

TEMA 20

EXPLOTACIÓN Y ADMINISTRACIÓN DE SISTEMAS OPERATIVOS MONOUSUARIO Y MULTIUSUARIO

ÍNDICE

1. INTRODUCCIÓN
2. SISTEMA OPERATIVO MONOUSUARIO Y MULTIUSUARIO
3. SISTEMA OPERATIVO MONOUSUARIO: DOS
 - 3.1. Interfaz con el usuario
 - 3.2. Comandos y parámetros
 - 3.3. La sintaxis de las órdenes
 - 3.4. Gestor de memoria
 - 3.5. Configuración y personalización
 - 3.6. Gestión de dispositivos de almacenamiento masivo
 - 3.7. Órdenes para la gestión de ficheros
 - 3.8. Órdenes para la gestión de directorios y subdirectorios
 - 3.9. Recuperar la información
 - 3.10. Copias de seguridad
 - 3.11. Compresor de disco
 - 3.12. Desfragmentador
4. SISTEMA OPERATIVO MULTIUSUARIO: UNIX
 - 4.1. Comandos y metacaracteres
 - 4.2. Entrada y salida del sistema
 - 4.3. Usuarios y grupos en Unix
 - 4.4. Gestión de permisos
 - 4.5. Gestión de directorios y ficheros
 - 4.6. Procesos –ps-
 - 4.7. Editor vi
 - 4.8. Comandos de propósito general
 - 4.9. Gestión de impresión
5. BIBLIOGRAFÍA

1. INTRODUCCIÓN

Un sistema operativo es un conjunto de programas encargados de controlar y coordinar el funcionamiento del ordenador y de todos sus componentes.

El sistema operativo se encarga de gestionar directamente los dispositivos y recursos del ordenador de forma eficiente.

2. SISTEMA OPERATIVO MONOUSUARIO Y MULTIUSUARIO

Los sistemas operativos pueden clasificarse atendiendo a diversas características, por ejemplo: Por el número de usuarios, Por la utilización de los recursos, Por el tiempo de respuesta, Por la portabilidad del software.

Si nos fijamos en el *número de usuarios a los que puede dar servicio* a la vez tenemos la siguiente clasificación:

Monousuario:

Cuando todos los recursos del ordenador están asignados a un solo usuario, es decir, un único terminal del ordenador es el que puede acceder a los mismos.

No se permite que dos usuarios, desde dos puestos de trabajo distintos, actúen en el mismo ordenador.

El DOS es el ejemplo más claro de un sistema monousuario.

Multiusuario:

Es aquél que soporta simultáneamente más de un usuario. De tal manera que una misma CPU (Unidad Central de Proceso) puede estar atendiendo a la vez a varios usuarios, que desde terminales diferentes, acceden al sistema.

Unos de los sistemas operativos multiusuario más importantes es el sistema operativo UNIX. También son muy importantes las diversas versiones de Microsoft Windows que soportan a varios usuarios: 95, 98, 2000, NT.

3. SISTEMA OPERATIVO MONOUSUARIO: DOS

Los nombres más utilizados son MS-DOS y PC-DOS, según provenga del acrónimo de Microsoft o de la denominación dada por IBM (Personal Computer). Son prácticamente idénticos, por lo que se puede utilizar el más extendido por el momento: MS-DOS.

Cuando nos referimos al DOS, estamos hablando de cualquiera de los nombres existentes, bien sea de los dos nombrados u otros semejantes (por ejemplo, DR-DOS).

La estructura del DOS se compone de varias rutinas que lo conforman. Contiene archivos de órdenes y archivos del sistema.

El intérprete de órdenes, llamado COMMAND.COM, es el responsable de las rutinas internas y externas del sistema operativo.

El DOS ha sido durante mucho tiempo el sistema operativo más utilizado dentro del mundo de los Ordenadores Personales (PC).

Tiene grandes limitaciones: monousuario, monotarea, pobre gestión de memoria.

Su gran arma es la gran difusión que tiene: la compatibilidad es uno de los aspectos más apreciados por el usuario en el mundo informática.

Algunas de las órdenes de DOS son: TIME, VOL, LABEL, DIR, CHKDSK, FORMAT, DISKCOPY, DISKCOMP, COPY, COMP, RENAME, DEL, TYPE, CLS, MODE, PROMPT, PRINT, BACKUP, RESTORE, RECOVER, MKDIR, CHDIR, RMDIR, TREE, PATH, GRAPHICS, etc.

3.1. Interfaz con el usuario

La interfaz con el usuario es la forma en que el sistema operativo permite al usuario ejecutar las órdenes.

