

UNIDAD 3



Introducción a los Sistemas Operativos

Sistemas Informáticos
1º de DAM Semipresencial
IES San Vicente 2020/2021

Index

Sistema operativo y evolución.....	3
¿Qué es un sistema operativo?.....	3
Evolución de los sistemas operativos.....	3
Funciones de un SO.....	5
Clasificación de los SO.....	5
Sistemas operativos actuales.....	8
Microsoft Windows.....	8
GNU / Linux.....	8
UNIX.....	8
MAC OS.....	9
Sistemas operativos para móviles.....	9
Sistema de archivos.....	10
Tipos de sistemas de archivos.....	10
Archivos.....	11
Directorios.....	11
Operaciones básicas con archivos y directorios.....	13
Conocer el directorio actual de trabajo.....	13
Movernos a otro directorio.....	13
Crear un archivo o directorio.....	13
Copiar archivos.....	14
Listar contenido de un directorio.....	14
Mover un archivo o directorio.....	14
Borrar un archivo o directorio.....	14
Ayuda en comandos.....	14
Caracteres comodín.....	15
Comprimir archivos con comandos (solo Linux).....	15
Mantener sistema actualizado e instalar nuevo software (solo Linux).....	16
Instalación de un sistema operativo.....	17

Sistema operativo y evolución

¿Qué es un sistema operativo?

Un sistema operativo se podría definir como el programa o conjunto de programas que actúa como intermediario entre el usuario (o aplicaciones de usuario) y el hardware del ordenador, gestionando los recursos y optimizando su uso.

El Sistema Operativo es el primer programa que se carga en memoria RAM al encender el ordenador. Cuando el ordenador arranca y la BIOS/UEFI comprueba que todo está correcto busca el cargador de arranque (Bootloader) del sistema operativo para iniciarlo. Este se encuentra en el sector de arranque del disco duro o SSD (MBR → Master Boot Record) si tenemos un sistema con BIOS, o en la partición EFI del disco para sistemas con UEFI.

La finalidad del sistema operativo es presentar al resto de aplicaciones, y a los usuarios del sistema, los recursos hardware de la máquina de una forma sencilla y directa de gestionar y programar.

El usuario o programador de aplicaciones no se tienen que preocupar de como está estructurada internamente la información que hay en el disco duro, ni de como se planifican los procesos para su ejecución, ni de como funciona un modelo concreto de tarjeta gráfica, o de red, etc. El sistema operativo se encarga de ello mediante los controladores o drivers, y presenta a las aplicaciones una interfaz más genérica para comunicarse con el hardware.



Evolución de los sistemas operativos

Procesamiento en serie (Años 40,50)

El programador interactuaba directamente con el hardware (debía conocerlo a fondo), no había sistema operativo. Se reservaba un tiempo de la máquina para ejecutar un programa (sólo 1 a la vez). Si el programa terminaba antes, se desperdiciaba tiempo. En el caso de producirse un error, el programa se tenía que revisar manualmente para corregir los fallos y volverlo a intentar ejecutar en otro momento que la máquina estuviera disponible.

Las máquinas disponían de periféricos de entrada (cintas magnéticas, tarjetas perforadas) desde donde se cargaban los programas y datos, y un dispositivo de salida, impresora (no habían monitores). Unos indicadores luminosos y unos botones permitían ver el estado de los bits de la memoria y registros, así como también si se había producido un error.



Sistemas por lotes (años 50)

Para mejorar la utilización de la máquina, se introdujo el primer sistema operativo (General Motors OS para la IBM 704).

Su funcionamiento se basa en un programa llamado “monitor” cargado siempre en memoria, que se encarga de gestionar los trabajos pendientes (antes se tenían que cargar de 1 en 1), y pasar al siguiente cuando el programa actual acaba o se produce un error.

Cuando un trabajo acaba, devuelve el control al “monitor”. Antes de ejecutar una tarea, el monitor introduce unas instrucciones de control de trabajos en cada tarea para que esta, cuando termine o si produce un error, devuelva el control de la máquina al monitor.

Características de estos sistemas operativos primitivos:

- **Protección de memoria:** Los programas de usuario no deben acceder a la zona de memoria donde está el monitor, en ese caso se produce un error y se le devuelve el control.
- **Temporizador:** Para que una tarea no monopolice el sistema, se puede cargar un temporizador con cada tarea que pasado un tiempo genera una interrupción y devuelve el control al monitor.
- **Instrucciones privilegiadas:** Estas instrucciones sólo las puede ejecutar el monitor, si algún programa las intenta ejecutar se producirá un error.

Sistemas por lotes con multiprogramación (años 60)

Los dispositivos de Entrada/Salida eran (y son) bastante lentos comparados con el procesador, así que el procesador permanecía ocioso una parte del tiempo.

