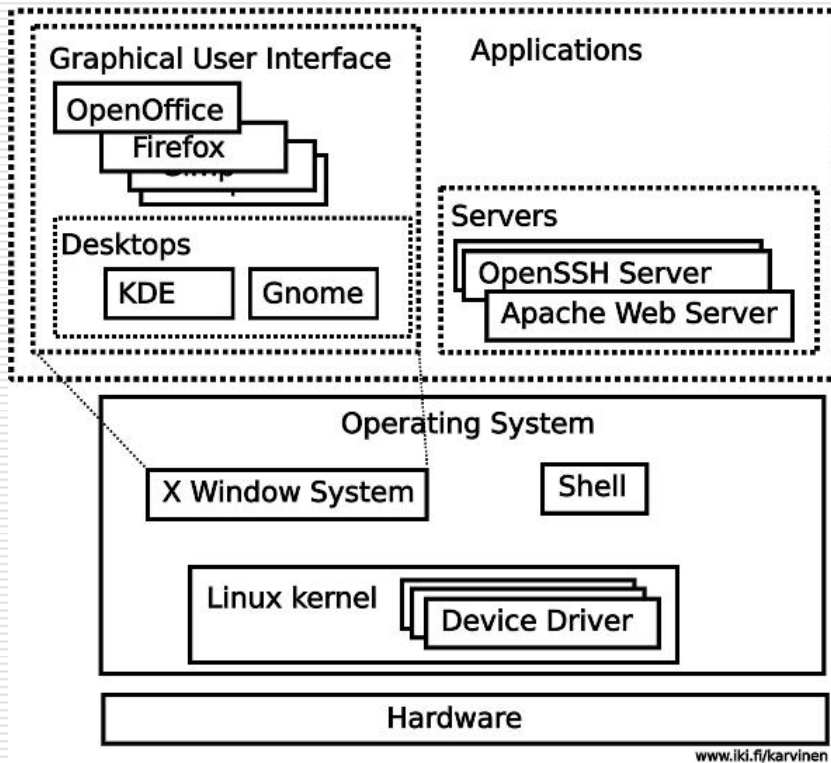
---

Tratando de resumir, un SO es:

- ☐ Un conjunto de componentes de software (de bajo nivel o de sistema) sobre la que funciona el software (de alto nivel o de aplicaciones)
  - ☐ Un gestor y coordinador de los recursos del ordenador
  - ☐ Un capa de abstracción entre la parte física (hardware) y las aplicaciones (software)
-

# 3.1.1 Definición de SO

Arquitectura de un sistema Linux      Arquitectura de sistema Windows NT

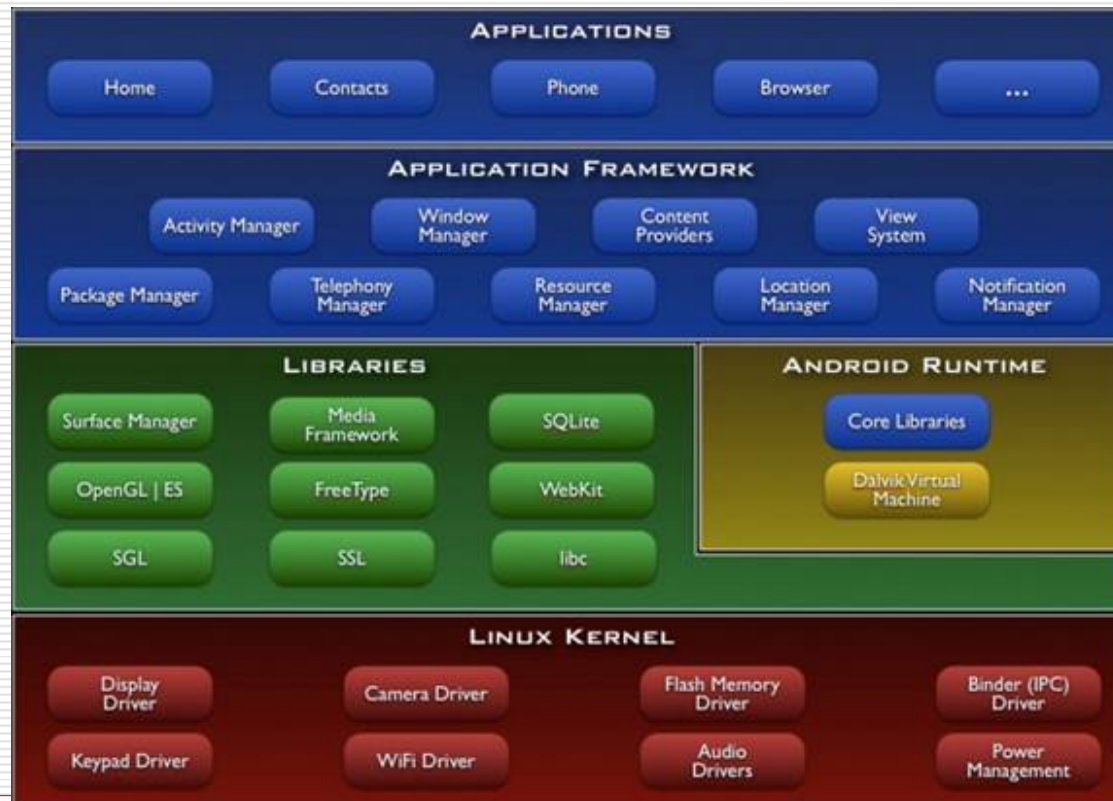


¿Dónde reside el Sistema Operativo en ambos casos?

# 3.1.1 Definición de SO

---

Arquitectura de un sistema Android, sistema basado en Linux (no aparece el hardware)



## 3.1.2 Qué tareas nos evita un SO

---

Algunos ejemplos que ya hemos visto:

- ❑ Cuando guardamos un fichero en disco duro, la localización en memoria no se hace por "Cilindros – Cabeza - Sector", sino por ficheros (sistemas de ficheros fat, fat32, ntfs, ext)
  - ❑ Los ficheros de texto no se deben "escribir" en binario (unos y ceros), sino usando caracteres de texto conforme a una tabla de caracteres (utf8, iso-8859-...)
  - ❑ La navegación por Internet no se debe hacer dividiendo la información en paquetes y asignándoles puertos, IP's, direcciones MAC; las aplicaciones (navegador...) y el sistema operativo nos evitan esa tarea
  - ❑ Al conectar un USB (un periférico) a nuestro ordenador no debemos buscar un controlador para el mismo, "montarlo"... El sistema operativo gestiona los dispositivos de I/O
-

### 3.1.2 Más ejemplos de tareas que nos evita un SO

---

- ❑ Nos evita realizar programación de bajo nivel (lenguaje ensamblador). El SO dota de APIs (interfaces o librerías de programación de aplicaciones) que abstraen la configuración de la máquina. Por ejemplo, "Win32 API", "Win16 API" en Windows, "POSIX" (Portable Operating System Interface) para UNIX, Linux y WinNT
-

### 3.1.2 Más ejemplos de tareas que nos evita un SO

---

- ❑ Coordinación y priorización de tareas (o procesos). El sistema operativo es capaz de ejecutar distintos procesos (reproducir música, enviar un mail, mantener la interfaz gráfica...) de forma que todos ellos se ejecuten simultáneamente (bien porque se paralelizan, bien porque se secuencian de modo que el usuario no lo nota)
-

## 3.1.2 Más ejemplos de tareas que nos evita un SO

Aplicaciones	Procesos	Servicios	Rendimiento	Funciones de red	Usuarios
Nombre de imagen	Nombre de usuario	CPU	Memoria ...	Descripción	
chrome.exe	gearansa	00	74.168 KB	Google Chrome	
chrome.exe	gearansa	00	55.152 KB	Google Chrome	
chrome.exe	gearansa	00	42.260 KB	Google Chrome	
explorer.exe	gearansa	00	37.280 KB	Explorador de Windows	
chrome.exe	gearansa	00	34.832 KB	Google Chrome	
chrome.exe	gearansa	00	30.964 KB	Google Chrome	
devcpp.exe	gearansa	00	30.716 KB	Dev-C++ IDE	
dwm.exe	gearansa	00	26.036 KB	Administrador de ventanas del escritorio	
WINWORD.EXE	gearansa	00	25.428 KB	Microsoft Office Word	
POWERPNT.EXE	gearansa	01	22.476 KB	Microsoft Office PowerPoint	
chrome.exe	gearansa	00	22.212 KB	Google Chrome	
chrome.exe	gearansa	00	19.916 KB	Google Chrome	
HPPA_Main.exe	gearansa	00	16.752 KB	HP Power Assistant	
PrivacyIconClient.exe	gearansa	00	12.728 KB	Intel(R) Management and Security Status	
HPWA_Main.exe	gearansa	00	12.664 KB	HP Wireless Assistant	
chrome.exe	gearansa	00	11.220 KB	Google Chrome	
chrome.exe	gearansa	00	9.192 KB	Google Chrome	
csrss.exe		01	6.620 KB		
sttray.exe	gearansa	00	5.992 KB	IDT PC Audio	
BTTray.exe	gearansa	00	3.524 KB	Bluetooth Tray Application	
SynTPEnh.exe	gearansa	00	3.120 KB	Synaptics TouchPad Enhancements	
LightScribeControlPanel.exe	gearansa	00	3.096 KB	LightScribeControlPanel.exe	
SnippingTool.exe	gearansa	00	3.088 KB	Recortes	
		--	--		

### 3.1.3 Facilidades que aporta un SO

---

- Es difícil definir una lista cerrada y completa de facilidades que aporta un sistema operativo. En la siguiente lista no pretendemos ser exhaustivos, aunque deberíamos ser capaces de señalar las principales funciones que aporta cualquier (o al menos algunos) SO
-



## 3.1.3 Facilidades que aporta un SO

---

- ❑ Interfaz de usuario
    - Interfaz de línea de mandatos
    - Interfaz gráfica de usuario
  - ❑ El kernel (o núcleo)
    - Ejecución de programas
    - Interrupciones
    - Modos
    - Gestión de memoria
    - Memoria Virtual
    - Multitarea
    - Acceso a discos y sistema de ficheros
    - Drivers (controladores) de dispositivos
  - ❑ Funciones de red
  - ❑ Seguridad
-

## 3.1.3 Interfaz de usuario

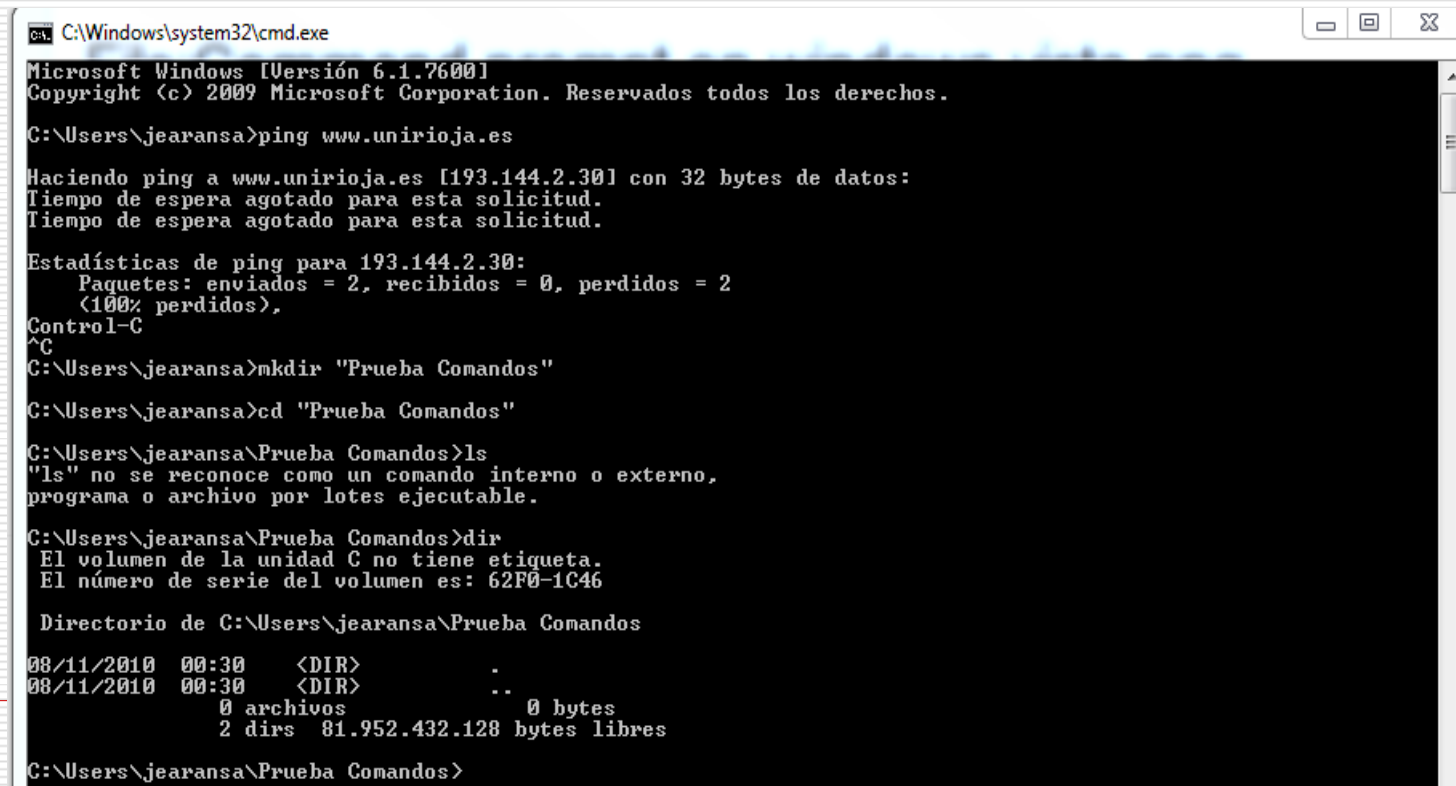
---

- Cualquier dispositivo que interactúa con seres humanos requiere una interfaz de usuario. Generalmente son de dos tipos:
    - Interfaz de línea de mandatos (Command Line Interface, CLI)
    - Interfaz gráfica de usuario (Graphical User Interface, GUI)
-

## 3.1.3 Interfaz de línea de mandatos

---

Es un mecanismo para interactuar con el sistema operativo o con el software de un ordenador por medio de la escritura de comandos que llevan a cabo tareas específicas



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\jearansa>ping www.unirioja.es

Haciendo ping a www.unirioja.es [193.144.2.30] con 32 bytes de datos:
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 193.144.2.30:
    Paquetes: enviados = 2, recibidos = 0, perdidos = 2
              (100% perdidos),
Control-C
^C
C:\Users\jearansa>mkdir "Prueba Comandos"

C:\Users\jearansa>cd "Prueba Comandos"

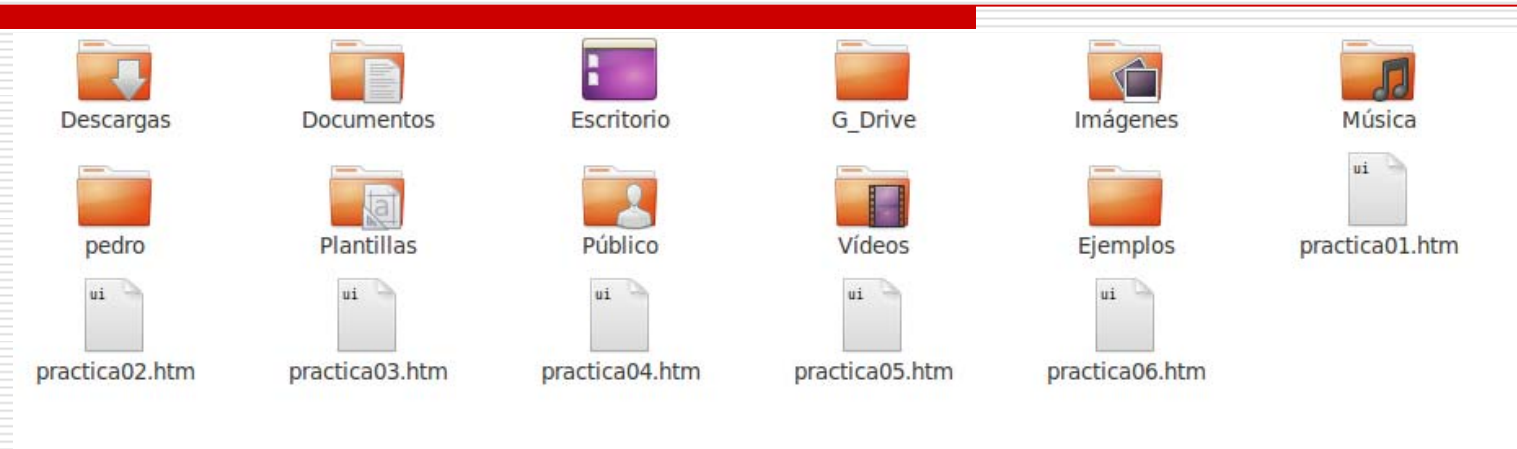
C:\Users\jearansa\Prueba Comandos>ls
"ls" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\jearansa\Prueba Comandos>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 62F0-1C46