En general, la palabra interfaz (del inglés, interface) se usa para designar un elemento que sirve como puente de comunicación entre dos partes. La interfaz con el usuario es el puente de comunicación que se establece entre el usuario y el sistema operativo (las rutinas del núcleo).

En algunos sistemas esta interfaz es simple y sencilla, trabajando en blanco y negro y en modo texto (por ejemplo, el indicador del sistema operativo DOS, que sólo presenta en pantalla el símbolo C:\>). Por el contrario, en otros sistemas operativos existe una interfaz con el usuario gráfica, basada en ventanas, menús y cuadros de diálogo y que se gestiona con el ratón.

El indicador o PROMPT del DOS siempre muestra la unidad activa, la unidad donde el usuario está en ese momento. La letra anterior al símbolo '>' corresponde a la unidad activa.

3.2. Comandos y parámetros

El DOS es un sistema operativo que se gestiona introduciendo órdenes o comandos en el indicador del DOS.

Un comando es simplemente una palabra en inglés que expresa la operación a realizar.

Todas las órdenes se activan introduciendo su nombre en el indicador o del sistema y pulsando la tecla Intro.

Las órdenes del DOS también se suelen llamar mandatos o comandos del DOS.

Ejecutar órdenes del DOS

Las órdenes del DOS se ejecutan escribiendo detrás del indicador el nombre de la orden y los posibles parámetros. Luego hay que pulsar la tecla Intro. Antes de pulsar Intro es posible utilizar las teclas Retroceso y Supr para borrar caracteres, las teclas Cursor Izquierda y Cursor Derecha para mover el cursor y la tecla Ins para activar el modo inserción.

A la hora de introducir una orden en el indicador hay que tener en cuenta tres **reglas**.

- Primera: Es indiferente que el nombre y los parámetros se introduzcan en mayúsculas o minúsculas.

- Segunda: Entre el nombre de la orden y el primer parámetro tiene que haber al menos un espacio en blanco. También es obligatorio dejar un espacio en blanco entre los parámetros (en algunos casos no es obligatorio, pero sigue siendo recomendable, a efectos de legibilidad).

- Tercera: La longitud máxima de la orden, contando nombre, parámetros y espacios en blanco, no puede exceder los 127 caracteres.

3.3. La sintaxis de las órdenes

Se llama sintaxis de una orden al esquema que muestra cómo se ejecuta correctamente dicha orden. En la sintaxis tiene que aparecer el nombre de la orden, los parámetros válidos y las conjugaciones posibles entre el nombre y los parámetros.

3.4. Gestor de memoria

El objetivo de un gestor de memoria es optimizar la memoria de un ordenador.

En MS-DOS la gestión se puede hacer de forma manual o mediante el gestor MEMMAKER.

MEMMAKER es una herramienta diseñada para configurar automáticamente la memoria de ordenadores 386 o superiores.

Se trata de un optimizador totalmente automático que apenas necesita de la intervención del usuario. Tanto si se hace automáticamente como manualmente tendremos en cuenta los siguientes aspectos y/o acciones:

- Analizar las órdenes del CONFIG.SYS que permanecen residentes en memoria (files, buffers, lastdrive) ajustando sus valores.
- Añadir al CONFIG.SYS una orden device para cargar el controlador HIMEM.SYS, que crea memoria alta.
- Añadir la orden DOS = HIGH para cargar el núcleo del DOS en memoria alta.
- Añadir en el CONFIG.SYS la orden device que carga el controlador EMM386.EXE, que crea los bloques de memoria superior y la orden DOS = UMB.
- Realizar un chequeo de las zonas libres de memoria superior e introducir en EMM386 los parámetros adecuados para conseguir la mayor cantidad posible de memoria superior.
- Examinar los controladores de dispositivos cargados en CONFIG.SYS con las órdenes device o devicehigh e intentar cargarlos en memoria superior.
- Examinar los programas residentes que existen en el fichero AUTOEXEC.BAT (estén o no cargados con la orden loadhigh) e intenta cargarlos en memoria superior.

Con todas estas operaciones conseguiremos la mayor cantidad de memoria convencional libre.

3.5. Configuración y personalización

Uno de los aspectos más importantes que tiene un sistema operativo es su flexibilidad o adaptación a las exigencias de los usuarios.

Por ello, todos los sistemas disponen de un conjunto de órdenes para la personalización y optimización del sistema.