Se introdujo una nueva gestión de procesos en la cual, si un proceso estaba esperando datos de un dispositivo de E/S (Entrada/Salida → dispositivo de almacenamiento, periférico, etc) , el sistema podía para su ejecución y cargar otro trabajo en la CPU.

Los programas que se podían cargar a la vez en memoria dependía de el tamaño de esta, el sistema operativo podía ir intercambiando cada vez que una tarea se queda esperando una operación de E/S.

Sistemas de tiempo compartido

Entre la mejora de interacción del usuario con la máquina por medio de más avanzados dispositivos de entrada, y que, debido a los altos costes de las máquinas, era más económico que varios usuarios compartieran una máquina, se ideó un sistema que permitía la interacción de varios usuarios (multiusuario) a la vez, de forma aparente.

El sistema operativo no tiene por qué esperar a una operación de E/S por parte

de un programa para pasar a otro, sino que los programas tienen asignado un quantum, o tiempo de ejecución limitado, pasado el cual abandonan la CPU, dejando paso al siguiente, y se quedan esperando para volver a ejecutarse.

Con un quantum pequeño da la sensación de que se están ejecutando varias tareas al mismo tiempo (multiproceso), pero no es así de forma real (ya que solo había un procesador).

Sistemas multiprocesador

El sistema operativo es capaz de gestionar más de un procesador, y por lo tanto ejecutar más de un proceso al mismo tiempo de forma real.

De todas maneras, se mantiene también el avance anterior, ya que si disponemos de 2 CPUs, y debemos ejecutar 5 tareas, el quantum es la única forma de que parezca que están ejecutandose todas a la vez cuando realmente se ejecutan 2 de forma simultánea.

Funciones de un SO

Como antes hemos comentado, una de las funciones principales de un SO es la administración del hardware de la máquina por medio de drivers o controladores (integrados en el sistema o instalados a posteriori).

También se encarga del control de la ejecución de los programas (procesos):

- Crear y destruir procesos
- Parar y reanudar procesos
- Ofrecer mecanismos de comunicación y sincronización entre procesos

Gestión de la memoria principal:

- Saber que partes están siendo usadas y por quién
- Decidir que procesos o parte de ellos se cargarán en memoria cuando haya espacio.
- Asignar y reclamar un espacio de memoria cuando sea necesario

Gestión de usuarios y permisos:

- Usuarios que existen en el sistema y sus roles
- Permisos de esos usuarios para el uso de archivos y recursos

El SO también se encarga entre otras muchas cosas del control de concurrencia (varios procesos quieren acceder a un mismo recurso, por ejemplo a una unidad de E/S), control de errores, control de seguridad, etc.

Clasificación de los SO

En base a sus características podemos clasificar los sistemas operativos en

varios grupos. Esto no quiere decir que dichos grupos sean exclusivos, ya que un sistema operativo puede pertenecer a varios de ellos.

Según los servicios ofrecidos

Número de usuarios

- **Monousuario**
- **Multiusuario** → Capacidad de dar servicio a más de un usuario a la vez

Número de tareas simultáneas

- **Monotarea** → Un sólo proceso (hasta que no termine no se lanza el siguiente)
- **Multitarea** → Varios procesos de forma simultánea

Número de procesadores

- **Monoproceso** → Sólo aprovechan 1 CPU, pero pueden simular multitarea (quantum)
- **Multiproceso** → Soportan varios procesos simultáneos de forma real (varios procesadores o unidades de proceso).

Por la forma de ofrecer los servicios

Sistemas centralizados: Un ordenador central se encarga de realizar todas las tareas de computación. Los usuarios se conectan a él por medio de “terminales tontos” (sistemas periféricos sin memoria ni CPU o lo mínimo).

Algunos terminales son “clientes híbridos” con ciertas capacidades de computación. Y tareas repartidas con el ordenador central, algunas las ejecuta el cliente y otras no.

Sistemas distribuidos: Red de ordenadores independientes entre si, pero que se presentan como si fuese un único sistema para tareas de computación. El software que controla estas máquinas es capaz de repartir los cálculos entre ellas como si de un único sistema multiprocesador se tratara.

Por su disponibilidad

SO propietarios: Su código fuente, y por lo tanto, los derechos y permisos sobre su uso (Licencia) son propiedad de una empresa

SO libres: Software libre en general es aquel que cumple con las siguientes libertades:

0. La libertad de usar el programa, con cualquier propósito.
1. La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a tus necesidades.
2. La libertad de distribuir copias del programa.

3. La libertad de mejorar el programa y hacer públicas esas mejoras (obligación) a los demás, de modo que toda la comunidad se beneficie.