Directorio de C:\Users\jearansa\Prueba Comandos
08/11/2010  00:30    <DIR>          .
08/11/2010  00:30    <DIR>          ..
                0 archivos                0 bytes
                2 dirs  81.952.432.128 bytes libres

C:\Users\jearansa\Prueba Comandos>
```

## 3.1.3 Interfaz de línea de mandatos



Para poder borrar todos los ficheros de nombre practica\_\_.\_\_ desde la línea de mandatos

```
jesus@jesus-laptop:~$ rm practica*.*
```

Para algunas tareas repetitivas, es más rápido (y por supuesto consume menos recursos) que la interfaz gráfica

## 3.1.3 Interfaz gráfica de usuario (GUI)

- Tipo de interfaz de usuario que permite interactuar no sólo escribiendo mandatos (interfaces para ordenadores, reproductores de música o vídeo...), sino por medio de otros dispositivos, como ratón, táctiles...



Suelen estar basadas en el uso de iconos (para representar carpetas, ficheros, programas)

### 3.1.3 Interfaz gráfica de usuario (GUI)

---

Su gran ventaja es la facilidad de uso y la rapidez de aprendizaje, aparte de la mejor apariencia visual



## 3.1.3 Kernel

---

- Es la parte central de los SO. Constituye el puente entre el software de aplicaciones y el procesamiento de datos al nivel de hardware. También se encarga de gestionar procesos, uso de memoria, acceder al sistema de ficheros...

Veamos algunas de esas tareas de forma más detallada

---

## 3.1.3 Kernel: ejecución de programas

- Cada vez que se ejecuta un programa en el ordenador, el kernel crea uno o varios procesos, asigna memoria y otros recursos, establece una prioridad para el proceso, carga el código del programa a memoria y finalmente ejecuta el programa

Nombre de imagen	CPU	Espacio de trabajo (memoria)	Espacio máximo de trabajo (memoria)	Tamaño de asignación	Línea de comandos	Descripción
VirtualBox.exe	02	84.064 KB	94.128 KB	121.032 KB	C:\PROGRA~1\Orade\VIRTUA~1\Virtu...	VirtualBox GUI
chrome.exe	00	85.800 KB	129.200 KB	66.380 KB	"C:\Users\jearansa\AppData\Local\Go...	Google Chrome
chrome.exe	00	58.696 KB	77.044 KB	49.552 KB	"C:\Users\jearansa\AppData\Local\Go...	Google Chrome
explorer.exe	01	62.776 KB	77.300 KB	46.796 KB	C:\Windows\Explorer.EXE	Explorador de Windows
chrome.exe	00	42.500 KB	73.864 KB	43.924 KB	"C:\Users\jearansa\AppData\Local\Go...	Google Chrome
WINWORD.EXE	00	45.564 KB	55.064 KB	40.592 KB	"C:\Program Files\Microsoft Office\OFF...	Microsoft Office Word











## 3.1.3 Kernel: Interrupciones

---

- ❑ Las interrupciones permiten al sistema operativo interactuar con su entorno de forma más eficiente. Son gestionadas por el kernel
  - ❑ Una interrupción es una señal por la que el software o hardware del ordenador avisa al SO de que ha completado una tarea (p. ej. leer un fichero) para que éste pueda llevar a cabo su siguiente acción (p. ej. copiarlo a un USB)
  - ❑ Antes de que existieran las interrupciones el SO debía comprobar si una tarea estaba completa (de forma cíclica), para poder comenzar la siguiente. La pérdida de tiempo y recursos era considerable
-

## 3.1.3 Kernel: Interrupciones

---

- ▲  Solicitud de interrupción (IRQ)
  - .....  (ISA) 0x00000000 (00) Cronómetro del sistema
  - .....  (ISA) 0x00000001 (01) Standard 101/102-Key or Microsoft Natural PS/2 Keyboard with HP QLB
  - .....  (ISA) 0x00000008 (08) Sistema CMOS/reloj en tiempo real
  - .....  (ISA) 0x0000000C (12) Synaptics PS/2 Port TouchPad
  - .....  (ISA) 0x0000000D (13) Procesador de datos numéricos
  - .....  (ISA) 0x00000016 (22) RICOH SmartCard Reader
  - .....  (ISA) 0x00000017 (23) HP Mobile Data Protection Sensor

## 3.1.3 Kernel: modos

---

- El kernel gestiona el modo supervisor y el modo protegido
    - Modo supervisor: usado para tareas de bajo nivel que requieren acceso ilimitado al hardware (lectura y escritura directa a memoria, comunicación con la tarjeta gráfica). La BIOS, por ejemplo, se ejecuta en modo supervisor
    - Modo protegido: el kernel, cuando inicia un proceso, puede decidir que el mismo se ejecute en modo protegido, prohibiendo el acceso del mismo a las órdenes de la CPU
-

## 3.1.3 Kernel: gestión de memoria

---

- ❑ El kernel debe asegurar que la memoria principal (o RAM) en uso por cada programa no sea usada simultáneamente por ningún nuevo programa
  - ❑ Modo de funcionamiento (en breve): la memoria principal se segmenta en fragmentos, cada uno de los cuales es accesible sólo por un programa. Si un programa trata de acceder a un fragmento que no es el suyo, se producirá una “violación de segmentación”, y el kernel terminará el programa “invasor”
-

## 3.1.3 Kernel: gestión de memoria

Nombre de imagen	Espacio de trabajo (memoria)	Espacio máximo de trabajo (memoria)
VirtualBox.exe	84.072 KB	94.128 KB
chrome.exe	88.792 KB	129.200 KB
chrome.exe	66.832 KB	78.540 KB
explorer.exe	64.928 KB	77.300 KB
chrome.exe	42.804 KB	47.424 KB
chrome.exe	42.500 KB	73.864 KB
WINWORD.EXE	45.936 KB	55.064 KB
devcpp.exe	35.332 KB	46.112 KB
chrome.exe	33.628 KB	46.696 KB
HPPA_Main.exe	39.076 KB	44.328 KB
dwm.exe	32.124 KB	87.276 KB
VirtualBox.exe	25.368 KB	25.368 KB
PrivacyIconClient.exe	19.732 KB	56.240 KB
HPWA_Main.exe	30.816 KB	36.020 KB
chrome.exe	22.180 KB	32.432 KB
csrss.exe	34.992 KB	88.804 KB
sttray.exe	12.124 KB	14.476 KB
VBoxSVC.exe	9.364 KB	9.384 KB
POWERPNT.EXE	5.968 KB	50.276 KB

## 3.1.3 Kernel: memoria virtual

---

Además de segmentar la memoria, el kernel puede elegir zonas de memoria para ser usadas por varios programas (pero no simultáneamente, recordar la gestión de memoria)

La memoria se “pagina”, y estas páginas, accesibles para diversos procesos, se conocen como “memoria virtual”

Si un proceso trata de acceder a una página de memoria virtual usada por otro proceso se produce una interrupción y el kernel, posiblemente, le asignará una nueva página

---

## 3.1.3 Kernel: multitarea

---

- ❑ En los ordenadores con una CPU, sólo una tarea o proceso se puede estar ejecutando en un momento (en cada ciclo de reloj o Hertzio; de ahí los procesadores de "2,4GHz"...)
  - ❑ La multitarea, gestionada por el kernel por medio de un "scheduler" (planificador), da apariencia de que varios procesos se ejecutan a la vez
  - ❑ El planificador determina cuánto tiempo va a pasar ejecutándose cada programa y da acceso a cada uno (secuencialmente) para usar la CPU y la memoria
-

### 3.1.3 Kernel: acceso a discos y sistema de ficheros

---

- ❑ El acceso a datos guardados en dispositivos de entrada o salida (discos duros, usb's, discos externos) es una de las misiones centrales de los discos duros
  - ❑ Los ficheros permiten acceso más rápido (recordad las tablas FAT), mayor fiabilidad, y mejor uso del espacio libre
  - ❑ Los ficheros, además, permiten otras propiedades como asignar nombres, programas de apertura, compresión, permisos para usuarios...
  - ❑ A esta forma de organización se la conoce como "sistema de ficheros", y suele estar basada en un árbol de directorios
-



## 3.1.3 Kernel: acceso a discos y sistema de ficheros



A screenshot of a file manager window. At the top, there are two tabs labeled 'jesus' and 'pedro'. Below the tabs is a table with four columns: 'Nombre', 'Tamaño', 'Tipo', and 'Fecha de modificación'. The table lists various system folders like 'Descargas', 'Documentos', 'Escritorio', 'G\_Drive', 'Imágenes', 'Música', 'pedro', 'Plantillas', 'Público', 'Vídeos', and 'Ejemplos'. Each folder has a corresponding icon, size (e.g., '0 elementos'), type ('Carpeta'), and modification date. The 'Ejemplos' folder is highlighted, showing a size of '179 bytes' and a type of 'Archivo de configuración del escritorio'.

Nombre	Tamaño	Tipo	Fecha de modificación
+ Descargas	0 elementos	Carpeta	jue 04 nov 2010 17:24:39 CET
+ Documentos	2 elementos	Carpeta	vie 05 nov 2010 13:40:11 CET
+ Escritorio	0 elementos	Carpeta	jue 04 nov 2010 17:24:39 CET
+ G_Drive	0 elementos	Carpeta	vie 05 nov 2010 12:15:53 CET
+ Imágenes	0 elementos	Carpeta	jue 04 nov 2010 17:24:39 CET
+ Música	0 elementos	Carpeta	jue 04 nov 2010 17:24:39 CET
+ pedro	0 elementos	Carpeta	vie 05 nov 2010 12:04:07 CET
+ Plantillas	0 elementos	Carpeta	jue 04 nov 2010 17:24:39 CET
+ Público	0 elementos	Carpeta	jue 04 nov 2010 17:24:39 CET
+ Vídeos	0 elementos	Carpeta	jue 04 nov 2010 17:24:39 CET
Ejemplos	179 bytes	Archivo de configuración del escritorio	jue 04 nov 2010 14:22:42 CET