El DOS dispone de dos ficheros encaminados a este fin dependiendo de la labor que realizan, así tenemos:

AUTOEXEC.BAT Encaminado a personalizar el sistema. Es un fichero especial dentro de todos los ficheros por lotes del MS-DOS. Se ejecuta cada vez que se enciende o reinicializa el ordenador. Debido a ello se puede utilizar para ejecutar aquellas órdenes que se desean cargar siempre que se empieza a trabajar.

CONFIG.SYS Su función es configurar el sistema.

Ejemplo de esas órdenes puede ser: MODE, DATE, TIME, PATH, ECHO OFF, PROMPT, KEYB.

- Orden **Mode**: Mediante la orden mode (modo) podemos definir las características del teclado.
- Orden **Date** y **Time**: Los comandos date y time presentan las instrucciones para que se introduzcan la fecha y hora correctas.
- Orden **Path**: Indica los directorios en los que el DOS deberá buscar el archivo de programa que se desee ejecutar. *Ejemplo*:

PATH c:\;c:\dos;c:\windows;

- Orden **PROMPT**: El indicador de órdenes o PROMPT puede cambiar su aspecto adoptando el que más le guste al usuario.
- Orden **KEYB**: Es casi obligatorio que se instale el programa KEYB, pues de lo contrario el teclado asumirá la plantilla de Estados Unidos y, por ejemplo, en lugar de la 'ñ' aparecerá el carácter ';'. KEYB es un programa residente que consume poco espacio en memoria (6K).

KEYB sp Instala la plantilla de teclado español

En DOS la optimización y/o configuración es posible gracias al fichero CONFIG.SYS.

- Orden **Device** o **Devicehigh**: Carga en memoria diversos controladores de dispositivos. Un controlador de dispositivo es un conjunto de rutinas, funciones, instrucciones (software) que permite comunicar un dispositivo físico con el S.O. Ejemplos de dispositivos son: Ratón, scanner, cd-rom, etc.

Hay controladores residentes (internos) se cargan al encenderse el ordenador, disquetes, pantalla, teclado...

Otros son externos o instalables. Necesitan ser instalados para que funcionen.

La orden device se encarga de realizar esta operación.

DEVICE = controlador

Dentro del DOS ya hay una serie de controladores que se pueden instalar. Estos controladores habitualmente tienen la extensión SYS.

Alguno de ellos son:

Controlador de Dispositivo	Función
HIMEM.SYS	Crea memoria alta (HMA) y extendida (XMS)
EMM386.SYS	Convierte memoria extendida en expandida (EMS)
ANSI.SYS	Gestiona características del monitor y del teclado
DISPLAY.SYS	Soporta páginas de códigos para la pantalla
SETVER.EXE	Define la versión con la que debe ejecutarse un programa
EGA.SYS	Necesario para tarjetas EGA y DosShell
DBLSPACE.SYS	Determina la posición final de dblspace.bin
RAMDRIVE.SYS	Crea un disco virtual en la memoria RAM

3.6. Gestión de dispositivos de almacenamiento masivo

Uno de los recursos más importante que posee un ordenador son dispositivos de almacenamiento masivo o discos.

La característica fundamental de los discos es que almacenan la información permanentemente, de forma que el usuario puede recuperarla en cualquier momento y trabajar con ella.

Las órdenes principales para gestionar discos DOS son las siguientes:

- Orden **FDISK**: Creación de particiones en el disco duro.

Antes de formatear un disco duro es preciso particionarlo. Las particiones sólo se pueden crear en los discos duros. Nunca en los disquetes.

Una partición es simplemente un conjunto de cilindros contiguos que forman una unidad independiente. La función de las particiones es doble. Por una parte, sirven para almacenar distintos sistemas operativos (cada uno en una partición). Por otra parte, dentro del mismo sistema operativo, cada unidad puede servir para dividir la unidad física del disco duro en varias unidades lógicas.

Conviene advertir de la peligrosidad de cualquier cambio en la tabla de particiones: la pérdida inevitable de los datos.

- Orden **FORMAT**: Formatear un disco.

Lo primero que hay que hacer con todo disco es formatearlo. Tal como vienen de fábrica, los discos son solamente dispositivos magnéticos sin estructura que pueda almacenar los datos. Para poder utilizarlo hay que crear una estructura lógica (sector de arranque, FAT, directorio raíz, y espacio para datos en el caso del DOS).

La orden FORMAT permite realizar cuatro tipos distintos de formateos:

1º. Formateo Seguro: (es el sistema por defecto). Se formatea sin borrar la información almacenada.

2º Formateo Rápido (/Q): Pone a cero la FAT pero no verifica la integridad de los sectores del disco.