Otras clasificaciones

Sistemas de tiempo real: Diseñados para tareas críticas como puede ser el control de los sistemas electrónicos de un coche o avión. Deben ser capaces de responder correctamente en un espacio de tiempo muy limitado. Si tardan en reponder a una petición crítica se considera un fallo. La fiabilidad es fundamental.

Sistemas operativos en red: Sistemas que mantienen conectados entre sí a 2 o más ordenadores compartiendo recursos entre sí. Las máquinas suelen ser independientes y comparten información, hardware y software. Los más orientados a esta tarea serían Windows Server o cualquier sistema Linux/UNIX. Otros sistemas domésticos también ofrecen estos servicios pero menos avanzados.

Sistemas operativos cliente o de escritorio: Hoy en día con funciones de Sistema Operativo en red, pero más orientados a su uso como cliente de una red u ordenador independiente (Windows 10, o distribuciones Linux como Ubuntu, OpenSuse,...).

Sistemas operativos actuales

Microsoft Windows

Sistema Operativo propietario desarrollado por Microsoft (versión 1.0 lanzada en 1985). Desde Windows 1.0 hasta 3.11, se podría decir que era una aplicación con entorno gráfico de MS-DOS, más que un sistema operativo en sí.

Windows 95, 98 y ME, aunque eran sistemas completos, aún dependían de una base MS-DOS para funcionar, aunque la utilizaban más bien poco, pero los limitaba.

Los Windows actuales derivan del Windows NT, un sistema operativo diferente, con interfaz gráfica, que se lanzó para competir con UNIX en entornos empresariales. Evolución → Windows NT (hasta la 4.0), Windows 2000 (NT 5.0), Windows XP (NT 5.1), Windows Vista (NT 6.0), Windows 7 (NT 6.1), Windows 8 (NT 6.2) y Windows 10 (NT 10). Además están las versiones orientadas a servidores con más capacidades que las equivalentes domésticas (Windows Server).

GNU / Linux

Núcleo del SO desarrollado por Linus Torvalds y lanzado por primera vez en 1991, bajo licencia (GPL) de software libre. La intención original era crear un sistema similar a UNIX que funcionase en ordenadores x86 como un hobby.

Linux puede funcionar en más de 20 arquitecturas diferentes de CPU, muchas más que Windows.

Existen cientos de distribuciones (agrupaciones de software junto con el núcleo del SO) de Linux en la actualidad orientadas a muchos tipos de tarea. Ej: servidores → (Debian,...), Escritorio → (Ubuntu, Opensuse,...), Móviles (Android,...), Ordenadores antiguos (Puppy Linux,...) y muchas más (<http://distrowatch.com>).

A diferencia de Windows, la interfaz gráfica no forma parte del sistema operativo, lo que le dota de mayor flexibilidad frente a diferentes necesidades (funcionalidad, recursos,...).

UNIX

El Unix original salió a principios de los años 70, con capacidades multitarea, multiusuario y políticas de seguridad de acceso a archivos.

De la versión original, varias empresas obtuvieron licencia para su desarrollo y crearon versiones personalizadas (al estilo de las distribuciones de Linux).

Existen varias familias:

- BSD nació en la Universidad de California (Berkeley), en esta familia fue donde primero se implementó el protocolo TCP/IP que dio lugar a internet. De aquí derivan versiones libres como FreeBSD, OpenBSD y NetBSD.

- AIX es la versión de IBM
- Solaris es la versión de Sun Microsystems (ahora es de Oracle)
- HP-UX es la versión de HP

MAC OS

Originalmente un Sistema Operativo creado exclusivamente por Apple para sus ordenadores Macintosh en 1984. Tenía una interfaz gráfica al estilo de otros SO de la época avanzados a su tiempo, como AmigaOS.

Las últimas versiones desde la versión 10 del sistema operativo (llamado OS X), se reescribieron desde 0 con respecto a versiones 9 y anteriores. Este SO está basado en el Unix FreeBSD, por lo que a nivel interno es similar a Linux.

La última versión del SO (10.15) se llama "Catalina". Se ejecuta en máquinas x86, pero Apple tiene muy controlado el hardware de forma que es muy difícil instalarlo en PCs que no sean de la marca. A una máquina no fabricada por Apple que ejecuta OSX se le conoce como Hackintosh.

Sistemas operativos para móviles

Los sistemas operativos para dispositivos móviles suelen estar basados en sistemas operativos de uso en ordenadores y servidores pero adaptados a las necesidades y consumo de recursos de los móviles (más limitados generalmente).