```
drwx----- 3 jesus jesus 4096 2010-11-05 11:41 .thumbnails
drwx----- 2 jesus jesus 4096 2010-11-04 17:25 .update-notifier
-rw-r----- 1 jesus jesus    5 2010-11-08 00:48 .vboxclient-clipboard.pid
-rw-r----- 1 jesus jesus    5 2010-11-08 00:48 .vboxclient-display.pid
-rw-r----- 1 jesus jesus    5 2010-11-08 00:48 .vboxclient-seamless.pid
drwxr-xr-x  2 jesus jesus 4096 2010-11-04 17:24 Vídeos
-rw-----  1 jesus jesus 2608 2010-11-08 02:59 .xsession-errors
-rw-----  1 jesus jesus 4163 2010-11-05 14:55 .xsession-errors.old
```

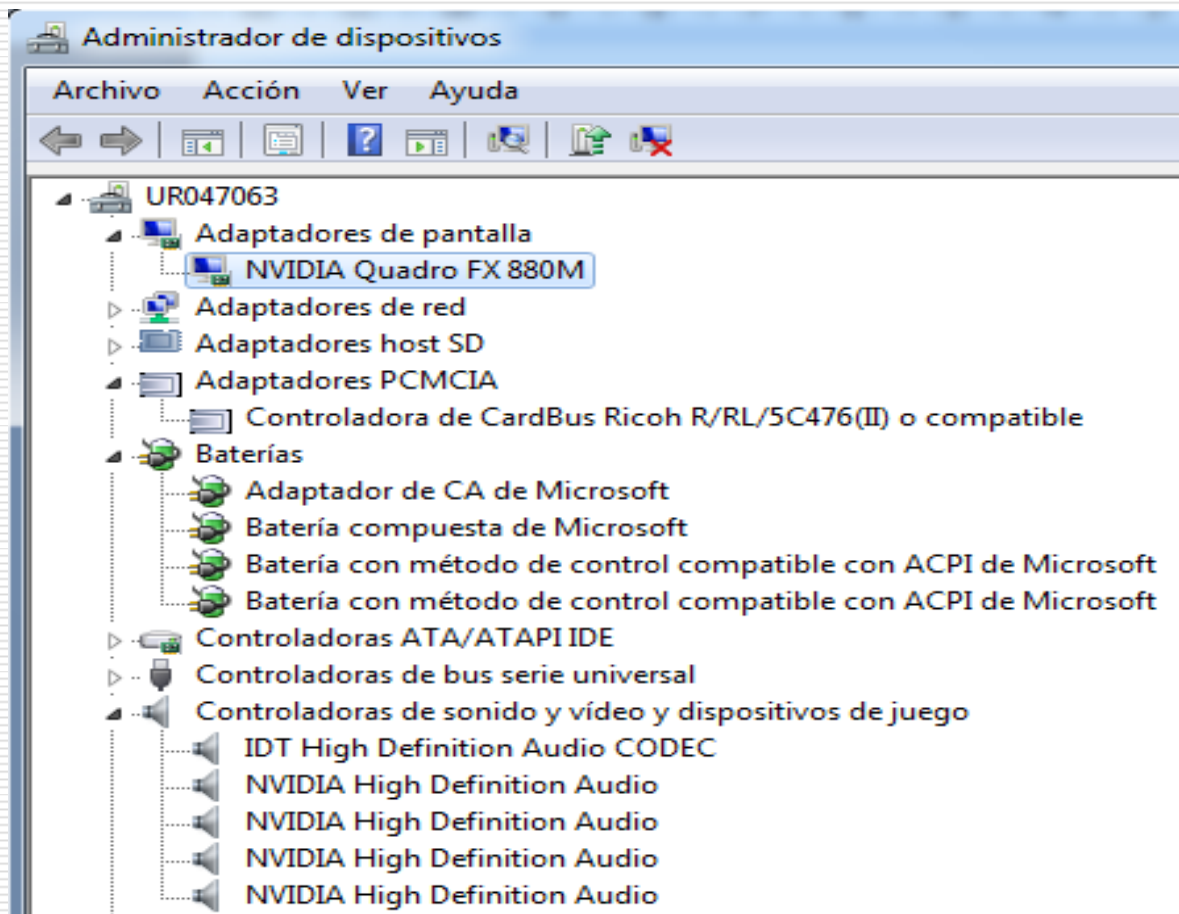
### 3.1.3 Kernel: controladores de dispositivos

---

- ❑ Los controladores de dispositivos (drivers) son programas que permiten la interacción con determinado hardware
  - ❑ Permiten la interacción del kernel con los dispositivos, y proveen del sistema de interrupciones necesario para la comunicación
-

## 3.1.3 Kernel: controladores de dispositivos

---



Los dispositivos tienen su controlador para que el kernel pueda comunicarse con ellos. Generalmente, cada "tipo" de dispositivos tiene un driver genérico que se debe aplicar a cada dispositivo

### 3.1.3 Funciones de red

---

- ❑ La mayor parte de los SO soportan una variedad de protocolos de red, hardware, y aplicaciones para usarlos (Tema 2)
  - ❑ Por medio de funciones de red el SO puede acceder a recursos de ordenadores remotos y usarlos como si fueran locales (práctica 8)
-

## 3.1.3 Seguridad

---

- ❑ El SO, estando conectado a Internet o redes locales, debe ser capaz de distinguir las peticiones que recibe que deben ser procesadas y las que no; por ejemplo, por medio del uso de usuarios y contraseñas, del uso de modos de ejecución (privilegiado o protegido)...
-

## 3.1.4 Estructura de un SO

---

Los sistemas operativos, según su estructura, se suelen dividir en:

- Sistemas operativos monolíticos
- Sistemas operativos distribuidos

También mencionaremos, por el uso que haremos de ellas, la estructura de las máquinas virtuales

---

## 3.1.4 Sistema Operativo Monolítico

---

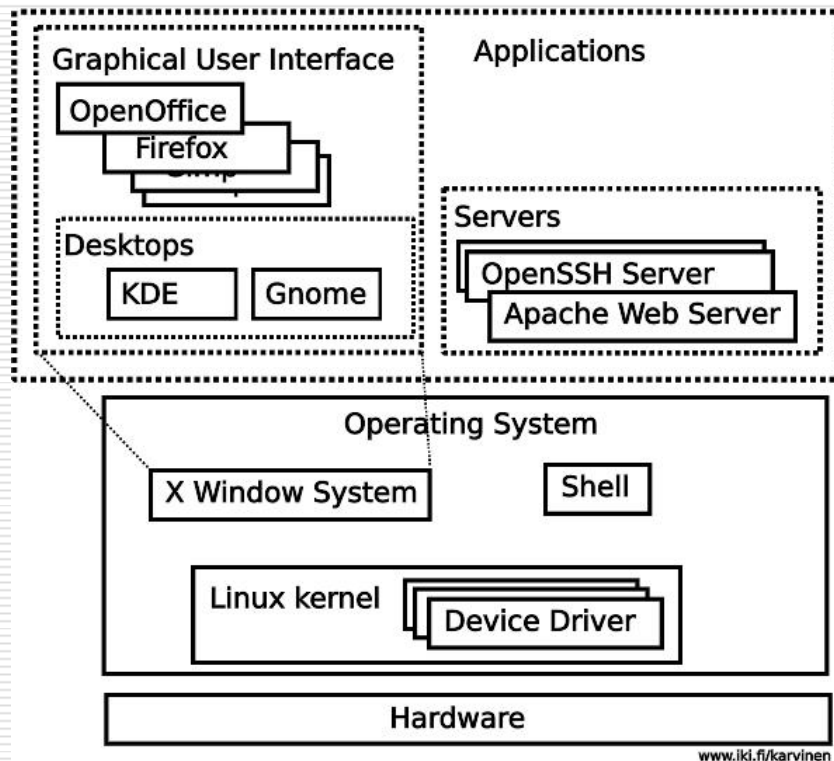
Características:

- ☐ No tienen estructura definida
  - ☐ Todos los componentes (gestión de I/O, de usuarios, procesos, memoria) se realizan por el mismo programa (el SO)
  - ☐ Todas las funciones se ejecutan en modo privilegiado
-

## 3.1.4 Sistema operativo monolítico

Ejemplos:

- ❑ MSDOS
- ❑ Unix, Linux (el kernel hace de pieza que controla todas las funciones del SO)



Empezaron como pequeños sistemas de uso particular, que al popularizarse fueron creciendo de forma no estructurada

Estructura del SO Linux (centrada en el kernel)



## 3.1.4 Sistema operativo monolítico

---

### Desventajas

- ❑ Difícil añadir o modificar funciones
- ❑ Compuestos por miles o millones de líneas en un (o en pocos) programas
- ❑ No hay ocultación de la información (siempre modo privilegiado)

Para hacerlos mas útiles, y más fáciles de mantener y desarrollar, debemos dotarlos de algo de estructura

---

## 3.1.4 Sistema operativo estructurado

---

Sólo nos detendremos en la arquitectura cliente/servidor. Características:

- ❑ La mayor parte de operaciones se ejecuta como procesos de modo protegido (llamados servicios)
  - ❑ Sólo algunas acciones se ejecutan en modo privilegiado (micronúcleo)
  - ❑ El micronúcleo suele encargarse de gestión de interrupciones, procesos, memoria y comunicación básica entre procesos
-

## 3.1.4 Sistema operativo estructurado

---

### Funcionamiento

- ❑ Las peticiones del usuario se entienden como clientes de los servicios que presta el SO. Cada servicio puede demandar servicios adicionales
  - ❑ Los servicios modularizan la estructura del SO y permiten que los fallos en un servicio sólo afecten a ese módulo
-

### 3.1.4 Sistema operativo estructurado

---

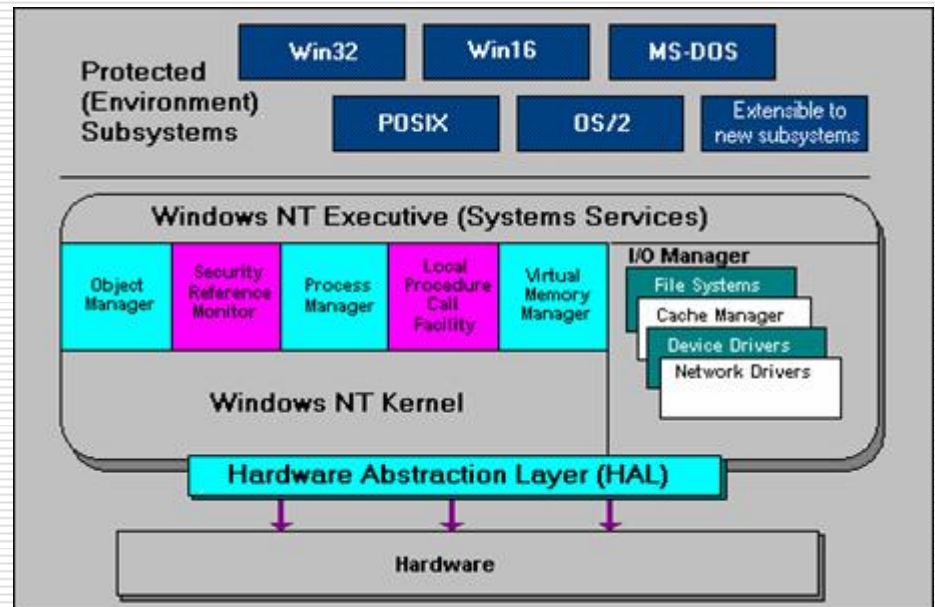
Inconvenientes:

- ❑ Mayor sobrecarga en el tratamiento de los servicios, ya que cada servicio se ejecuta en un espacio distinto
-

## 3.1.4 Sistema operativo estructurado

---

- Ejemplos:
  - Windows NT
  - Minix



Estructura de Windows NT (observar los Servicios de Sistema, "System Services", propios de la arquitectura cliente/servidor)

## 3.1.4 Máquinas virtuales

---

Consisten en una implementación por software de una máquina que ejecuta instrucciones como si fuera una máquina física. Pueden ser de dos tipos:

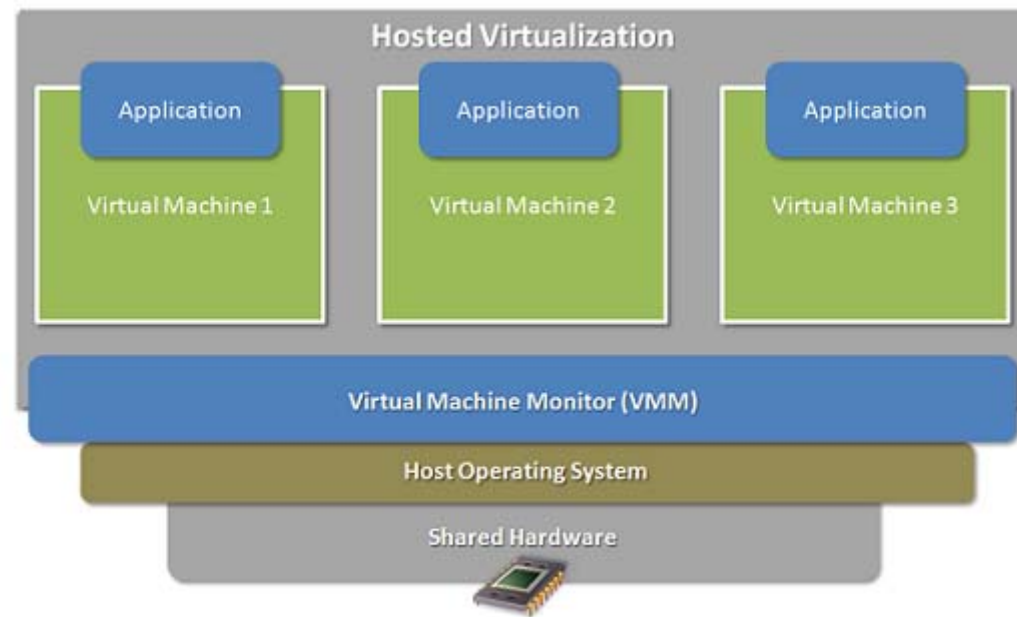
- Virtualizaciones de hardware
  - Virtualizaciones de aplicaciones
-

## 3.1.4 Virtualización de hardware

---

- ❑ Permiten compartir los recursos físicos de la máquina subyacente entre diferentes máquinas virtuales (cada una con su SO)
- ❑ “Esconde” el hardware original, permitiendo usar las máquinas virtuales creadas
- ❑ Cada máquina virtual se ejecuta sobre un “monitor”

Ejemplos:  
VMware, Virtual PC,  
VirtualBox



## 3.1.4 Virtualización de aplicaciones

---

- ❑ Una máquina virtual de aplicación permite “esconder” las características propias del SO, haciendo que dicha aplicación se ejecute directamente sobre la máquina virtual
  - ❑ Permite la portabilidad de la aplicación a distintos SO
  - ❑ Ejemplos: JVM (Java Virtual Machine), Flash Player
-



## 3.1.5 Gestión de memoria y recursos por el SO

---

El gestor de memoria se encarga de:

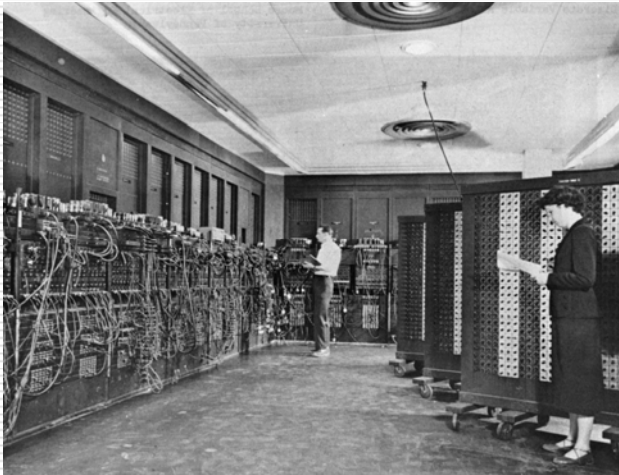
- ☐ Asignar memoria a procesos para crear su imagen de memoria
  - ☐ Proporcionar o liberar memoria a los procesos cuando así lo requieran
  - ☐ Tratar errores de accesos a memoria (fallos de segmentación...)
  - ☐ Permitir que los procesos puedan compartir memoria y comunicarse
  - ☐ Gestionar la jerarquía de memoria y tratar fallos en las memorias virtuales (paginación)
-

## 3.2 Algunos ejemplos de SO

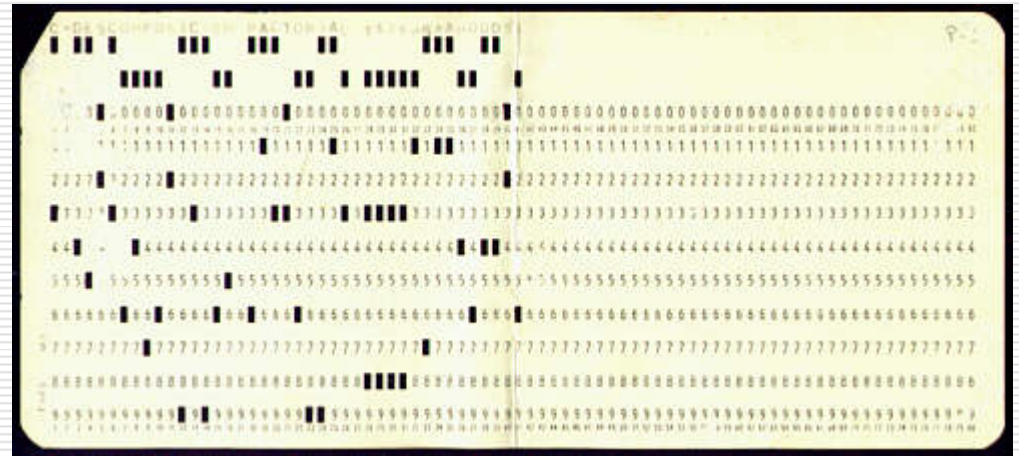
---

Un poco de historia sobre los SO:

- ❑ 1940 – 1950: Primeros ordenadores (ENIAC, EDVAC). No disponen de SO. Los programas se codifican en instrucciones máquina. La entrada era por tarjetas perforadas, la salida por cinta de papel



Computadora ENIAC (1946)



Tarjetas perforadas

## 3.2 Un poco de historia sobre los SO:

---

1950 – 1960. Algunas características de los SO:

- Procesaban un único flujo de trabajo en lotes
  - Disponían de rutinas de E/S
  - Usaban mecanismos que facilitaban pasar de un trabajo a otro
  - Permitían la recuperación del sistema si un trabajo no concluía satisfactoriamente
-

## 3.2 Un poco de historia sobre los SO:

---

1960 – 1970: empiezan a surgir algunas de las ideas presentes en los ordenadores actuales:

- Multiprocesadores
  - Operaciones de E/S por acceso directo a memoria, o DMA (sin intervención de la CPU)
  - Primeros sistemas de tiempo compartido (permitiendo trabajar a varios usuarios simultáneamente)
  - Sistemas OS/360 de IBM, MULTICS de IBM y BELL Laboratories
-

## 3.2 Un poco de historia sobre los SO:

---

1970 – 1980: aparición de los primeros SO de propósito general:

- Primera versión disponible de Unix (programado en C, sólo un pequeño núcleo en ensamblador)
  - Las distintas versiones de Unix (como BSD, 1982) incluyen ideas como la memoria virtual o sockets para aplicaciones TCP/IP
-

## 3.2 Un poco de historia sobre los SO:

---

1980 - ...: Múltiples mejoras,  
incluyendo:

- Máquinas virtuales que simulan hardware
  - Bases de datos sustituyendo ficheros
  - Aparición de los gestores de ventanas
  - GNU/Linux, MSDOS, Windows ..., MAC
-

## 3.2 Unix y las distribuciones Linux

---

Unix nació en los años setenta con el propósito de implementar gestión de

- procesos
- memoria
- ficheros
- dispositivos externos

dentro del núcleo del SO (pero con la mínima funcionalidad para permitir múltiples políticas para cada módulo)

---

## 3.2 Unix y las distribuciones Linux

---

De este modo, generaron un kernel minimal y eficiente (en vez de un SO modular)

En la actualidad el sistema ha evolucionado a:

- ☐ permitir multiprocesadores y hardware distribuido
- ☐ soportar protocolos de red, dispositivos gráficos

El kernel Unix se ha vuelto complejo y bastante grande, pero sigue manteniendo la estructura monolítica

---



## 3.2 Unix y las distribuciones Linux

---

Desde su aparición en 1991, Linux se ha caracterizado por:

- ser una implementación robusta de Unix
  - ser libre (aunque haya distribuciones de pago), General Public License (GPL)
  - ser una implementación significativa del conjunto de llamadas al SO POSIX
  - estar desarrollado por un grupo de colaboradores que trabajan de forma desinteresada
-

## 3.2 Unix y las distribuciones Linux

---

Linux se caracteriza también por disponer de un núcleo monolítico

GNU nace (años 80) del propósito de desarrollar programas libres. Dentro de esta ambición, se desarrollaron editores de texto (emacs, vi), terminales (bash), compiladores (g++, gcc), visores de imágenes (gimp)...

De la conjunción del kernel Linux y de las aplicaciones GNU nació el SO GNU/Linux

---

## 3.2 Unix y las distribuciones Linux

---

GNU/Linux se puede conseguir a través de programas libres:

- El kernel (<http://www.kernel.org/>)
  - Las aplicaciones GNU (<http://www.gnu.org/software/software.es.html>)
  - El sistema de ventanas (<http://www.xfree86.org/>)
  - El entorno gráfico KDE (<http://www.kde.org/>)
  - El entorno gráfico GNOME (<http://www.gnome.org/>)
  - ...
-

## 3.2 Unix y las distribuciones Linux

---

O también a partir de distribuciones con todos los anteriores programas ya disponibles (y probados juntos):

■ Slackware (<http://www.slackware.com>), Debian (<http://www.debian.org/>) sólo admiten software o programas libres

■ Redhat (<http://www.redhat.com/>), SUSE (<http://www.novell.com/linux/>), Mandriva (<http://www2.mandriva.com/es/>), Ubuntu (<http://www.ubuntu.com/>) admiten programas y complementos propietarios

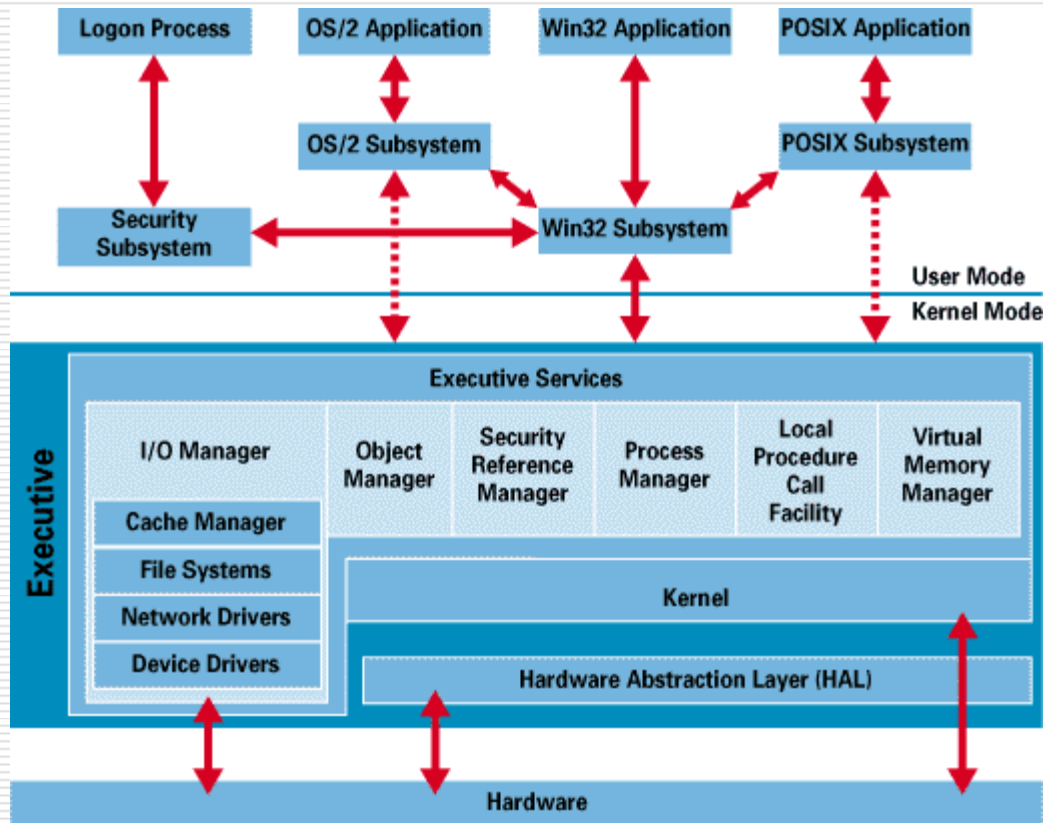
---

## 3.2 Windows

---

- ❑ Su primera versión (Windows 1.0) se remonta al año 1985
  - ❑ En el año 1995 con Windows 95 se rediseña la interfaz gráfica
  - ❑ Windows NT 3.1 (1993) trata de aportar soluciones a negocios (mayor seguridad, multiusuario...)
  - ❑ A partir de Windows XP se fusionan la interfaz de Windows XP y las capacidades de Windows NT
-

## 3.2 Windows



La estructura del sistema se basa en capas (HAL, Kernel, Executive Services) que aportan distintas funcionalidades, y capas de abstracción sobre la parte del Usuario

## 3.2 Mac OS

---

- ❑ Creado por Apple para ordenadores Macintosh
  - ❑ Primer SO con una interfaz gráfica (1984)
  - ❑ A partir de la versión Mac OS X, el SO se deriva de Unix pero manteniendo la interfaz gráfica
-

## 3.2 Android (SO móviles)

- ❑ Basado en una versión modificada del kernel Linux
- ❑ Cada proceso corre una instancia de la máquina virtual Dalvik
- ❑ Cada proceso es una aplicación Java

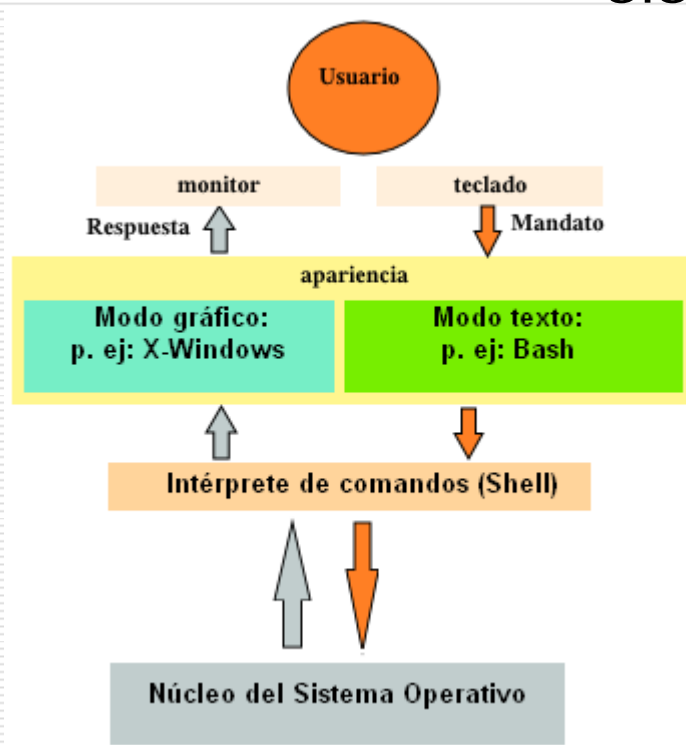




## 3.2 Tipos de interfaces habituales en un SO: “shell”, entorno de ventanas

---

Ejemplo de interacción usuario-máquina en un sistema operativo GNU/Linux



En general, para cualquier SO, siempre habrá una interfaz entre usuario y núcleo (de tipo Command Line Interface, CLI, si es de línea de mandatos, o de tipo GUI, interfaz gráfica de usuario)

## 3.2 Shell o CLI

---

En general una shell (en Linux) nos permitirá realizar muchas de las tareas propias del SO:

- ❑ Definir variables de entorno
  - ❑ Definir alias para las órdenes más habituales
  - ❑ Tiene algunos caracteres especiales, o metacaracteres, que interpreta de manera particular para realizar funciones específicas
  - ❑ Algunas combinaciones de teclas las interpreta de un modo específico
  - ❑ Tiene un lenguaje de programación propio
  - ❑ Interpreta lo que escribimos según la sintaxis establecida y, en caso de que pretendamos ejecutar un comando o programa, lo busca y se lo entrega al núcleo para que lo ponga en ejecución (el shell es el padre de todos los procesos que ejecutes dentro de su entorno)
  - ❑ Dispone de comandos internos (por ejemplo `cd`, `echo`, `set`, ...)
  - ❑ Tiene caracteres y palabras reservadas para usos propios
-

## 3.2 Shell o CLI

---

- ❑ La shell más común de Linux es la conocida como "bash" (es una renovación de "shell", otras disponibles son "c shell", o "csh", Z shell "zsh"...); en Windows disponemos de "cmd"
-

## 3.2 GUI o terminal gráfica

---

- ❑ En Windows es la opción por defecto
  - ❑ En GNU/Linux hay varias capas de software responsables de la interfaz gráfica:
    - El sistema de ventanas X (habitualmente X11), independiente del SO. Se encarga de operaciones gráficas básicas como dibujar objetos, determinar la resolución de pantalla, profundidad de color, etc...
    - El gestor de ventanas, complementa las funcionalidades del sistema de ventanas X porque gestiona los bordes y botones y permite su movimiento, cierre, etc... Hay muchos gestores de ventanas disponibles para X11 entre los que podemos citar kwin (del entorno de escritorio KDE), Metacity (habitualmente usado por GNOME), etc...
    - El entorno de escritorio, es la capa más alta. Da un paso más y añade un gestor gráfico de ficheros para poder arrastrar y soltar, un panel para lanzar aplicaciones y muchas aplicaciones y utilidades propias
-

## 3.2 CLI y GUI

---

Algunos argumentos de la discusión "Command Line Interface" vs "Graphical User Interface":

- ❑ La GUI hace que el SO sea más "amigable" con el usuario  
La GUI hace que el SO sea "vulgarizado"
- ❑ La CLI concede al usuario más control y funciones  
La CLI está anticuada
- ❑ Xwindows es sinónimo de evolución  
Xwindows presenta un gran riesgo de seguridad
- ❑ Usar la GUI es más rápido. Elegir y ejecutar iconos es más rápido que tratar de ejecutar mandatos  
Usar CLI es mucho más rápido. Por medio del teclado podemos trabajar mejor que en una GUI
- ❑ GUI consume demasiada CPU y memoria  
Con el desarrollo actual de los ordenadores, la diferencia no es significativa

Posiblemente, la ventaja está en usar ambas de una forma equilibrada y adecuada

---

## 3.2 Gestión de entrada/salida, dispositivos externos, controladores

---

Una de las tareas fundamentales del SO es la gestión de la comunicación con los dispositivos externos (monitor, teclado, disco duro). Los dispositivos de E/S se pueden agrupar en tres categorías:

- Periféricos: permiten comunicar al usuario con el computador, tanto para entrada (ratón, teclado...) como para salida (impresoras, pantalla)
  - Dispositivos de almacenamiento: proporcionan almacenamiento no volátil de datos (discos, sistemas de ficheros...)
  - Dispositivos de comunicaciones: permiten conectar el ordenador a una red (modem, router, tarjeta de red...)
-

## 3.2 Gestión de entrada/salida, dispositivos externos, controladores

---

- Principal característica: la velocidad de acceso a todos ellos, comparada con la velocidad del procesador (o CPU), es varios órdenes de magnitud más lenta (de  $10^3$  a  $10^6$  veces más lenta)
-

## 3.2 Gestión de entrada/salida, dispositivos externos, controladores

---

El sistema operativo debe controlar el funcionamiento de los dispositivos de E/S para:

- ☐ Facilitar el manejo de periféricos
  - ☐ Optimizar la E/S del sistema
  - ☐ Proporcionar dispositivos virtuales que permitan conectar nuevos dispositivos físicos
-

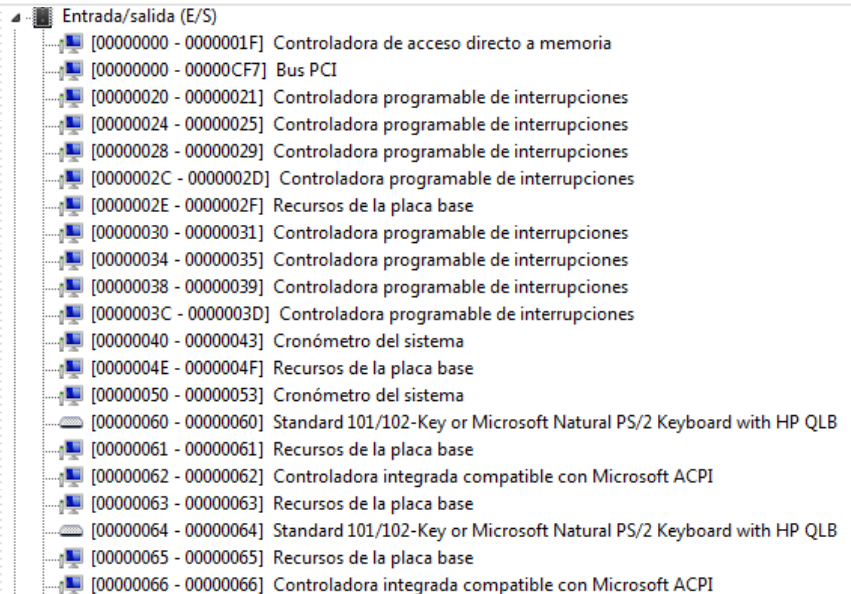


# Tipos de dispositivos de E/S

Dispositivos por puertos:

- Clásico de las arquitecturas Intel
- Cada dispositivo, a través de su controlador, recibe un puerto de E/S, una interrupción hardware y un vector de interrupción

Direcciones de E/S  
asignadas a algunos  
dispositivos de E/S en un  
sistema Windows



# Tipos de dispositivos de E/S

---

## Dispositivos de bloques

- Son los dispositivos de almacenamiento secundario (discos duros) o terciario (sistemas de ficheros) o tarjetas de red que manejan información en unidades de tamaño fijo
  - Se pueden leer y escribir bloques de forma independiente
  - Para discos duros la unidad mínima de transferencia será de 512 bytes, para tarjetas de red 1,5 KB (¿por qué?)
-

# Tipos de dispositivos de E/S

---