3º. Formateo Incondicional (/U): Destruye los datos del disco sin posibilidad de recuperación.

4º. Formateo Ultrarrápido (/U /Q): No guarda los datos antes de formatear y pone la FAT a 0.

La orden FORMAT con el parámetro /S permite crear un disco de sistema del cual poder arrancar el sistema.

- Orden **DISKCOPY**: Copiar disquetes.

La orden DISKCOPY solo sirve para copiar disquetes, nunca discos duros. Realiza una copia exacta del disco origen al disco de destino incluyendo ficheros, directorios y subdirectorios.

- Orden **CHKDSK**: Chequeo de la estructura lógica de los datos.

La orden CHKDSK comprueba la integridad de los datos y nos muestra información relativa al estado actual del disco y la memoria (bytes libres, bytes ocupados, byte en sectores defectuosos, etc.).

- Orden **SCANDISK**: Chequeo lógico y físico de un disco

Es la nueva herramienta para verificar la integridad de los datos. Además SCANDISK realiza un recorrido físico del disco para detectar posibles sectores defectuosos.

3.7. Órdenes para la gestión de ficheros

- Orden **TYPE**: Visualizar ficheros de textos.

Un fichero de texto es aquel que contiene datos que se interpretan según el código ASCII.

- Orden **PRINT**: Imprimir ficheros de textos.

La orden PRINT envía el/los fichero/s que especifiquemos a la cola de impresión (lista que contiene todos los ficheros que se han de imprimir). Mediante una cola de impresión nuestro fichero es enviado a la lista a la espera su turno de impresión. Mientras nosotros podemos seguir trabajando en el ordenador con otros programas. *Ejemplo* de uso:

PRINT ayuda.txt Imprimir en la impresora el fichero ayuda.txt

- Orden **REN**: Renombrar Ficheros.

Ejemplo:

REN PROGRAM.EXE PROGRAMA.EXE

- Orden **DEL**: Elimina uno o más ficheros.

La orden DEL, igual que ERASE o DELETE, elimina uno o más ficheros del disco. *Ejemplo:*

DEL *.TMP Elimina todos los ficheros que tengan la extensión TMP.

- Orden **COPY**: Copia ficheros.

Es una de las órdenes más utilizadas. Es la orden por excelencia para realizar copias de ficheros entre unidades y directorios. *Ejemplo:*

COPY f1.txt f2.txt Realiza una copia del fichero f1.txt al fichero f2.txt.

- Orden **ATTRIB**: Ver y/o cambiar los atributos de un fichero.

Todos los ficheros a parte de la fecha, el tamaño, el nombre; tienen otra característica: el atributo.

Éste puede ser: R = Sólo lectura, S = Sistema, H = Oculto, A = Archivo.

Para activar un atributo se debe especificar delante el signo +. Para desactivarlo se indica precedido por el signo -.

Ejemplo: Activar el atributo de Archivo a todos los ficheros

ATTRIB *.* +A

3.8. Órdenes para la gestión de directorios y subdirectorios

- Orden **DIR**: Ver el contenido de un directorio.

La orden DIR es la orden más utilizada dentro del DOS. La sintaxis es bastante compleja pero la mayor parte de los casos su empleo es bastante fácil y se reduce a tres o cuatro opciones.

La información que nos muestra la orden DIR hace referencia al nombre del volumen, ruta del directorio completa, las características de cada uno de los ficheros, el espacio que ocupan y el espacio libre total del disco.

- Orden **MD**: Crear un directorio

El único directorio que debe existir en un disco es el raíz; los demás los crea el usuario.

Ejemplo:

MD CARTAS crea el directorio CARTAS en el directorio activo.

- Orden **RD**: Borrar un directorio.

La orden RD o RMDIR borra un directorio teniendo en cuenta que:

1. No se puede borrar un directorio si no está vacío.
2. No se puede borrar el directorio activo.
3. No se puede borrar el directorio raíz.

- Orden **CD**: Cambiar o visualizar el de directorio activo.

La orden CD o CHDIR nos permite cambiar de directorio activo o visualizar el directorio activo.

La orden CD sin nombre de directorio nos muestra cual es nuestro directorio activo.

En el caso de especificar directorio, cambia de directorio activo.

Ejemplo:

CD	Nos muestra el directorio activo.
CD C:\WINDOWS	WINDOWS que parte del raíz pasa a ser nuestro directorio activo.

- Orden **TREE**: Visualiza al Árbol de Directorios.