Sistemas basados en OS X

- **IOS** → Dispositivos móviles de Apple (Iphone, Ipad). Última versión: 13

Sistemas basados en Linux

- **Android** → Creado por Android Inc. (comprada luego por Google). Trabajan varias empresas en él bajo el consorcio "Open Handset Alliance". Casi todo el SO es libre. Última versión: 10
- **Tizen** → Desarrollado por Samsung y basado en Meego
- Ubuntu Mobile, Sailfish OS, etc.

Proyectos abandonados → Web OS (Palm), BADA (Samsung), Meego (Intel y Nokia), Firefox OS, Windows 10 Mobile.

Sistema de archivos

Los sistemas de archivos (o de ficheros) estructuran la información guardada en una unidad de almacenamiento. Cada sistema operativo puede trabajar con una de sistemas de archivos.

El sistema de archivos controla en qué sectores del disco duro se reparten los archivos y directorios, así como la gestión del espacio libre. También guarda información sobre la jerarquía de directorios (en forma de árbol).

También, para cada archivo o directorio se guardan una serie de atributos y de permisos, gestionados por una “lista de control de acceso”, que gestiona quién puede acceder y qué puede hacer con ellos.

Tipos de sistemas de archivos

Sistemas soportados nativamente por Windows

FAT32 → Utilizado desde Windows 95, similar al que usan las memorias USB hoy en día (vFAT o extFAT). No soporta permisos ni algunos atributos. Tamaño máximo de archivo: 4GB. Sufre de gran fragmentación.

NTFS → Usado desde las últimas versiones de Windows NT, este sistema soporta permisos en los archivos, tamaños superiores a 4GB. Su fragmentación es mucho menor. Y sobre todo, se introduce un registro transaccional (journaling) que protege al sistema de archivos para que no se corrompa ante, por ejemplo, un error del sistema o un apagón inesperado (no garantiza que no se pierda información, sólo que el sistema de archivos seguirá íntegro).

Sistemas de archivos más utilizados en Linux

Ext2, ext3, ext4 → Sistemas de archivos más utilizados en Linux, con características similares a NTFS, incorporan journaling desde la versión 3. La versión 4 incorpora mejoras en el rendimiento así como un aumento del tamaño máximo de archivo de 2TB (terabytes) a 16TB.

XFS → Sistema de archivos creado por Silicon Graphics. Es especialmente rápido con archivos de gran tamaño. Soporta archivos de mayor tamaño que ext y características avanzadas como journaling.

JFS → Creado por IBM para su sistema UNIX. Pensado para servidores de alto rendimiento. Busca un rendimiento lo más equilibrado posible en todos los escenarios.

Btrfs → Más nuevo que los anteriores. Se centra en la tolerancia a fallos, y optimización para unidades SSD (memoria flash). Aún no consigue el rendimiento de ext4 globalmente aunque destaca ya en algunas tareas.

Sistema de archivos de Solaris

ZFS → Diseñado por Sun Microsystems para su Unix (Solaris). Soporta características avanzadas como autorreparación, archivos de 16 exabytes (16 millones de terabytes), e instantáneas → Guarda copia de seguridad de los datos que se van modificando para poder recuperarlos en cualquier momento.

Sistema de archivos de OS X

APFS → Apple File System. Muy parecido a los sistemas de archivos de Linux, e incluso al sistema NTFS de Windows. Este sistema de archivos es bastante moderno y muy centrado en optimizar el trabajo con unidades de almacenamiento SSD.

Archivos

Los archivos representan la información almacenada en el disco. Se identifican con un “nombre de archivo”, terminando por una “extensión”, que indica el tipo de archivo (En Linux no es obligatoria aunque se suele poner) → **nombre.extensión**.

Windows no distingue entre mayúsculas y minúsculas en los nombres de archivo, Linux (y OSX) sí: ‘archivo’ y ‘Archivo’ → nombres diferentes.

Además del nombre, se almacenan por cada archivo, una serie de atributos y permisos que varían según el sistema de archivos utilizado.

Algunos atributos que asocia **Windows** con los archivos son:

- **(S) Atributo de Sistema** → Indica que el archivo es parte del SO (Sólo Windows)
- **(H) Atributo de oculto** → Indica que el archivo es oculto y no se debe mostrar. En Linux se hace esto poniendo un punto '.' delante del nombre.
- **(R) Atributo de sólo lectura** → Indica que el archivo es de sólo lectura y no se puede modificar. Linux gestiona esto mediante permisos.
- **(D) Atributo de Directorio** → Indica que el archivo es un directorio o carpeta.
- Fecha de creación y última modificación
- Tamaño que ocupa en disco