```
jesus@jesus-laptop:~$ ls -la /dev | grep br
brw----- 1 root root 7, 0 2008-07-02 12:16 loop0
brw-rw---- 1 root disk 1, 0 2010-11-22 23:07 ram0
brw-rw---- 1 root disk 1, 1 2010-11-22 23:07 ram1
brw-rw---- 1 root disk 1, 10 2010-11-22 23:07 ram10
brw-rw---- 1 root disk 1, 11 2010-11-22 23:07 ram11
brw-rw---- 1 root disk 1, 12 2010-11-22 23:07 ram12
brw-rw---- 1 root disk 1, 13 2010-11-22 23:07 ram13
brw-rw---- 1 root disk 1, 14 2010-11-22 23:07 ram14
brw-rw---- 1 root disk 1, 15 2010-11-22 23:07 ram15
brw-rw---- 1 root disk 1, 2 2010-11-22 23:07 ram2
brw-rw---- 1 root disk 1, 3 2010-11-22 23:07 ram3
brw-rw---- 1 root disk 1, 4 2010-11-22 23:07 ram4
brw-rw---- 1 root disk 1, 5 2010-11-22 23:07 ram5
brw-rw---- 1 root disk 1, 6 2010-11-22 23:07 ram6
brw-rw---- 1 root disk 1, 7 2010-11-22 23:07 ram7
brw-rw---- 1 root disk 1, 8 2010-11-22 23:07 ram8
brw-rw---- 1 root disk 1, 9 2010-11-22 23:07 ram9
brw-rw----+ 1 root cdrom 11, 0 2010-11-22 23:07 scd0
brw-rw---- 1 root disk 8, 0 2010-11-22 23:07 sda
brw-rw---- 1 root disk 8, 1 2010-11-22 23:08 sda1
brw-rw---- 1 root disk 8, 2 2010-11-22 23:07 sda2
brw-rw---- 1 root disk 8, 5 2010-11-22 23:07 sda5
```

Algunos dispositivos de bloque ("b") en la carpeta "dev" ("devices") en un sistema Linux; equivalen a partes de la memoria "ram", del cdrom y de los discos duros y la partición "swap" (sda1, sda2 ... sda5)

---

# Tipos de dispositivos de E/S

---

## Dispositivos por caracteres

- No almacenan información en bloques de tamaño fijo
  - Sirven para representar terminales (tty), impresoras, modems, teclados...
  - Gestionan flujos de caracteres de forma lineal y sin estructura de bloque
-

# Tipos de dispositivos de E/S

---