Sintaxis: TREE [directorio] [/F] [/A]

Ejemplo: TREE \ Muestra todo el árbol de la unidad por defecto.

- Orden **DELTREE**: Borrar directorios no vacíos.

Sintaxis: DELTREE [directorio]

Ejemplo: DELTREE \COPIA Borra el directorio COPIA, aunque no esté vacío.

- Orden **MOVE**: Renombrar un directorio.

Para cambiar de nombre un directorio existente se usa MOVE. La orden MOVE no puede renombrar el directorio activo.

Sintaxis: MOVE direc_antiguo direc_nuevo

Ejemplo: MOVE CTA1 CTA2 CTA1 ahora se llama CTA2

3.9. Recuperación de la información

- Orden **UNDELETE**:

Es muy habitual que el usuario borre un fichero accidentalmente, ya sea desde el indicador del DOS o desde un programa de aplicación. Para recuperar ficheros borrados el DOS 6.2 incorpora un programa UNDELETE que se ejecuta desde la línea de órdenes del DOS.

La orden **UNDELETE** puede usar tres métodos distintos para recuperar ficheros borrados accidentalmente: método estándar, método registro y método centinela.

- Orden **UNFORMAT**

Además de la posibilidad de recuperar ficheros borrados, el DOS permite desformatear discos; es decir, recuperar datos de discos formateados.

UNFORMAT A:

3.10. Copias de seguridad

- Orden **MSBACKUP**

En un sistema monousuario los medios para restringir el acceso al mismo usuario son muy elementales, casi inexistentes. Solo un usuario accede a la información y por tanto apenas existen limitaciones para acceder a cualquier archivo. Las técnicas de protección y seguridad se basan sobre todo en realizar frecuentemente copias de seguridad.

La herramienta **MSBACKUP.EXE** del MS-DOS efectúa copias de seguridad o restaura uno o más archivos de un disco a otro.

Con ella podremos realizar:

- Una copia de seguridad de todos los archivos de un disco.
- De los archivos que hayan cambiado desde que se creara la última copia de seguridad.

Podremos programar la creación de copias de seguridad de tal manera que se creen automáticamente en intervalos regulares y restaurar archivos de los que se hicieron copias de seguridad.

3.11. Compresor de disco

A partir de la versión 6 de MS-DOS, éste incorpora una utilidad nueva y bastante útil. Se trata del compresor de disco **DoubleSpace**, con el que podremos doblar la capacidad de nuestro disco duro. Este aumento se consigue mediante un programa residente que comprime automáticamente todos los datos antes de grabarlos en el disco y los descomprime cuando algún programa los solicita.

Para comprimir una unidad de disco duro basta teclear la orden **DBLSPACE**. Nos aparecerá una pantalla y mediante un sistema de menús iremos seleccionando las opciones deseadas.

3.12. Desfragmentador

Los ficheros del DOS se almacenan en el disco en unidades llamadas clusters que no necesitan ser secuenciales (un fichero puede empezar en el cluster 6, continuar en el cluster 1.305 y terminar en el cluster 27). Cuando los ficheros están muy fragmentados (divididos en clusters muy alejados entre si) el sistema tarda más tiempo en acceder a ellos. Si un fichero tiene sus clusters repartidos por diferentes partes del disco duro, la unidad de disco tendrá que desplazar el cabezal de lectura/escritura de una parte a otra del disco, con el consiguiente tiempo perdido en mover constantemente el cabezal.

El programa **DEFRAG** desfragmenta un determinado disco, es decir, recompone el disco almacenando los ficheros en clusters secuenciales, lo que acelera el rendimiento general del disco.

Se puede utilizar **DEFRAG** tanto en disquetes como en discos duros, pero sólo se aprovecha realmente en los discos duros.

A la hora de realizar su tarea, DEFRAG puede utilizar dos métodos de desfragmentación. En la documentación y el programa estos dos métodos se llaman Optimización plena y Desfragmentar sólo archivos.

4. SISTEMA OPERATIVO MULTIUSUARIO: UNIX

Es conceptualmente multiusuario. Se crea partiendo del lenguaje de alto nivel C (portable a distintos procesadores) desarrollado por Dennis Ritchie. El sistema operativo Unix es un sistema de tiempo compartido y, por lo tanto, multiusuario, que se puede ver como un conjunto de capas.

El sistema operativo LINUX es una versión completa de UNIX considerada de dominio público. Cuenta con potentes servicios de comunicaciones entre ordenadores con estructura cliente/servidor: SLIP (TCP/IP a través de puerto serie), PLIP, PPP, NFS. Además, incluye un completo servidor de X-Windows (Xfree86), y controladores aceleradores para multitud de tarjetas de vídeo.