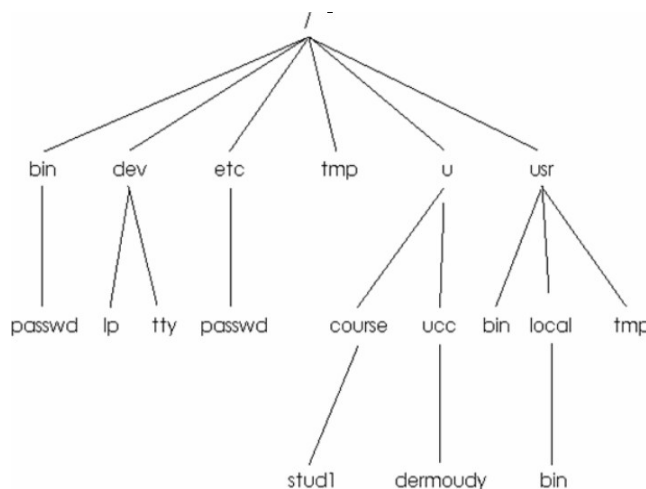
Otros atributos como el propietario de un archivo, o los permisos que se aplican a los usuarios sobre un archivo los veremos más adelante con mayor detalle.

Los archivos además, pueden ser ejecutables (y tener permisos de ejecución o no), lo cual indica que al actuar sobre ellos cargarán un programa.

En sistemas tipo Unix o Linux, todo está representado internamente por archivos, lo cual indica que no siempre van a contener información, sino que también pueden ser el punto de acceso a un dispositivo como un disco duro, ratón, teclado, etc.

Directorios

Llamados también carpetas cuando se manejan desde un entorno gráfico. Forman una estructura jerárquica en forma de árbol, de tal manera que pueden contener a su vez a otros directorios (subdirectorios) y archivos.



En un árbol siempre va a haber una raíz. En Linux (Unix, Mac) el directorio raíz se representa como “/”, mientras que en Windows sería el nombre de la unidad “C:\”.

Los separadores de directorios

Linux (Unix, Mac): “/” → /home/usuario/Documentos

Windows: “\” → C:\Documents and settings\usuario\Mis documentos

Para referenciar un archivo o un directorio del árbol podemos seguir 2 tipos de rutas:

Absoluta → Desde la raíz del árbol (las rutas anteriores de ejemplo), empezando por la barra “/” en Linux o la unidad en Windows.

Relativa → Desde el directorio actual donde nos encontremos trabajando. Se distinguen porque no empiezan por barra en Linux (Unix, Mac) o por la unidad en Windows. **Ejemplo:** Documentos/archivo.pdf.

Existen dos “tipos de directorios” o entradas especiales en una ruta:

. → Representa al directorio actual, en el que estamos

.. → Representa al directorio anterior. Al “padre”.

Las operaciones y los permisos de los directorios son muy parecidos a los archivos. Con diferencias, por ejemplo, en lugar de ejecutar (un directorio no se puede ejecutar), se aplicará el permiso de mostrar contenido.

Operaciones básicas con archivos y directorios

Nota: Si el nombre de un archivo o directorio contiene espacios lo mejor es poner dicho nombre (o ruta completa) entre comillas ("archivo de texto.txt"), ya que el espacio actúa como separador de parámetros en los comandos.

Conocer el directorio actual de trabajo

Aunque seguramente en el terminal nos aparecerá el directorio en el que nos encontramos, podemos ejecutar estos comandos para averiguarlo en cualquier momento:

- **pwd** (Linux)
- **echo %cd%** (Windows)

Movernos a otro directorio

Para movernos de un directorio a otro en línea de comandos usaremos el comando: **cd nueva_ruta**. Recordemos que en Windows la barra de separación de los directorios en una ruta es esta "\", mientras que en el resto de sistemas es esta "/".

Ruta absoluta

- **cd /home/usuario/Documentos** (Linux)
- **cd c:\Users\usuario/Documents** (Windows)

Ruta relativa

- **cd ..** (directorio anterior o padre)
- **cd ../..** (dos directorios hacia arriba)
- **cd ../Videos** (directorio Videos que está dentro del directorio padre)
- **cd peliculas** (directorio peliculas que está dentro del actual)

Crear un archivo o directorio

- Windows/Linux → **mkdir directorio** (crea un directorio)
- Linux → **touch archivo** (crea un archivo vacío)

Se pueden crear varios directorios o archivos si especificamos varios después del comando separados por espacios. También se pueden crear en directorios diferentes al actual si especificamos la ruta completa:

mkdir ../directorio1 ../directorio2 (crea 2 directorios dentro del directorio padre).

Copiar archivos

Windows → **copy ruta\archivo ruta_destino**

Linux → **cp -r ruta/archivo ruta_destino** (-r es necesario si queremos copiar directorios y sus contenidos).

En la ruta de destino se puede especificar un directorio existente (se generaría una copia del archivo con el mismo nombre) o un nombre de archivo (o directorio) que aún no exista (se copiaría con ese nombre).