```
crw-rw----+ 1 root    audio    14,   4 2010-11-22 23:22 audio
crw-----  1 root    dialout   5,   1 2010-11-22 23:22 console
crw-rw----+ 1 root    audio    14,   3 2010-11-22 23:22 dsp
crw-rw-rw-  1 root    root      1,   7 2010-11-22 23:22 full
crw-rw----  1 root    fuse     10, 229 2010-11-22 23:22 fuse
crw-rw----  1 root    root     253,  0 2010-11-22 23:22 hidraw0
crw-rw----  1 root    root     10, 228 2010-11-22 23:22 hpet
crw-r----- 1 root    kmem      1,   2 2008-07-02 12:16 kmem
crw-rw----  1 root    root      1,  11 2010-11-22 23:22 kmsg
crw-r----- 1 root    kmem      1,   1 2010-11-22 23:22 mem
crw-rw----+ 1 root    audio    14,   0 2010-11-22 23:22 mixer
crw-rw-rw-  1 root    root      1,   3 2008-07-02 12:16 null
crw-rw----  1 root    video    195,  0 2010-11-22 23:22 nvidia0
crw-rw----  1 root    video    195, 255 2010-11-22 23:22 nvidiactl
crw-rw----  1 root    root      1,  12 2010-11-22 23:22 oldmem
crw-rw----  1 root    root      1,   4 2010-11-22 23:22 port
crw-r----- 1 root    root     108,  0 2008-07-02 12:16 ppp
crw-rw----  1 root    root      10,   1 2010-11-22 23:22 psaux
crw-rw-rw-  1 root    dialout   5,   2 2010-11-22 23:25 ptmx
crw-rw-rw-  1 root    tty       2, 176 2010-11-22 23:22 ptya0
```

Algunos dispositivos de caracteres ("c") en la carpeta "dev" ("devices") en un sistema Linux; equivalen a dispositivos de audio, de vídeo y tarjeta gráfica, terminales

# Tipos de dispositivos de E/S

---

```
jesus@jesus-laptop:/dev$ sudo fdisk -l
```

```
Disco /dev/sda: 8589 MB, 8589934592 bytes  
255 cabezas, 63 sectores/pista, 1044 cilindros  
Unidades = cilindros de 16065 * 512 = 8225280 bytes  
Identificador de disco: 0x0001446e
```

Disposit.	Inicio	Comienzo	Fin	Bloques	Id	Sistema
/dev/sda1	*	1	993	7976241	83	Linux
/dev/sda2		994	1044	409657+	5	Extendida
/dev/sda5		994	1044	409626	82	Linux swap / Solaris

Las distintas particiones del disco duro en Linux también aparecen como dispositivos por bloques. Como formatos de bloques no son utilizables directamente por las aplicaciones. Debemos “montarlas” en el sistema de directorios de nuestra máquina (`sudo mount -t ext3 /dev/sda1 punto_de_montaje`) para que el sistema pueda acceder a ellas; por medio de la operación de montaje recuperamos su estructura de ficheros y directorios

# Mecanismos de comunicación de E/S

---

- ☐ E/S programada
- ☐ E/S por interrupciones
- ☐ Acceso directo a memoria

(Repasar la sección “Unidades de Entrada/Salida” del Tema 1)

---

## 3.3 Gestión de usuarios y grupos

---

1. Tipos de usuarios más comunes en SO Linux
  2. Utilidad del usuario "root"
  3. Gestión de usuarios desde el entorno de ventanas
  4. Tipos de usuarios más comunes en SO Windows
  5. Gestión de usuarios en SO Windows
-



## 3.3 Gestión de usuarios y grupos

---

- ❑ Una de las funciones del SO es controlar el acceso a un dispositivo (ordenador) y a los recursos del mismo
  - ❑ Esta misión se hace por medio de la gestión de usuarios, grupos y permisos
  - ❑ El SO debe garantizar que sólo los usuarios con permisos adecuados tendrán acceso a los recursos correspondientes
  - ❑ Asimismo, debe asegurar la presencia de al menos un usuario que pueda “administrar” el ordenador (un usuario privilegiado)
-

### 3.3.1 Tipos de usuarios más comunes en sistemas Linux

---

Algunos usuarios comunes en sistemas Linux:

- ☐ Usuario por defecto: en Ubuntu (no en otras distribuciones) al instalar el sistema se crea un usuario que pertenecerá, por defecto, al grupo "admin"
  - ☐ Usuario root: es el administrador del sistema. Puede realizar cualquier acción sobre el sistema (instalación, dar de alta o baja usuarios, modificar privilegios...)
  - ☐ Resto de usuarios: por defecto, sólo tendrán acceso a su directorio (y a recursos accesibles a todo tipo de usuarios)
-

## 3.3.1 Tipos de usuarios más comunes en sistemas Linux

---

### Grupos:

- Cada vez que creamos un usuario, por defecto se crea un grupo al que pertenece el mismo
  - El sistema (al menos Ubuntu) también define una serie de grupos que le permiten definir el acceso a ciertos dispositivos:
    - floppy, cdrom, audio... Solo los miembros de estos grupos tendrán permiso para usar dichos dispositivos
  - Un grupo especial (admin) al que pertenece al menos el primer usuario de la máquina, tiene capacidad de ejecutar aplicaciones como root (o súperusuario)
-

## 3.3.2 Utilidad del usuario root

---

- ❑ El usuario root puede hacer cualquier acción sobre el sistema. En particular, crear y eliminar usuarios, instalar y eliminar aplicaciones, cambiar propietarios y grupos de ficheros, recorrer el árbol de directorios completo, montar nuevos dispositivos...
  - ❑ Utilizarlo para acciones cotidianas es potencialmente peligroso (algunas distribuciones, como Ubuntu, lo tienen bloqueado, lo cual no impide que se pueda desbloquear)
  - ❑ Además, el sistema es mucho más vulnerable mientras estamos logados como "root" (cualquier código malicioso tendría acceso a todo el sistema con privilegios)
  - ❑ Una forma de evitar el uso del usuario "root" es por medio del mandato "sudo". "sudo" permite ejecutar cualquier mandato del sistema como "super user", como si se fuera "root"; sin embargo, siempre nos preguntará la contraseña antes de realizar una acción, asegurando que no hacemos algo por error (o alguien que no sea nosotros)
  - ❑ Dentro del grupo de "sudoers" están por defecto "root" y los miembros del grupo "admin" (lo puedes comprobar en el fichero /etc/sudoers)
-

## 3.3.3 Gestión de usuarios desde la "shell"

---

Algunos mandatos útiles:

- ❑ `adduser nombre`: requiere privilegios de "root" o "sudo"; permite crear un nuevo usuario
- ❑ `userdel usuario`: requiere privilegios de "root" o "sudo"; permite eliminar un usuario
- ❑ `usermod -a -G usuario grupo`: requiere privilegios de "root" o "sudo"; permite añadir un usuario a un grupo
- ❑ `passwd usuario`: requiere privilegios de "root" o "sudo"; permite cambiar la contraseña de cualquier usuario (incluido root)
- ❑ `group usuario`: lista de grupos a que pertenece "usuario"

Algunos directorios:

- ❑ `/etc/passwd` : directorio de usuarios de la máquina
  - ❑ `/etc/group` : directorio de grupos
  - ❑ `/etc/shadow` : directorio de contraseñas (encriptadas)
  - ❑ `/etc/sudoers` : directorio de usuarios que pueden utilizar "sudo"
-

## 3.3.3 Gestión de usuarios desde la "shell"

---

Aspecto del fichero /etc/paswd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
```

Cada fila representa un usuario de la máquina. Contiene información referente al nombre de usuario, contraseña, UID (identificador de grupo), GID (identificador de grupo), datos personales, directorio de inicio de sesión y shell o terminal por defecto

---

## 3.3.3 Gestión de usuarios desde la "shell"

---

Aspecto del fichero /etc/group

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:jesus,pedro
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
```

Cada fila representa un grupo de la máquina. Contiene información referente al nombre de grupo, contraseña, clave de grupo y miembros de cada grupo (aparte del usuario original, que está por defecto)

---

### 3.3.4 Tipos de usuarios más comunes en Windows NT

---

- ☐ En Windows NT, cada vez que abrimos una sesión es necesario emplear un usuario y contraseña
  - ☐ Por defecto, se crea una cuenta "Invitado" (desactivada por defecto) y una cuenta "Administrador"
  - ☐ También existe el tipo de usuario "Usuario", para todos los usuarios que creemos con el Administrador
  - ☐ Es recomendable cambiar el nombre de la cuenta Administrador
-



### 3.3.4 Tipos de usuarios más comunes en Windows NT

---

- ❑ La cuenta Administrador es la que tiene los privilegios de crear y eliminar cuentas, instalar aplicaciones para todos los usuarios...
  - ❑ También existen grupos para clasificar a los usuarios:
    - Administradores
    - Operadores de copia
    - Duplicadores
    - Usuarios Avanzados
    - Usuarios
    - Invitados
  - ❑ El sistema nos obliga a tener siempre al menos un Administrador
-

## 3.3.5 Gestión de usuarios en SO Windows

---

Algunos mandatos útiles (la mayor parte requieren ser administrador):

- `net user`: listado de usuarios
  - `net user usuario /add`: añadir un usuario
  - `net user usuario *`: asignar contraseña
  - `net user usuario /del`: borrar usuario
  - `net localgroup`: listado de grupos
  - `net localgroup grupo`: miembros de grupo
  - `net localgroup grupo usuario /add`: añade usuario a grupo
  - `net localgroup grupo usuario /del`: elimina usuario de grupo
-

## 3.4 Árbol de directorios

---

3.4.1 Árbol de directorios en SO Linux

3.4.2 Montaje de unidades externas

3.4.3 Gestión del árbol de directorios desde la "shell"

3.4.4 Árbol de directorios en Windows

3.4.5 Uso de las distintas particiones

3.4.6 Gestión del árbol de directorios desde "cmd"

---

## 3.4 Árbol de directorios

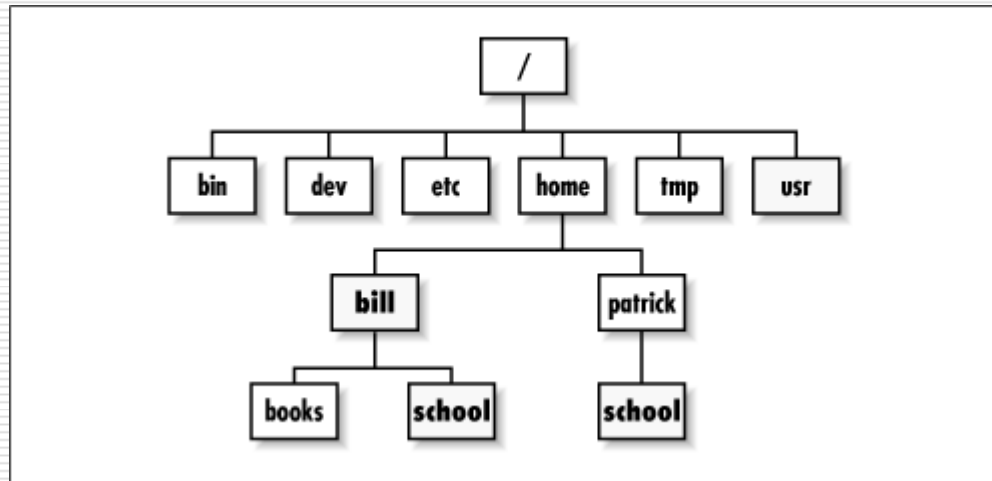
---

- ❑ El árbol de directorios permite estructurar la información que se encuentra en un disco duro, racionalizando su distribución
  - ❑ También permite asignar permisos a los distintos usuarios, facilitando o prohibiendo el acceso a ciertos conjuntos de directorios de forma sencilla
-

## 3.4.1 Árbol de directorios en SO Linux

---

- ❑ La base es un directorio “raíz” del cual “cuelgan” todos los directorios del sistema



### 3.4.1 Árbol de directorios en SO Linux

---

- ❑ Cualquier dispositivo que conectemos (o “montemos”) a nuestro ordenador pasa a compartir este sistema de directorios
  - ❑ Esto requiere una operación de “montaje” del mismo sobre el sistema de directorios
  - ❑ Ventajas: gestión de manera uniforme de todos los dispositivos externos (o de E/S): discos duros, terminales, sistemas de audio o vídeo. Todos ellos son ficheros de bloques o caracteres que se montan en nuestro sistema
-

## 3.4.1 Árbol de directorios en SO Linux

---

Algunas carpetas relevantes:

- /home : contiene los directorios de todos los usuarios (excepto "root")
  - /dev : contiene los ficheros de bloque o carácter de los distintos dispositivos de E/S del ordenador
  - /bin, /sbin, /usr/local/bin, /usr/local/sbin, /usr/local/bin : contienen algunos de los mandatos más usados (mucho propios de GNU)
  - /root : directorio de inicio del "root"
  - /mnt, /media : puntos de montaje, usados por defecto, de dispositivos externos
  - /proc : carpeta de procesos y del kernel (formado por directorios virtuales, es como un "mapping" del procesador)
-

## 3.4.2 Montaje de unidades externas

---

- ❑ Como hemos dicho, la forma de usar dispositivos externos es a través de su montaje
  - ❑ Esto “traduce” el dispositivo externo al sistema de directorios de Linux, incorporándolo en una carpeta del mismo (/mnt, /media...)
  - ❑ Linux es capaz de reconocer algunos de los sistemas de ficheros más comunes: ext2, ext3, ext4, ntfs, vfat, msdos, “auto”...
-



## 3.4.2 Montaje de unidades externas

---

- La operación de “montaje”, siempre y cuando el dispositivo aparezca en la carpeta “/dev” reconocido como un dispositivo de bloques o de caracteres, sólo requiere crear un directorio en nuestro disco duro donde se vaya a montar el dispositivo, y ejecutar el mandato:

```
jesus@jesus-laptop:/dev$ sudo mount -t ext3 /dev/sda1 /home/jesus/Escritorio/disco1/
```

La operación anterior ha “montado” el sistema de ficheros (de tipo ext3) del dispositivo “/dev/sda1” (la partición número 1 del disco duro de Linux!!) en la carpeta /home/jesus/Escritorio/disco1

1. La operación exige que usemos privilegios de administrador (sudo)
2. A partir de este momento tendremos acceso al disco duro Linux a través de “/” o de “/hom/jesus/Escritorio/disco1”

~~La misma operación puede ser llevada a cabo con USB's discos duros externos...~~

### 3.4.3 Gestión del árbol de directorios desde la “shell”

---

Algunos mandatos básicos que nos permiten gestionar el árbol de directorios:

- ❑ `pwd`: “print working directory”, nos da información sobre el directorio en que nos encontramos
-

## 3.4.3 Gestión del árbol de directorios desde la "shell"

Algunos mandatos básicos que nos permiten gestionar el árbol de directorios:

- ❑ ls: listado de los contenidos de una carpeta, o de las propiedades de un fichero
- ❑ Opciones interesantes:
  - -l: "Long format", o formato largo, incluyendo información sobre:
    - ❑ el tipo de fichero ("- " = archivo, "d" = directorio, "l" = link débil, "b" = fichero de bloque, "c" = fichero de cabecera)
    - ❑ los permisos correspondientes (9 caracteres), que pueden ser r = lectura, w = escritura, x = ejecución para el propio usuario, el grupo y el resto de usuarios
    - ❑ el número de ficheros que contiene o el número de enlaces fuertes
    - ❑ el nombre del propietario
    - ❑ el nombre del grupo
    - ❑ el tamaño en bytes
    - ❑ la fecha de la última modificación
    - ❑ el nombre del ficheros