4.1. Comandos y metacaracteres

El formato general de los comandos en UNIX es el que sigue:

<comando> [argumento] ...

Donde:

- Comando es el nombre del comando, y es requerido siempre.
- Los argumentos pueden necesitarse o no, según sea el comando.
- Los nombres de comandos en UNIX son siempre en minúsculas y no pueden ser abreviados.
- Los argumentos pueden ser de dos tipos:

- * Opciones o adverbios. Van precedidos de -.
- * Nombres de ficheros.

Los metacaracteres son caracteres que, además de representarse a sí mismos, tienen un significado especial: Se utilizan para sustituir a uno o más caracteres.

- ? - Sustituye a un solo carácter. *Ejemplo* de uso: Is a?, Is aa??a.
- * - Sustituye a cualquier cadena de caracteres (desde nulo o cero hasta varios caracteres).
Ejemplo: Is *, Is *.x, Is fiche*, Is *den*.

4.2. Entrada y salida del sistema

Para entrar en UNIX hay que identificarse. El usuario debe dar su nombre, aquel que se ha elegido como identificativo (**login**), no es un nombre real sino un alias, normalmente corto, y en minúsculas.

Acto seguido, es necesario dar la contraseña o clave, más conocida como **password**. Si el nombre es identificado, y la clave es correcta, se accede al sistema. Por el contrario, si algo falla, bien el login o el password, aparece "Login incorrect".

Para abandonar el sistema, lo habitual es mediante el comando **exit**, o el comando **logout**, o simplemente pulsando Ctrl-D.

En UNIX hay un concepto importante: es el de *consolas virtuales*. Se denomina consola del sistema al monitor y al teclado que está conectado directamente. Pero LINUX, al igual que otras versiones UNIX, tienen lo que se denominan consolas virtuales, VC, que permiten que un usuario tenga más de una sesión de login al mismo tiempo. Para acceder a una nueva consola virtual se utiliza la combinación de teclas *alt-tecla de función*.

Así para abrir una consola virtual después de haber entrado en el sistema se pulsaría alt-F2, y para conmutar a la primera alt-F1.

En este caso para abandonar el sistema se deben cerrar todas las sesiones abiertas.

4.3. Usuarios y grupos en Unix

Como hemos visto para que un usuario pueda entrar en UNIX, éste debe ser identificado por el sistema, para lo cual el usuario debe contar con una cuenta abierta en el sistema, esto significa que el sistema tiene los datos necesarios para verificar la entrada del usuario. Los datos con los que cuenta el sistema son:

Nombre: Nombre para el sistema.
Clave: Clave personal de acceso al sistema.
UID: Número de identificación del usuario.
GID: Número de identificación del grupo.
Directorio: Directorio inicial al entrar al sistema.
Nombre completo.
Login shell.

Dentro de los usuarios se tienen las siguiente *jerarquías*:

a) Superusuario, supervisor, o root:

Es el administrador del sistema. Tiene todos los privilegios.

b) Usuarios normales:

Son los usuarios normales del sistema. Se pueden asignar en grupos, que pueden tener una serie de propiedades comunes.

c) Usuarios especiales:

Asignados a tareas específicas por el sistema, generalmente de información o manejo de aplicaciones ya instaladas de uso común a usuarios externos o internos.

Desde el punto de vista del usuario, se ven los siguientes usuarios:

El propio usuario:	u
El conjunto de usuarios del mismo grupo	g
El resto de usuarios	o

El comando **useradd** o **adduser**, según versiones, añade un nuevo usuario al sistema, así como permite modificar algunos de sus parámetros: grupo al que pertenece, directorio de trabajo, password, etc.

Mediante **userdel** se puede suprimir un usuario del sistema, con la opción **-r**, que permite borrar todo su directorio de home (personal).

Para gestionar los grupos en UNIX, se utilizan básicamente los comandos **groupadd** y **groupdel**. Con **groupadd** creamos un nuevo grupo al sistema, con **groupdel** podemos suprimir un grupo.

4.4. Gestión de permisos

Uno de los aspectos más importantes de un sistema operativo multiusuario es la gestión de permisos de acceso o restricciones sobre la información.

Dado que UNIX es un sistema multiusuario, debe contar con algún mecanismo para proteger la información (ficheros y directorios) de un usuario particular, del acceso no autorizado de otros usuarios.