También se puede especificar una lista de archivos o directorios a copiar (separados por espacio) y un directorio destino (que ya exista). Se copiarían dentro del directorio con el mismo nombre. Ejemplo:

cp arch1 arch2 arch3 directorio_destino

Listar contenido de un directorio

Comandos:

Windows → **dir**

Linux → **ls** (ls -l para lista más detallada)

Mover un archivo o directorio

Mover un archivo es como copiarlo pero borrando el origen después de hacer dicha copia. Se puede utilizar también como una forma de renombrar.

Windows → **move ruta/archivo nueva_ruta**

Linux → **mv -r ruta/archivo nueva_ruta** (-r permite en Linux mover un directorio en lugar de sólo archivos)

Se aplican las mismas reglas que para copiar.

Borrar un archivo o directorio

Windows → **del archivo** o **rmdir /s directorio** (/s se utiliza para borrar directorios no vacíos)

Linux → **rm archivo** o **rm -r directorio**.

Ayuda en comandos

Un comando se compone de:

comando opciones parámetros

Ejemplo: **rm -r dir1**

Para obtener más ayuda de como usar un comando y las opciones que admite:

Windows → **comando /?**

Linux → **comando -h** (no todos) o **comando --help**. Ayuda más detallada: **man comando**.

Caracteres comodín

Para facilitar la selección de ficheros con comandos de listar (ls,dir), copiar, mover, borrar,etc. Disponemos de 2 caracteres comodín:

- ***** : Implica cualquier secuencia de caracteres
 - **rm *** → Borra TODOS los archivos del directorio (Linux)
 - **a*** → Todos los archivos que empiezan por 'a'
 - **r*.txt** → Todos los archivos que empiezan por 'r' y acaban en 'txt' (r.txt también sirve, ya que * puede ser vacío)
- **?** : Implica 1 sólo carácter cualquiera
 - **?ote.*** → Archivos que empiecen por cualquier carácter seguido de 'ote.' y cualquier extensión (note.txt, lote.dat,...)
 - **docum_*.od?** → Archivos que empiecen por 'docum_' seguidos de cualquier cosa, y tengan la extensión '.od' + un carácter. (docum_viaje.ods, docum_trabajo.odt,...)

Comprimir archivos con comandos (solo Linux)

Estos son los comandos en Linux con los que podremos comprimir y descomprimir sin necesidad de herramientas gráficas cualquier tipo de archivo.

- **Gzip** → Algoritmo de compresión bastante rápido, aunque no es el más efectivo, pero la velocidad/ratio de compresión es buena.
 - **tar cvzf comprimido.tar.gz archivo1 achivo2 archivo3**(La extensión puede abreviarse a. tgz)
 - **tar xvzf comprimido.tar.gz** (para descomprimir)
- **Bzip2** → Herramienta muy similar a Gzip y también muy clásica dentro del mundo UNIX/Linux. Consigue mayor compresión que Gzip en bastantes casos, perdiendo algo de velocidad.

- **tar cvjf archivo.tar.bz2 archivo1 archivo2 archivo3 ...**
- **tar xvjf archivo.tar.bz2**
- **Zip y unzip** → Utiliza el mismo algoritmo de compresión que gzip, aunque no son intercambiables, si que tienen un rendimiento similar.
 - **zip archivo.zip archivo1 archivo2 archivo3**
 - Sin embargo, tiene la pega de que si alguno de los archivos es un directorio, no almacenará los contenidos del mismo, para ello se tiene que utilizar la opción '-r': **zip -r archivo.zip archivo1 dir1 archivo2 dir2**
 - Para descomprimir, simplemente utilizar la herramienta unzip: **unzip archivo.zip**
- **Rar y unrar** → Funciona de una forma muy similar a zip. Normalmente suele ser más efectivo para comprimir, de forma similar a bzip2, o mejor, aunque sensiblemente más lento (suele ser el más lento para comprimir). Para comprimir (-r para directorios):
 - **rar a -r archivo.rar arch1 arch2 arch3 ...**
 - **unrar x archivo.rar** (Para descomprimir, o '**rar x archivo.rar**')
 - **xz** → Este comando de compresión utiliza el algoritmo LZMA2 (el mismo que 7zip), con un tiempo de compresión medio (un poco menor que rar, parecido a bzip2), un ratio de compresión alto (a menudo comprime mejor que bzip2 y rar), y un tiempo de descompresión bastante bueno (entre gzip y bzip2).
- **tar cvJf archivo.tar.xz archivo1 archivo2 archivo3 ...**
- **tar xvJf archivo.tar.xz** (descomprimir)

Mantener sistema actualizado e instalar nuevo software (solo Linux)

La herramienta **apt** permite instalar paquetes situados en los repositorios online de una distribución Linux basada en Debian / Ubuntu / Mint, y mantener el sistema actualizado.