```
drwxr-xr-x 4 jesus jesus    4096 2010-11-24 21:58 practical6
-rw-r--r-- 1 jesus jesus 2017280 2010-11-24 23:47 practical6.tar
-rw-r--r-- 1 jesus jesus  682588 2010-11-25 00:23 practical6.tar.gz
```

### 3.4.3 Gestión del árbol de directorios desde la "shell"

---

- ❑ ls: listado de los contenidos de una carpeta, o de las propiedades de un fichero
  - ❑ Opciones interesantes:
    - -a: todos los ficheros y directorios, incluyendo los ocultos (cuyo nombre empieza por el carácter ".")
    - -i: información sobre los "inodos" o referencias en memoria de los objetos
    - -R: recursivo, muestra los contenidos del directorio y sus subdirectorios
-

## 3.4.3 Gestión del árbol de directorios desde la "shell"

---

Algunos mandatos básicos que nos permiten gestionar el árbol de directorios:

- ❑ `cd`: "change directory", cambiar de directorio. Si no recibe ningún parámetro nos devuelve el nombre del directorio actual. Si recibe un directorio nos traslada al mismo
  - ❑ Diferencia entre rutas relativas y rutas absolutas:
    - Rutas absolutas: aquellas de la forma `"/directorio1/directorio2/fichero"` o `"/directorio1/directorio2/"`; dan información sobre la ruta absoluta de un fichero en el árbol de directorios (es decir, partiendo desde el directorio raíz `"/"`)
      - ❑ Ventaja: son independientes del directorio en que nos encontremos (pero no de la máquina, poca portabilidad)
      - ❑ Desventaja: son más largas, generalmente, que las relativas
    - Rutas relativas: aquellas de la forma `"directorio/fichero"` o `"directorio"`; toman como origen el directorio en que nos encontramos
      - ❑ Ventajas: son más breves y son más portables (por ejemplo en un fichero `tar`)
      - ❑ Pueden ser inestables dentro de una misma máquina
-

## 3.4.3 Gestión del árbol de directorios desde la "shell"

---

Algunos otros mandatos nos permiten:

- ☐ mover fichero a carpeta:
    - `mv fichero carpeta`
  - ☐ renombrar fichero1 a fichero2:
    - `mv fichero1 fichero2`
  - ☐ renombrar carpeta directorio1 a directorio2:
    - `mv directorio1 directorio2`
  - ☐ copiar fichero a carpeta:
    - `cp fichero carpeta`
  - ☐ copiar fichero1 a fichero2:
    - `cp fichero1 fichero2`
  - ☐ copiar directorio1 (y sus contenidos) a directorio2:
    - `cp -r directorio1 directorio2`
-

### 3.4.3 Gestión del árbol de directorios desde la “shell”

---

Algunos otros mandatos que nos permiten:

- ☐ `rmdir`: eliminar un directorio vacío
  - ☐ `rm`: eliminar un fichero
  - ☐ `rm -r`: eliminar un directorio, y de forma recursiva todos sus componentes
-

### 3.4.3 Enlaces simbólicos y enlaces “duros”

---

Los sistemas de ficheros propios de Linux (ext3, ext4) gestionan los objetos en memoria por medio de inodos.

Todos los ficheros y directorios en ext3 o ext4 están representados por un inodo. Los inodos guardan información sobre el tamaño, permisos, propietarios, enlaces y localización en disco del fichero o directorio. Puedes comprobarlo con:

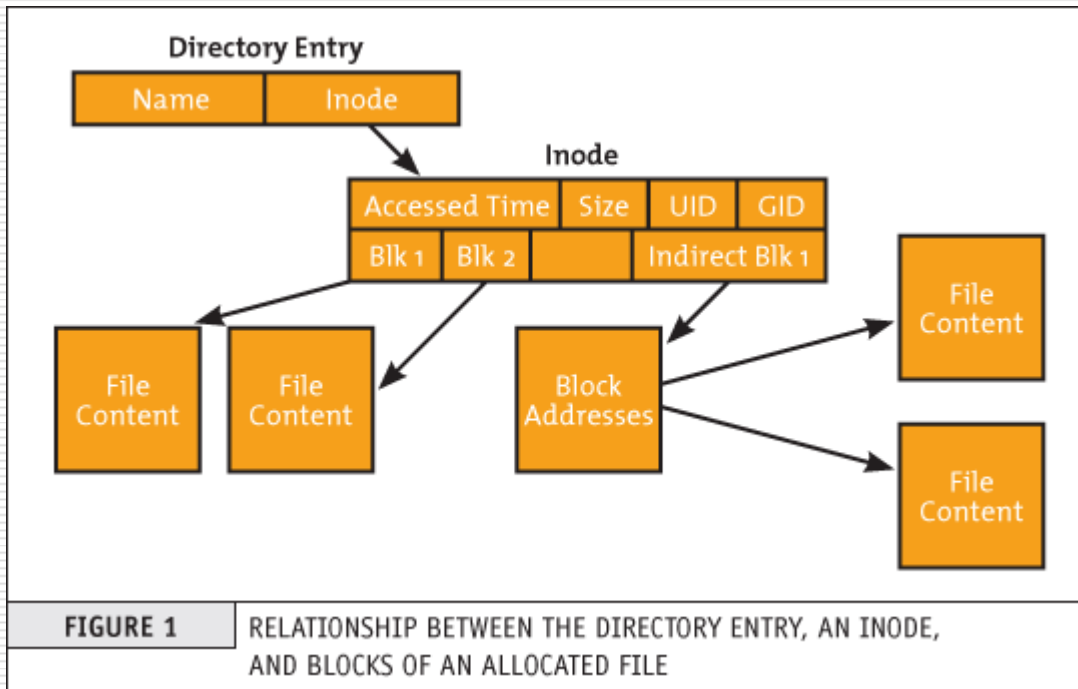
`stat fichero`

`stat directorio`

---



### 3.4.3 Enlaces simbólicos y enlaces “duros”



A cada nombre de fichero le corresponde un inodo (a varios ficheros les puede corresponder el mismo). Este inodo es el que nos permite llegar en memoria a los distintos bloques (Blk1, Blk2, Indirect Blk1) que ocupa el fichero o directorio en memoria

### 3.4.3 Enlaces simbólicos y enlaces “duros”

---

- ❑ Si varios ficheros en memoria están asociados con el mismo inodo, diremos que ese fichero tiene varios enlaces duros o fuertes. Cualquiera de los ficheros nos permitirá acceder a ese fichero en memoria. Si borramos uno de los ficheros pero no los otros, el objeto permanecerá en memoria
  - ❑ Si un fichero contiene sólo la dirección o el nombre de otro fichero, diremos que este fichero es un enlace débil o simbólico al fichero original. Si borramos el fichero original, el enlace simbólico queda roto (e inservible)
-

### 3.4.3 Enlaces simbólicos y enlaces “duros”

---

- ❑ Mandatos para definir enlaces duros:
    - In destino enlace
    - Es posible entre ficheros, no entre carpetas
  - ❑ Mandatos para definir enlaces simbólicos:
    - ln -s destino enlace
    - Válido para crear enlaces a ficheros en una carpeta o enlaces entre carpetas
-

## 3.4.4 Árbol de directorios en Windows

---

### Estructura convencional de un árbol de directorios en Windows



Cada unidad o dispositivo de almacenamiento dispone de su propio árbol de directorios. Esto hace innecesario el montaje de unidades, directamente se incorporan al árbol de directorios como nuevos "caracteres" (E:, F:...). Por el contrario, evita un tratamiento unificado de los ficheros en distintas unidades (permisos, compresión, encriptación)

### 3.4.5 Gestión del árbol de directorios desde “cmd”

---

- ❑ Algunos mandatos para gestionar el árbol de directorios en Windows:
    - `cd` (sin parámetros): devuelve la ruta del directorio en que nos encontramos
    - `cd ruta`: nos permite cambiar de directorio (por medio de rutas absolutas, del tipo `cd C:\Users\...`) o por medio de rutas relativas (`cd Escritorio`)
    - `dir`: lista el contenido de un directorio
-

### 3.4.5 Gestión del árbol de directorios desde "cmd"

---

- ☐ move origen destino: permite mover directorios
  - ☐ rename origen destino: renombrado de carpetas y directorios
  - ☐ copy origen destino: copiar un fichero
  - ☐ xcopy origen destino: copiar una carpeta de forma recursiva (con todos sus contenidos, ficheros y carpetas)
-

## 3.5 El sistema de archivos

---

3.5.1 Necesidad de uso de los archivos

3.5.2 Archivos en SO Linux. Características, propiedades, limitaciones

3.5.3 Archivos en SO Windows.  
Características, propiedades,  
limitaciones

---

## 3.5.1 Necesidad de uso de los archivos

---

- ❑ Los archivos nos permiten agrupar conjuntos de información. En el disco duro toda la información se almacena en base a sectores, pistas... Los archivos nos dan una capa de abstracción que permite omitir esa representación
  - ❑ Para poder disponer de ficheros, en primer lugar debemos "formatear" el disco y asignarle un sistema de ficheros. El sistema de ficheros determinará ciertas propiedades como la información que se guarda sobre cada fichero (nombre, tamaño, permisos), el espacio que el mismo ocupa en memoria (tamaño del clúster...)
-



### 3.5.2 Archivos en SO Linux. Características y limitaciones

---

El tipo de archivos más común en las distribuciones Linux es ext3 (y ext4). Sin embargo, el kernel Linux cuenta con VFS (Virtual Filesystem Switch), una interfaz que le permite gestionar de modo idéntico distintos sistemas de ficheros como ext2, ReiserFS, MSDOS, VFAT, NTFS, HFS, AFS... (y siguen aumentando)

En realidad, es el VFS el que se encarga de gestionar los inodos y algunas de las llamadas básicas a sistema como "stat", "chmod", "chgrp"

---

## 3.5.2 Archivos en SO Linux. Características y limitaciones

---

Algunas características básicas sobre ext3:

- ❑ Máxima longitud de nombre de fichero: 254 caracteres
  - ❑ Máximo tamaño de unidad: de 2 a 16TB (dependiendo del tamaño de bloque)
  - ❑ Máximo de 32000 subdirectorios por directorio
  - ❑ Permisos: los propios de sistemas Linux (r, w, x, s, S)
  - ❑ Permite conversión directa de las particiones ext2 a formato ext3
  - ❑ Dispone de journaling, que permite guardar traza de los cambios realizados sobre un fichero y recuperar versiones originales bajo fallos del sistema
-

## 3.5.2 Archivos en SO Linux. Características y limitaciones

---

Algunas características básicas sobre ext4:

- ☐ Tamaño máximo de archivo: 16TB
  - ☐ Tamaño máximo de sistema:  $10^6$  TB
  - ☐ Máximo número de ficheros: 4 billones
  - ☐ Permisos: los propios de POSIX (o UNIX)
  - ☐ Es compatible con ext3 y ext2
  - ☐ Dispone de journaling
-

## 3.5.2 Archivos en SO Linux. Características y limitaciones

---

Algunas operaciones básicas sobre ficheros:

- ❑ stat: muestra el estatus de un fichero

```
jesus@jesus-laptop:~/Escritorio$ stat enl_debil
  File: «enl_debil» -> «/usr/games/gnomine»
  Size: 18             Blocks: 0          IO Block: 4096   vínculo simbólico
Device: 801h/2049d    Inode: 305435       Links: 1
Access: (0777/lrwxrwxrwx)  Uid: ( 1000/   jesus)   Gid: ( 1000/   jesus)
Access: 2010-11-30 00:25:04.000000000 +0100
Modify: 2010-11-29 00:07:48.000000000 +0100
Change: 2010-11-29 00:07:48.000000000 +0100
```

Entre la información mostrada podemos observar el tamaño del fichero (18 bytes), el tipo de fichero (vínculo simbólico), el dispositivo en que se encuentra almacenado (device), el inodo de memoria que ocupa y cuántos enlaces al mismo hay, los permisos de acceso (en notación numérica u octal y simbólica), el propietario y el grupo, y las fechas de acceso, modificación y cambio del fichero

---

## 3.5.2 Archivos en SO Linux. Características y limitaciones

---

- ❑ `file nombre_de_fichero`: permite conocer el contenido de un fichero (si el mismo es un binario, texto plano, texto con formato html, xml...)

El mandato anterior sobre todo nos da una idea de cómo asocia Linux un fichero con un programa; no depende de la “extensión”, al estilo Windows, sino de su contenido

También se puede notar que dicha asociación no forma parte de la información existente en el sistema de ficheros (observa que “stat” no decía nada al respecto)

---

## 3.5.2 Archivos en SO Linux. Características y limitaciones

---

### Permisos:

Los tipos de permisos en sistemas Linux se dividen en tres tipos: lectura (r), escritura (w) y ejecución (x).

Estos permisos se pueden asignar o quitar al propietario del fichero o carpeta (por defecto, el creador del mismo, "u"), al grupo (por defecto el grupo propio del creador, "g") y al resto de usuarios de la máquina ("o")

---

## 3.5.2 Archivos en SO Linux. Características y limitaciones

---

### Significado de los permisos:

- ❑ r (ó 4 en octal):
    - Para ficheros: permiso de lectura
    - Para directorios: listar el contenido
  - ❑ w (ó 2 en octal):
    - Para ficheros: modificar el fichero
    - Para directorios: crear y borrar ficheros y directorios dentro del directorio
  - ❑ x (ó 1 en octal):
    - Para ficheros: ejecutar ficheros o scripts
    - Para directorios: acceder a un directorio
-

## 3.5.2 Archivos en SO Linux. Características y limitaciones

---

Asignación de permisos:

```
$chmod ug+rw,o+r fichero
```

("+" añade permisos)

```
$chmod a-x fichero
```

("a" abrevia a "todos", "-" quita permisos)