Los permisos se dividen en tres categorías, lectura, escritura y ejecución. Así a cada tipo de usuario (u, g, o) le corresponderá una triada donde se refleje sus privilegios de lectura, escritura y ejecución con respecto a un fichero o un directorio particular.

- El *permiso de lectura* **r**:

Permite a un usuario consultar el contenido de un fichero, o en el caso de los directorios listar el contenido del mismo.

- El *permiso de escritura* **w**:

Permite a un usuario modificar o borrar un fichero, en el caso de los directorios permiten crear o borrar ficheros.

- El *permiso de ejecución* **x**:

Permite al usuario ejecutar un fichero ejecutable o un script, en el caso de los directorios, tener permiso de ejecución permite entrar en el directorio, es decir hacer `cd` al directorio en cuestión.

Para ver los permisos de un fichero se tiene que utilizar la orden **ls -l**, esta orden es de las más utilizadas en UNIX, por este motivo existe un alias para este comando en la mayoría de los sistemas UNIX, este alias es **ll**.

4.4.1. La orden **chmod**

Para cambiar la cadena de permisos se utiliza el comando **chmod**. Sólo el dueño puede cambiar los permisos de un fichero. Este comando tiene dos sintaxis diferentes, una basada en cadenas mnemotécnicas, que es más intuitiva, y otra basada en combinaciones en base octal, que es más potente.

La sintaxis basada en cadenas mnemotécnicas es:

chmod [a, u, g, o] + | - [r, w, x] <nombre de ficheros>

Donde:

- a:** Significa todos los usuarios.
- u:** Significa el dueño.
- g:** Significa el grupo.
- o:** Significa el resto de los usuarios.
- +**: Significa activar un permiso.
- : Significa desactivar un permiso.
- r:** Significa permiso de lectura. **w:** Significa permiso de escritura.
- x:** Significa permiso de ejecución.

Ejemplos:

chmod a +r mifichero

Da a todos los usuarios permiso de lectura sobre el fichero mifichero.

chmod +r mifichero

Lo mismo que el anterior. Cuando no se indica a quien, se asume todos (a) por defecto.

chmod og-x mifichero

Se desactiva el permiso de ejecución para el grupo y para el resto de los usuarios.

chmod u+rwX mifichero

Se activan todos los permisos para el usuario.

4.5. Gestión de directorios y ficheros

cd Cambiar de directorio

Sintaxis: cd
 cd ..
 cd <nombre-directorio>

Descripción: Mediante cd se cambia a \$HOME (personal), cd .. cambia al padre, y cd <nombre-directorio> cambia a dicho directorio.

Ejemplos: cd, cd .., cd/ etc.

mkdir Crear un directorio

Sintaxis: mkdir <nombre-directorio>

Ejemplo: mkdir bak

rmdir Borrar un directorio

Sintaxis: rmdir <nombre-directorio>

Ejemplo: rmdir bak

pwd Informa del path absoluto del directorio actual partiendo del directorio raíz

Sintaxis: pwd

ls Lista el contenido de un directorio

Sintaxis: ls [opción][<nombre-directorio>/<ficheros>]

Las principales opciones son:

- a** Lista todos los ficheros. Incluyendo a aquellos que comienzan por “.”.
- c** Ordena por fecha de creación.
- l** Formato largo. Indica permisos, número de enlaces, propietario, grupo, tamaño, fecha y hora de creación y el nombre de los ficheros.

Ejemplos: ls -l /*, ls /etc.

En Linux existe un alias a ls -l que es dir. En otros muchos sistemas un alias para ls -l es ll.

cat Dirige el contenido de un fichero a la salida estándar

Sintaxis: cat <nombre fichero>

Ejemplo: cat /etc/profile

more Muestra el contenido de un fichero por pantalla, pero haciendo pausa cada vez que se llena la pantalla de información

Para ver la siguiente pantalla se pulsa la barra espaciadora, para ir avanzando línea a línea se pulsa INTRO. Y si se quiere salir se pulsa q.

Sintaxis: more <nombre-fichero>

Ejemplo: more /etc/profile

cp Copiar ficheros

Sintaxis: cp <fuente> <destino>

Ejemplo: cp mi-fichero bak/mi-fichero.copia

mv Mover ficheros

Sintaxis: mv <fuente> <destino>

Ejemplo: mv mi-fichero mío

rm Borrar ficheros

Sintaxis: rm [opciones] <ficheros>

Las principales opciones son:

-i Pregunta antes de borrar.

-r Borra un directorio de forma recursiva, borrando primero los ficheros que están en él.