Instalar nuevos paquetes de software (programas, librerías)

apt-get install paquete

Buscar un paquete

apt-cache search nombre

Desinstalar (si en lugar de remove se utiliza purge, se borran también los ficheros de configuración)

apt-get remove paquete

Sincronizar con los repositorios (nuevas versiones, cambios en los repositorios,...)

apt-get update

Actualizar sistema

apt-get upgrade (sólo actualiza lo que está instalado. Si un paquete para actualizarse necesita instalar algo extra, no se actualiza)

apt-get dist-upgrade (actualiza e instala paquetes necesarios para resolver dependencias y borra paquetes obsoletos).

Instalación de un sistema operativo

Windows 10

Requisitos mínimos recomendados Windows 10:

- Procesador 1Ghz o superior
- 1GB de RAM (2GB si el sistema es de 64 bits)
- T. Gráfica compatible DX9
- 16GB espacio libre en disco duro 32bits (20GB 64 bits)

Para pasar la ISO de windows a un USB si no disponemos de unidad DVD, podemos utilizar el programa [Rufus](#).

Distribuciones Linux

Una distribución de linux es una distribución de software basada en el núcleo Linux como Sistema Operativo que incluye determinados “paquetes de software” para satisfacer las necesidades de un grupo específico de usuarios. Los requisitos mínimos de instalación pueden variar mucho de una distribución a otra.

Existen distribuciones para multitud de ámbitos: doméstico, servidores, empresariales, ordenadores antiguos, routers, etc. Las 3 distribuciones más antiguas (que aún continúan), y que han servido de base para casi todas las distribuciones modernas son Slackware, Debian y Red Hat.

http://es.wikipedia.org/wiki/Anexo:Distribuciones_Linux

https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux_Distribution_Timeline.svg

Algunas distribuciones según el ámbito para el que están más pensadas:

- **Escritorio:** Ubuntu, OpenSuse, Mint, ...
- **Servidor:** Debian, RedHat (CentOS), Gentoo, ...
- **Específicas:** Parted Magic, Gparted, Clonezilla (copias de seguridad y herramientas de diagnóstico,etc...). IPFire, ClearOS, Vyatta (Cortafuegos, proxy,etc...).

Y un montón de especializaciones y distribuciones más:

<http://distrowatch.com/search.php>

Hoy en día se ha extendido el formato LiveCD o LiveDVD. Un formato que te permite ejecutar y probar el sistema operativo desde el CD o DVD (más lento y sin

posibilidad de guardar cambios), antes de instalar. También se siguen distribuyendo CDs y DVDs de sólo instalación (ámbito más profesional).

En las páginas web de las distribuciones hay instrucciones de como pasar el LiveCD a un dispositivo USB. Programas como [Unetbootin](#) también sirven para pasar una imagen de CD a un USB (e incluso la descarga el propio programa). También otros programas muy conocidos como [Rufus](#) hacen esta función.

Entornos de escritorio

En Linux, al contrario que Windows o Mac OS, el escritorio que se maneja no está integrado en el SO, así que podemos elegir entre una multitud de opciones. Existen 2 grupos:

- **Entorno de escritorio** → Es un conjunto de software que te ofrece todo lo necesario para gestionar el sistema de forma gráfica (ventanas, gestor de archivos, barra de tareas, iconos, menús, etc...)

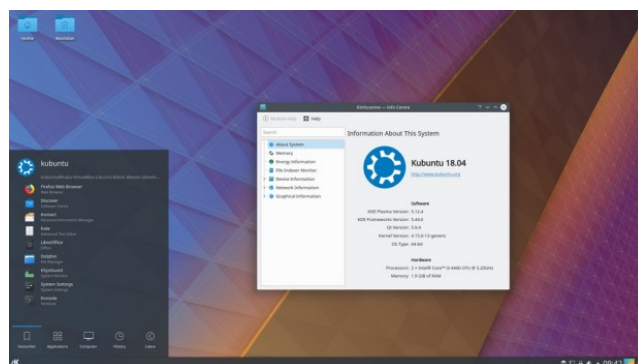
Gnome, KDE, XFCE, LXDE, Cinnamon, Mate, Enlightenment, ...

- **Gestor de ventanas** → Sólo se encarga de la gestión de las ventanas que tengamos abiertas, para el resto de tareas (fondo e iconos de escritorio, gestor de archivos, barras de herramientas) suele requerir de otros programas en combinación.

IceWm, FluxBox, Window Maker, JWM,...