```
$chmod u=rw,o=r,a= fichero
```

("=" asigna exactamente los permisos señalados, y los demás los quita)

La sintaxis se compone del mandato "chmod", seguido de la cadena de caracteres que queremos representar los permisos que queremos asignar, y el fichero o carpeta correspondiente

---



## 3.5.2 Archivos en SO Linux. Características y limitaciones

---

También podemos asignar permisos de forma numérica, con respecto a las siguientes reglas:

Permiso de lectura (r): 4

Permiso de escritura (w): 2

Permiso de ejecución (x): 1

Los permisos de un fichero o carpeta se asignan por medio de tres números (propietario, grupo y otros), dados por la suma de los anteriores valores:

`chmod 652 fichero`

El anterior mandato otorga: lectura y escritura al propietario ( $4 + 2 = 6$ ), lectura y ejecución al grupo ( $4 + 1 = 5$ ), escritura a otros (2).

---

## 3.5.2 Archivos en SO Linux. Características y limitaciones

---

Algunas combinaciones numéricas habituales:

`chmod 755 fichero`

(todos los permisos al propietario; grupo y otros sólo lectura y ejecución)

`chmod 777 fichero`

(todos los permisos para cualquier tipo de usuario)

`chmod 664 fichero`

(el fichero puede ser leído y modificado por el propietario y el grupo, sólo leído por el resto de usuarios, y no puede ser ejecutado)

`chmod 444 fichero`

(el fichero sólo es de lectura)

`chmod 555 fichero`

(sobre una carpeta, podemos abrirla y explorarla, pero no escribir en su interior)

---

## 3.5.2 Archivos en SO Linux. Características y limitaciones

---

### Otros mandatos útiles:

- `chwon usuario fichero(s)`: cambiar el propietario de fichero(s) o directorio(s)
  - `chgrp grupo fichero(s)`: cambiar el grupo de fichero(s) o directorio(s)
  - `umask`: determinar una máscara de tal modo que cualquier fichero o directorio que se cree después reciba los permisos señalados
    - Máscara en letras:  
`umask u=rwx,g=rwx,o=r`
    - Máscara en octal:  
`umask 022`  
(Las máscaras en octal hay que leerlas como el complementario, los permisos asignados a partir de la orden anterior serán 755)
-

### 3.5.3 Archivos en SO Windows.

#### Características y limitaciones

---

- ❑ Los dos tipos de particiones más comunes en Windows son FAT32 y NTFS. Desde las versiones de Windows NT (Windows 2000) el sistema NTFS se impuso por sus mayores prestaciones
-

### 3.5.3 Archivos en SO Windows. Características y limitaciones

---

#### FAT32:

- Tamaño máximo de unidad: 2 - 8TB
  - Tamaño máximo de fichero: 4GB
  - Número máximo de ficheros:  $4 \cdot 10^6$
  - No previene la fragmentación de las unidades
  - No tiene soporte para permisos
-

### 3.5.3 Archivos en SO Windows. Características y limitaciones

---

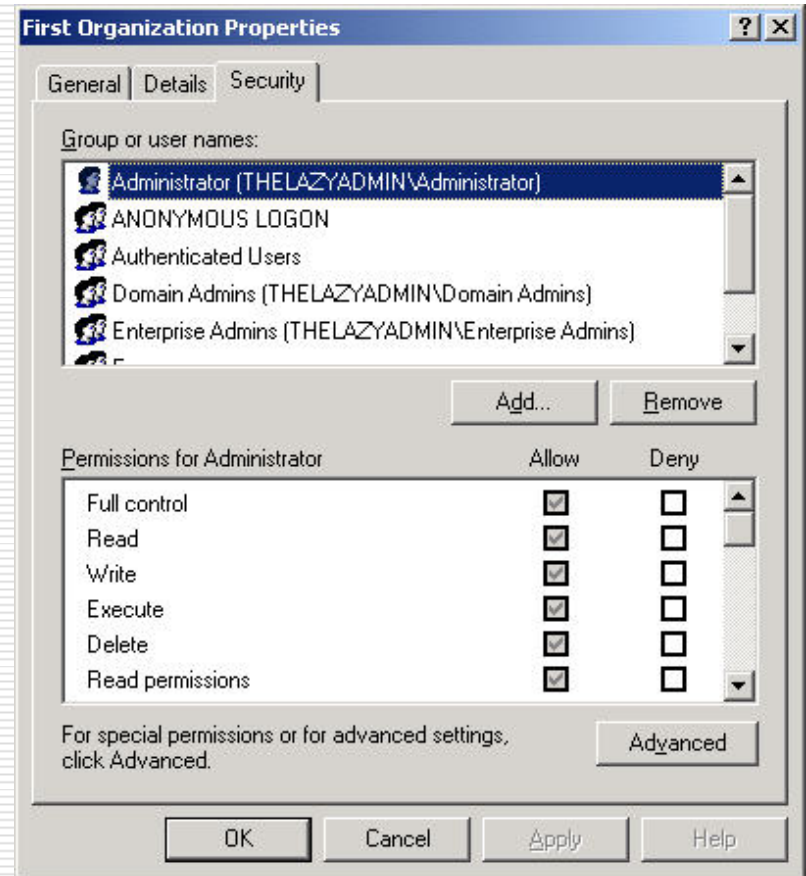
#### □ NTFS

- Longitud de nombre de fichero: 255 caracteres
  - Tamaño de fichero: 16TB
  - Tamaño de unidad: 256TB
  - Máximo número de ficheros:  $4 * 10^9$
  - Permisos: Access Control List (ACL), extensión de POSIX
  - Admite enlaces duros, journaling, compresión, encriptación...
-

## 3.5.3 Archivos en SO Windows. Características y limitaciones

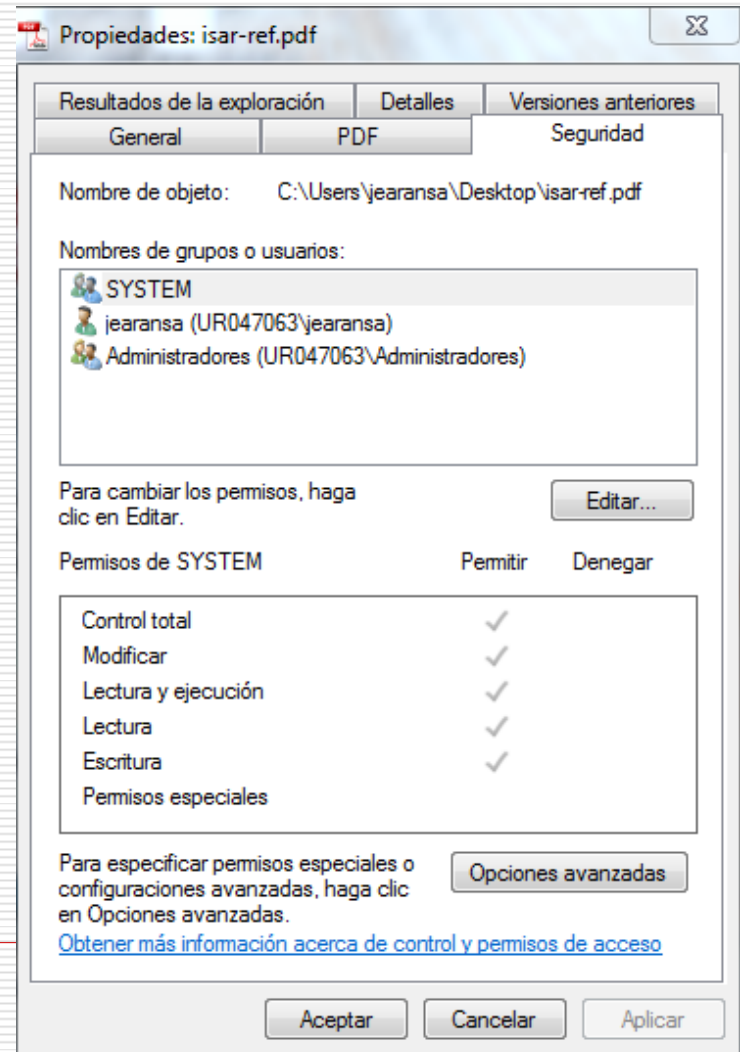
Gestión de permisos en ntfs:

Está basada en listas de control de acceso (ACL), listas de entradas que especifican un usuario y la lista de los derechos de acceso que el mismo tiene sobre el recurso (del tipo “permitir” o “denegar”):



# 3.5.3 Archivos en SO Windows. Características y limitaciones

## Permisos sobre ficheros:





## 3.5.3 Archivos en SO Windows.

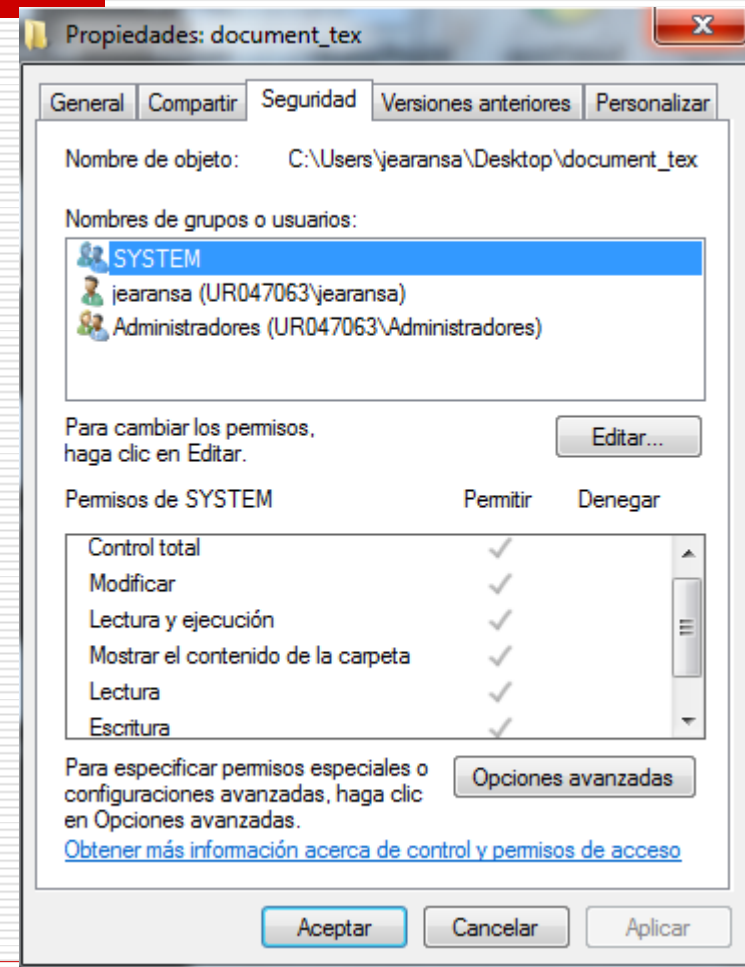
### Características y limitaciones

---

- ❑ Control total: cambiar permisos, tomar la propiedad y configurar las acciones permitidas por todos los demás permisos ntfs
  - ❑ Modificar: modificar y borrar el archivo y configurar las acciones permitidas por el permiso de escritura y el de lectura y ejecución
  - ❑ Lectura y ejecución: ejecución de aplicaciones y configurar las acciones permitidas por el permiso leer
  - ❑ Lectura: leer el archivo y ver sus atributos, propietario y permisos
  - ❑ Escritura: sobrescribir el archivo, cambiar sus atributos y ver el propietario del archivo y sus permisos
-

## 3.5.3 Archivos en SO Windows. Características y limitaciones

Permisos sobre  
carpetas:



## 3.5.3 Archivos en SO Windows.

### Características y limitaciones

---

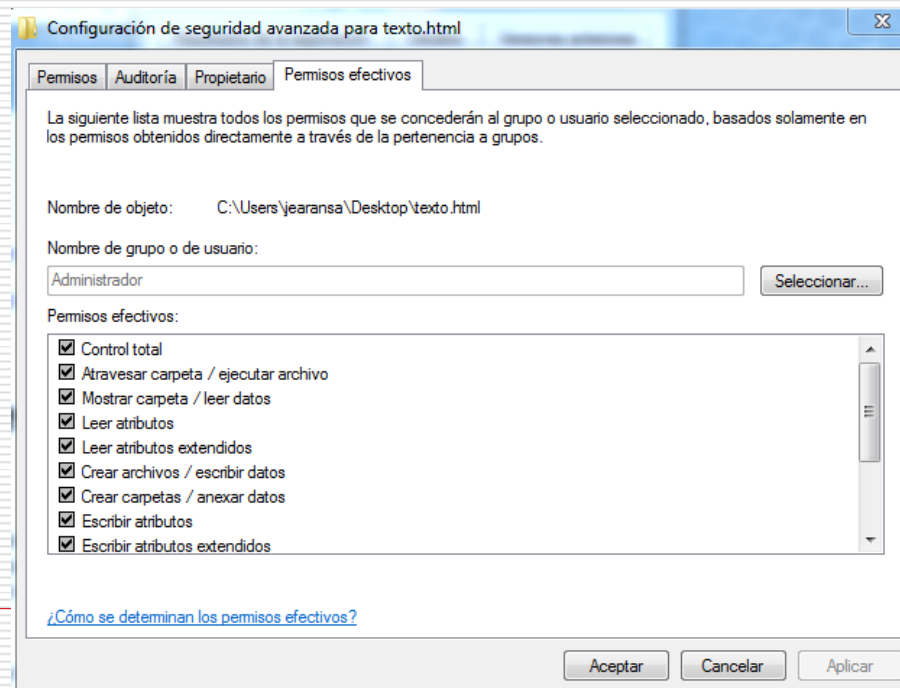
- ❑ Control total: cambiar permisos, tomar la propiedad, borrar subcarpetas y archivos y configurar las acciones permitidas por todos los demás permisos ntfs
  - ❑ Modificar: borrar la carpeta y configurar las acciones permitidas por el permiso de escritura, de lectura y de ejecución
  - ❑ Lectura y ejecución: navegar entre carpetas y configurar las acciones permitidas por el permiso de leer y el de listar contenidos de carpeta
  - ❑ Mostrar el contenido de la carpeta: ver los nombres de los archivos y subcarpetas en la carpeta
  - ❑ Lectura: ver archivos y subcarpetas en la carpeta, ver sus atributos, el propietario y sus permisos
  - ❑ Escritura: crear nuevos archivos y subcarpetas en la carpeta, cambiar los atributos de la carpeta y ver el propietario de la carpeta y sus permisos
-

## 3.5.3 Archivos en SO Windows.

### Características y limitaciones

---

Los permisos efectivos son otra herramienta que también nos permite conocer los permisos sobre un archivo o carpeta para cada usuario de una máquina; se pueden encontrar en "Propiedades->Seguridad->Opciones avanzadas->Permisos efectivos"; se calculan a partir de los permisos "reales" y de los propios e la carpeta y el usuario deseado:



### 3.5.3 Archivos en SO Windows.

## Características y limitaciones

---

Cómo modificar los permisos de un fichero en Windows: la herramienta que nos permite modificar los permisos es "Icalcs" (versión actualizada de cacs). Por medio de Icacs en el intérprete cmd se puede conseguir ayuda sobre el mismo

---

# 3.5.3 Archivos en SO Windows.

## Características y limitaciones

---

Algunos ejemplos de uso

```
icacls c:\windows\* /save AclFile /T
```

Guarda todos las listas de control de accesos de todos los ficheros y carpetas en c:\windos en un fichero llamado AclFile

```
icacls file /grant Administrator: (D,WD)
```

Garantiza al usuario de nombre Administrator los derechos de borrado (Delete) y de escritura (WD) para el fichero de nombre "file"

```
icacls file /grant *S-1-1-0: (D,WD)
```

Garantiza al usuario o grupo de identificador S-1-1-0 los permisos de borrado y de escritura para un fichero llamado "file"

```
icacls c:\windows\explorer.exe
```

Muestra la lista de control de acceso del fichero "c:\windows\explorer.exe"

```
icalcs file /setowner user
```

Cambia el propietario de "file" a "user"

---

## 3.6 Tareas o procesos y servicios

---

- 3.6.1 Procesos dentro de un sistema operativo. Qué son y para qué sirven
  - 3.6.2 Servicios en un sistema operativo
  - 3.6.3 Herramientas de gestión de procesos y servicios en Linux
  - 3.6.4 Herramientas de gestión de procesos y servicios en Windows
-

## 3.6.1 Procesos dentro de un SO

---

- ❑ Dentro de las tareas propias de un SO, una de las más importantes consiste en la gestión de procesos (junto con la gestión de E/S, gestión de ficheros ó gestión de memoria)
  - ❑ Un proceso es un programa que se encuentra en ejecución (un programa es sólo un conjunto de órdenes máquina)
  - ❑ También lo podemos definir como la unidad de procesamiento gestionada por el SO
-



## 3.6.1 Procesos dentro de un SO

---

- ❑ Durante su ejecución, un proceso debe residir en el mapa de memoria principal del ordenador (el gestor de procesos lo “copiará” en la misma al iniciar la ejecución). Esta copia se denomina imagen de memoria del proceso.
  - ❑ Desde el mapa de memoria principal, el proceso va modificando los registros del computador (el estado del procesador). A la par modifica su imagen de memoria
  - ❑ Los procesos son volátiles ya que, una vez completada su ejecución, desaparecen de la memoria principal (mientras que los programas perduran en el “disco duro” hasta que sean borrados)
-

## 3.6.1 Procesos dentro de un SO

---

El sistema operativo (por medio del gestor de procesos) debe proveer servicios para:

- Crear procesos: un proceso es creado cuando así lo solicita otro proceso (que se convierte en “padre” del primero). Hay dos posibilidades:
    - Creación a partir de la imagen del proceso padre. El proceso hijo es una imagen exacta del proceso padre, pero que luego se puede usar para fines completamente distintos. En Linux (y POSIX) el mandato que nos permite hacer esto es “fork”
      - Por ejemplo, cuando ejecutamos una tubería desde la shell  
**find . -name "\*.cpp" -print | wc -l**  
dos copias de la shell (fork) ejecutarán cada uno de los procesos involucrados
    - Creación a partir de un fichero ejecutable
-

## 3.6.1 Procesos dentro de un SO

---

El sistema operativo (por medio del gestor de procesos) debe proveer servicios para:

- ❑ Ejecutar un proceso (de tres formas, batch, interactiva y en segundo plano):
    - Batch o de lotes: no está asociado a ninguna terminal; toma sus datos de entrada de ficheros y vuelca sus salidas a otros ficheros
    - interactivo: el proceso está asociado a una terminal. Por defecto, leerá de la terminal sus datos de entrada y mandará sus resultados a la misma (por ejemplo, "ls" o "dir")
    - en segundo plano: estos procesos están asociados a una terminal, pero no bloquean la misma
-

## 3.6.1 Procesos dentro de un SO

---

El sistema operativo (por medio del gestor de procesos) debe proveer servicios para:

- ❑ Terminar la ejecución de un proceso. Puede haber varios motivos para lo mismo:
    - el proceso ha llegado a su final
    - el proceso ha sufrido un error en su ejecución
    - otro proceso o usuario decide que debe terminar dicho proceso (por ejemplo, por medio de kill)
-

## 3.6.1 Procesos dentro de un SO

---

El sistema operativo (por medio del gestor de procesos) debe proveer servicios para:

- ❑ Cambiar el ejecutable de un proceso: el programa que está asociado a un proceso es cambiado por otro proceso distinto (no se genera un nuevo proceso, sino que el proceso original se sustituye por uno nuevo). En Unix el servicio "exec" se encarga de dicha operación
-

## 3.6.1 Procesos dentro de un SO

---

El sistema operativo (por medio del gestor de procesos) debe proveer servicios para:

- ❑ Enviar señales a los procesos que permitan modificar su ejecución. Por ejemplo, por medio del mandato "kill" en Linux:

kill -15 PID (PID es el identificador del proceso)

kill -SIGTERM PID

kill -9 PID

kill -SIGKILL PID

Siempre que arrancamos un proceso, el mismo recibe un número PID (process identification) gracias al cual podemos comunicarnos con el mismo

---

## 3.6.2 Servicios dentro de un SO

Los servicios son aplicaciones (o conjuntos de aplicaciones) que se ejecutan en segundo plano a la espera de ser usadas, o llevando a cabo tareas esenciales (por ejemplo, MySQL, pero también otras más básicas para gestión de ficheros, funciones de red, usuarios...).



## 3.6.2 Servicios dentro de un SO

Servicios (locales)					
Seleccione un elemento para ver su descripción.					
Nombre	Descripción	Estado	Tipo de inicio	Iniciar sesión como	
Aplicación auxiliar de NetBIOS sobre TCP/IP	Proporciona compatibilidad para el servicio NetBIOS ...	Iniciado	Automático	Servicio local	
Aplicación auxiliar IP	Proporciona conectividad de túnel mediante tecnolo...	Iniciado	Automático	Sistema local	
Aplicación del sistema COM+	Administra la configuración y el seguimiento de los c...		Manual	Sistema local	
Archivos sin conexión	El servicio de archivos sin conexión realiza actividade...	Iniciado	Automático	Sistema local	
Asignador de detección de topologías de nivel de vínculo	Crea un mapa de red con información sobre la topol...		Manual	Servicio local	
Asignador de extremos de RPC	Resuelve identificadores de interfaces RPC en extrem...	Iniciado	Automático	Servicio de red	
Audio de Windows	Administra el audio para programas basados en Win...	Iniciado	Automático	Servicio local	
Audio Service	Manages audio jack configurations.	Iniciado	Automático	Sistema local	
Ayuda del Panel de control de Informes de problemas y soluciones	Este servicio proporciona ayuda para ver, enviar y bor...		Manual	Sistema local	
Bluetooth Service	Ejecuta la instalación y eliminación de los dispositi...	Iniciado	Automático	Sistema local	
BranchCache	Este servicio almacena en caché el contenido de la re...		Manual	Servicio de red	
Brillo adaptable	Supervisa los sensores de luz ambiental para detectar ...		Manual	Servicio local	
Captura SNMP	Recibe mensajes de captura generados por agentes l...		Manual	Servicio local	
Centro de seguridad	El servicio WSCSV (Centro de seguridad de Window...	Iniciado	Automático (i...	Servicio local	
Cliente de directiva de grupo	Este servicio es responsable de aplicar en el equipo y l...	Iniciado	Automático	Sistema local	
Cliente de seguimiento de vínculos distribuidos	Mantiene los vínculos entre archivos NTFS dentro de ...	Iniciado	Automático	Sistema local	
Cliente DHCP	Registra y actualiza las direcciones IP y los registros D...	Iniciado	Automático	Servicio local	
Cliente DNS	El servicio Cliente DNS (dnscache) almacena en cach...	Iniciado	Automático	Servicio de red	
Cliente web	Habilita los programas basados en Windows para qu...		Manual	Servicio local	
Cola de impresión	Carga archivos en la memoria para imprimirlos más t...	Iniciado	Automático	Sistema local	
Com4QLBEx		Iniciado	Manual	Sistema local	
Compilador de extremo de audio de Windows	Administra los dispositivos de audio para el servicio d...	Iniciado	Automático	Sistema local	
Componente de control central de Trend Micro	Gestiona todos los componentes de Trend Micro Inte...	Iniciado	Automático	Sistema local	
Conexión compartida a Internet (ICS)	Proporciona servicios de traducción de direcciones d...		Manual	Sistema local	
Conexiones de red	Administra objetos en la carpeta Conexiones de red y...	Iniciado	Manual	Sistema local	
Configuración automática de redes cableadas	El Servicio de configuración automática de redes cabl...		Manual	Sistema local	
Configuración automática de WLAN	El servicio WLANSVC proporciona la lógica necesaria ...	Iniciado	Automático	Sistema local	
Configuración automática de WWAN	Este servicio administra conexiones y adaptadores de ...		Manual	Servicio local	



### 3.6.3 Herramientas de gestión de procesos y servicios en Linux

---

Algunos mandatos útiles para gestión de procesos en Linux:

- ❑ `top` (table of processes): muestra, en tiempo real, la lista de procesos que se encuentran en ejecución en el sistema (para todos los usuarios). Contiene información, por ejemplo, sobre el PID (process identification), el usuario que ha lanzado el proceso, porcentaje de memoria usado, la terminal que lo ejecuta...
-

## 3.6.3 Herramientas de gestión de procesos y servicios en Linux

---

Algunas opciones útiles para top:

- ❑ `top -u usuario`: mostrar sólo los procesos de "usuario"
- ❑ `top -p PID`: mostrar sólo la información sobre el proceso "PID"

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5336	jesus	20	0	21424	12m	7948	S	1.3	2.5	0:01.88	metacity
5311	jesus	20	0	40160	9m	8200	S	0.7	2.0	0:00.30	gnome-settings-
5526	jesus	20	0	78400	22m	11m	R	0.7	4.6	0:09.36	gnome-terminal
5339	jesus	20	0	78440	26m	13m	S	0.3	5.3	0:00.85	nautilus
5174	jesus	20	0	7836	4764	2004	S	0.0	0.9	0:00.38	gconfd-2
5176	jesus	20	0	14524	2248	1836	S	0.0	0.4	0:00.00	gnome-keyring-d
5178	jesus	20	0	29296	7812	6556	S	0.0	1.5	0:00.08	x-session-manag
5285	jesus	20	0	5564	1824	1436	S	0.0	0.4	0:00.00	VBoxClient
5293	jesus	20	0	5600	1608	1160	S	0.0	0.3	0:00.00	VBoxClient
5297	jesus	20	0	5544	1208	864	S	0.0	0.2	0:00.00	VBoxClient
5306	jesus	20	0	23520	7048	5076	S	0.0	1.4	0:00.12	seahorse-agent
5310	jesus	20	0	2696	1100	736	S	0.0	0.2	0:00.04	dbus-daemon
5315	jesus	20	0	28600	5800	3336	S	0.0	1.1	0:01.31	pulseaudio
5318	jesus	20	0	5776	2252	1836	S	0.0	0.4	0:00.00	gconf-helper

Country	Year	Value
Algeria	2000	0.0000
Algeria	2001	0.0000
Algeria	2002	0.0000
Algeria	2003	0.0000
Algeria	2004	0.0000
Algeria	2005	0.0000
Algeria	2006	0.0000
Algeria	2007	0.0000
Algeria	2008	0.0000
Algeria	2009	0.0000
Algeria	2010	0.0000
Algeria	2011	0.0000
Algeria	2012	0.0000
Algeria	2013	0.0000
Algeria	2014	0.0000
Algeria	2015	0.0000
Algeria	2016	0.0000
Algeria	2017	0.0000
Algeria	2018	0.0000
Algeria	2019	0.0000
Algeria	2020	0.0000
Algeria	2021	0.0000
Algeria	2022	0.0000
Algeria	2023	0.0000
Algeria	2024	0.0000
Algeria	2025	0.0000
Algeria	2026	0.0000
Algeria	2027	0.0000
Algeria	2028	0.0000
Algeria	2029	0.0000
Algeria	2030	0.0000
Algeria	2031	0.0000
Algeria	2032	0.0000
Algeria	2033	0.0000
Algeria	2034	0.0000
Algeria	2035	0.0000
Algeria	2036	0.0000
Algeria	2037	0.0000
Algeria	2038	0.0000
Algeria	2039	0.0000
Algeria	2040	0.0000
Algeria	2041	0.0000
Algeria	2042	0.0000
Algeria	2043	0.0000
Algeria	2044	0.0000
Algeria	2045	0.0000
Algeria	2046	0.0000
Algeria	2047	0.0000
Algeria	2048	0.0000
Algeria	2049	0.0000
Algeria	2050	0.0000
Algeria	2051	0.0000
Algeria	2052	0.0000
Algeria	2053	0.0000
Algeria	2054	0.0000
Algeria	2055	0.0000
Algeria	2056	0.0000
Algeria	2057	0.0000
Algeria	2058	0.0000
Algeria	2059	0.0000
Algeria	2060	0.0000
Algeria	2061	0.0000
Algeria	2062	0.0000
Algeria	2063	0.0000
Algeria	2064	0.0000
Algeria	2065	0.0000
Algeria	2066	0.0000
Algeria	2067	0.0000
Algeria	2068	0.0000
Algeria	2069	0.0000
Algeria	2070	0.0000
Algeria	2071	0.0000
Algeria	2072	0.0000
Algeria	2073	0.0000
Algeria	2074	0.0000
Algeria	2075	0.0000
Algeria	2076	0.0000
Algeria	2077	0.0000
Algeria	2078	0.0000
Algeria	2079	0.0000
Algeria	2080	0.0000
Algeria	2081	0.0000
Algeria	2082	0.0000
Algeria	2083	0.0000
Algeria	2084	0.0000
Algeria	2085	0.0000
Algeria	2086	0.0000
Algeria	2087	0.0000
Algeria	2088	0.0000
Algeria	2089	0.0000
Algeria	2090	0.0000
Algeria	2091	0.0000
Algeria	2092	0.0000
Algeria	2093	0.0000
Algeria	2094	0.0000
Algeria	2095	0.0000
Algeria	2096	0.0000
Algeria	2097	0.0000
Algeria	2098	0.0000
Algeria	2099	0.0000
Algeria	2100	0.0000
Algeria	2101	0.0000
Algeria	2102	0.0000
Algeria	2103	0.0000
Algeria	2104	0.0000
Algeria	2105	0.0000
Algeria	2106	

```
gnome-settings-daemon 3.36.0-1ubuntu1~ppc64el
```

### 3.6.3 Herramientas de gestión de procesos y servicios en Linux

---

En la raíz del árbol siempre se encuentra el proceso “init”, proceso responsable del arranque del sistema (y de todos los otros procesos). El proceso “init” admite distintos niveles de arranque. Los modos de arranque disponibles son:

---

## 3.6.3 Herramientas de gestión de procesos y servicios en Linux

---

- Nivel de ejecución 0:
    - Halt. Este nivel de ejecución se encarga de detener todos los procesos activos en el sistema, enviando a la placa madre una interrupción para el completo apagado del equipo
  - Nivel de ejecución 1:
    - Single. Nivel de ejecución monousuario, sin acceso a servicios de red. Este nivel es regularmente utilizado en tareas de mantenimiento del sistema, y el usuario que se ejecuta es root (modo a prueba de fallos)
  - Nivel de ejecución 2:
    - Al igual que el nivel de ejecución monousuario, pero con funciones de red y compartición de datos mediante nfs
  - Nivel de ejecución 3:
    - Sistema multiusuario, con capacidades plenas de red, sin entorno gráfico. Este nivel de ejecución es el recomendado para sistemas de servidor, ya que evita la carga innecesaria de aplicaciones consumidoras de recursos
  - Nivel de ejecución 4: Nivel especificado, pero no se utiliza
  - Nivel de ejecución 5:
    - Al igual que el nivel de ejecución 3, pero con capacidades gráficas. Ideal para entornos de escritorio (es el que suele arrancar por defecto en ordenadores de sobremesa)
  - Nivel de ejecución 6:
    - Reboot. Este nivel de ejecución se encarga de detener todos los procesos activos en el sistema, enviando a la placa madre una interrupción para el reinicio del equipo
-

### 3.6.3 Herramientas de gestión de procesos y servicios en Linux

---

Al arrancar el sistema, el proceso init cargará los procesos que encuentre en la carpeta `/etc/rc{0..6}.d/` , dependiendo del nivel de inicio (de 0 6) indicado

---

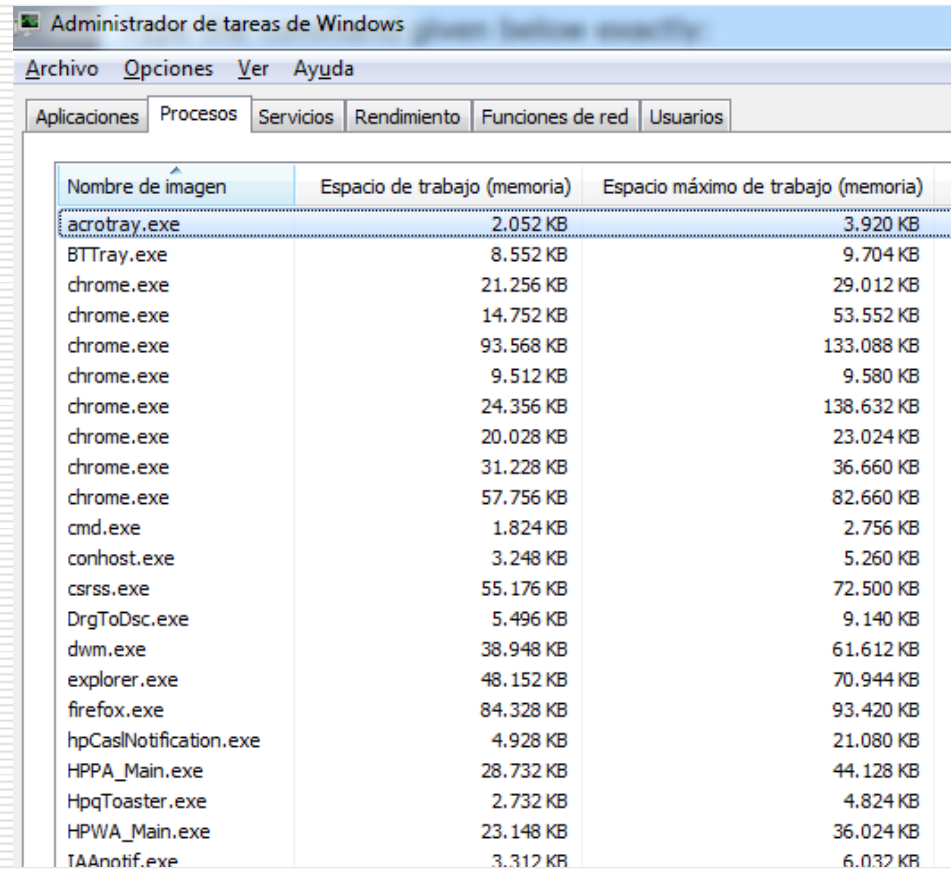
## 3.6.3 Herramientas de gestión de procesos y servicios en Linux

---

- Algunos mandatos útiles para trabajar con procesos:
    - &: envía el proceso iniciado a segundo plano: gedit fichero &  
Esto permite seguir usando el intérprete de mandatos
    - jobs: muestra la lista de procesos iniciados desde una terminal
    - fg: permite traer un proceso al primer plano
    - bg: permite enviar un proceso al segundo plano
    - kill -s SIGNAL PID: permite mandar la señal "SIGNAL" al proceso "PID"
  - Algunas señales comunes:
    - SIGTERM ó 15: señal de terminación (saliendo del proceso de forma correcta, cerrando ficheros abiertos, liberando memoria...) (Ctrl + C)
    - SIGKILL ó 9: señal de terminación inmediata (generalmente debería usarse sólo si SIGTERM no funciona)
    - SIGSTOP ó 19: detiene la ejecución de un proceso (Ctrl + Z)
-

## 3.6.4 Herramientas de gestión de procesos y servicios en Windows

taskmgr.exe:  
gestor de  
procesos (del  
entorno de  
ventanas) de  
Windows



Administrador de tareas de Windows

Archivo Opciones Ver Ayuda

Aplicaciones Procesos Servicios Rendimiento Funciones de red Usuarios

Nombre de imagen	Espacio de trabajo (memoria)	Espacio máximo de trabajo (memoria)
acrotray.exe	2.052 KB	3.920 KB
BTTray.exe	8.552 KB	9.704 KB
chrome.exe	21.256 KB	29.012 KB
chrome.exe	14.752 KB	53.552 KB
chrome.exe	93.568 KB	133.088 KB
chrome.exe	9.512 KB	9.580 KB
chrome.exe	24.356 KB	138.632 KB
chrome.exe	20.028 KB	23.024 KB
chrome.exe	31.228 KB	36.660 KB
chrome.exe	57.756 KB	82.660 KB
cmd.exe	1.824 KB	2.756 KB
conhost.exe	3.248 KB	5.260 KB
csrss.exe	55.176 KB	72.500 KB
DrgToDsc.exe	5.496 KB	9.140 KB
dwm.exe	38.948 KB	61.612 KB
explorer.exe	48.152 KB	70.944 KB
firefox.exe	84.328 KB	93.420 KB
hpCaslNotification.exe	4.928 KB	21.080 KB
HPPA_Main.exe	28.732 KB	44.128 KB
HpqToaster.exe	2.732 KB	4.824 KB
HPWA_Main.exe	23.148 KB	36.024 KB
IAAnotif.exe	3.312 KB	6.032 KB