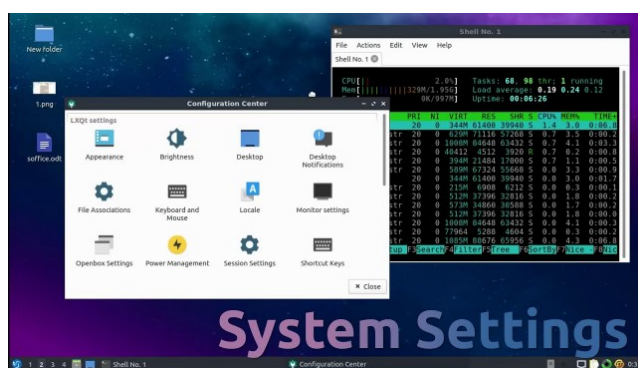
GNOME



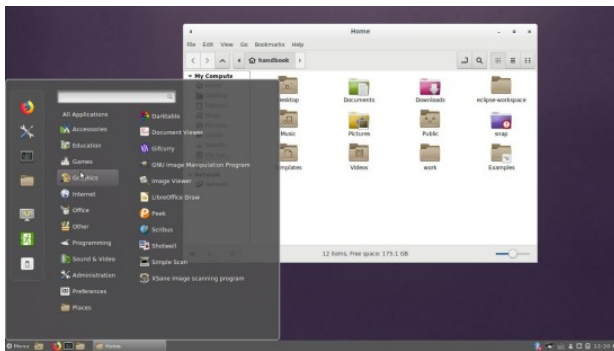
KDE



XFCE



LXQT



Cinnamon



Enlightenment

Particionado del disco

En un disco duro puede haber como máximo 4 particiones primarias, o 3 primarias y una extendida que a su vez puede contener muchas particiones lógicas.

Windows suele necesitar por lo menos 1 partición primaria para la unidad principal (C:). Se pueden crear unidades adicionales (para datos de usuarios, copias de seguridad, etc.) en otras particiones primarias o lógicas. Linux puede instalarse completamente en particiones lógicas, aunque no es necesario.

Para instalar Linux se necesitan como mínimo 2 particiones. Una que contenga la instalación raíz y que se etiquetará con “/”, y otra de memoria Swap o partición de intercambio (donde se guardan los datos menos usados de la RAM cuando esta se llena).

Es recomendable también separar el directorio “/home” en otra partición. Ahí es donde se guardarán todos los archivos y configuraciones personales de los usuarios. Así, si reinstalamos el sistema y formateamos la raíz no se pierde esa información.

Ejemplo de como quedarían las particiones en una instalación de la distribución Debian (el disco es sólo de 10,7GB. Normalmente en un disco mucho mayor, se le daría bastante más espacio a la partición /home que a la raíz “/”, que con entre 30 y 40GB máximo tendríamos de sobra).

```
[!!!] Particionado de discos

Éste es un resumen de las particiones y puntos de montaje que tiene configurados actualmente. Seleccione una partición para modificar sus valores (sistema de ficheros, puntos de montaje, etc.), el espacio libre para añadir una partición nueva o un dispositivo para inicializar la tabla de particiones.

Particionado guiado
Configurar RAID por software
Configurar el Gestor de Volúmenes Lógicos (LVM)
Configurar los volúmenes cifrados

SCSI1 (0,0,0) (sda) - 10.7 GB ATA VBOX HARDDISK
#1 primaria 5.0 GB B f ext4 /
#5 lógica 5.0 GB f ext4 /home
#6 lógica 736.1 MB f intercambio intercambio

Deshacer los cambios realizados a las particiones
Finalizar el particionado y escribir los cambios en el disco

<Retroceder>
```

En una instalación mixta (Windows/Linux) dentro de un mismo disco duro podríamos tener las siguientes particiones (por ejemplo):

- Primaria 1 → Partición de sistema de Windows (se crea automáticamente en la instalación y guarda información para el arranque).
- Primaria 2 → Partición principal Windows (unidad C:)
- Primaria 3 → Partición principal Linux (Raíz → /)
- Extendida (4)
 - Lógica (5) → Partición /home de Linux
 - Lógica (6) → Partición swap o de intercambio de Linux

Si tenemos 2 discos, debemos tener en cuenta que Windows debería estar situado en el principal (según la BIOS) y Linux en el secundario. Primero se instalaría Windows siempre y después Linux, ya que el cargador de arranque de Windows no es capaz de iniciar Linux pero sí a la inversa.

El cargador de arranque (GRUB en el caso de Linux), siempre se instalará en el disco duro de arranque (principal) del sistema, aunque el sistema operativo esté instalado en otro disco. En linux los discos se suelen nombrar con las letras del alfabeto en función del orden que tienen en la BIOS. El principal sería 'sda', secundario 'sdb', etc.