Ejemplo: rm -i *.o

ln Permite crear vínculos entre archivos

Sintaxis: ln [opción]<fich_origen> <fich_destino>

La principal opción es:

-s Crea vínculos simbólicos para directorios.

Ejemplo: ln fichero1 fichero2

4.6. Procesos -PS-

Cada vez que se ejecuta un programa, arranca lo que se conoce como un proceso, que es una especie de nombre para referirnos a un programa en ejecución.

El comando **ps** muestra una lista de los procesos que en ese momento están corriendo de forma concurrente.

Cuando se ejecuta **ps** sólo se muestra información de los procesos que pertenecen al usuario y que se están ejecutando en ese momento. Si el que ejecuta el comando es el root entonces obtendrá información de todos los procesos de todos los usuarios.

ps es un comando que admite varias opciones, las principales son:

- I: Formato largo. Muestra todos los detalles.
- u: Formato usuario. Muestra el nombre de usuario y tiempo de comienzo.
- a: Muestra los procesos de todos los usuarios.
- x: Muestra los procesos sin control de terminal.

Ejemplo: /home/fran# ps

Resultado:

PID	TT	STAT	TIME	COMMAND
24	3	S	0:03	(bash)
161	3	R	0:00	ps

El PID es el identificativo (ID) del proceso (P), un único número que identifica a cada proceso. La última columna es el nombre de proceso que se está ejecutando. TT hace referencia al terminal que ha lanzado el proceso STAT indica como se encuentra (Running, Stop ...) y TIME muestra el tiempo consumido de CPU.

4.6.1. Foreground y Background

Los “jobs” pueden estar en foreground o en background. Un job que se encuentra en foreground, es el job con el que se va a interactuar, recibe la entrada del teclado y envía la salida a la pantalla (a no ser que se redireccione).

Los trabajos en background no reciben datos del terminal, y en general se ejecutan sin necesidad de interacción.

La manera más habitual de ejecutar un proceso en background es añadir el carácter & al final de la línea de comando:

comando&

4.6.2. Eliminar un proceso -kill-

La orden **kill** permite eliminar o matar un proceso. Para ello hay que dar simplemente el número de proceso (PID), o el número de job (en este caso se debe preceder el número de job del carácter %). Ejemplo: kill %1.

4.7. Editor vi

Un editor de textos es un programa imprescindible en cualquier sistema operativo que se utilice. Su misión es la de editar ficheros que contienen texto, ya sea un programa fuente, un fichero de configuración, un mensaje. Uno de los editores más populares e importantes dentro del entorno UNIX es vi.

Este editor no es el más fácil de utilizar, ni tiene nada de intuitivo, especialmente si el usuario proviene del mundo DOS.

El vi tiene tres *formas de trabajo*:

modo comando: Este modo permite desplazarse por el texto, borrar, insertar, cambiar texto.

modo edición: El modo texto es el modo que te permite entrar texto.

modo de comando
de línea: Modo editor de líneas.

Cuando se entra en vi, se está en modo comando, desde el cual se podrán realizar ciertas operaciones sobre el texto, y pasar a modo edición, mediante alguno de los comandos de edición (por ejemplo i). Estando en modo edición, para volver a modo comando se debe pulsar la tecla Esc.

Por lo que respecta al modo de comando de línea (o de última línea) es un modo especial para ejecutar comandos. Para acceder a este modo, se debe estar en modo comando y pulsar: Desde aquí se podrá por ejemplo salir salvando los cambios (**wq**) o salir sin grabar **q!**).

La forma más típica de iniciar el vi es:

vi <nombre-fichero >

4.8. Comandos de propósito general

man **Es la ayuda en línea**

Sintaxis: man <comando>
 man <Nº de Manual> <comando>
 man -k <comando>

Descripción: Este comando muestra por la salida estándar una ayuda del comando que se le pasa como argumento. Si no se indica número de manual se toma por defecto 1, pero se puede indicar un número de manual para obtener información desde otros puntos de vista del mismo comando. Si se utiliza -k se obtendrá toda la información existente sobre dicho comando.

Ejemplos: man man, man 2 write

date **Fecha y hora del sistema**

Sintaxis: date

cal **Facilita un calendario**

Sintaxis: cal mes año

Ejemplo: cal 01 1996

who **Da información de quién está en el sistema**

Sintaxis: who

who am i **Información de quién está en el terminal**

Sintaxis: who am i

hostname **Indica el nombre de la máquina que aloja el sistema**

Sintaxis: hostname

echo **Muestra mensajes, es decir representa el eco de sus argumentos**

Sintaxis: echo <mensaje>
 echo <\$variable>

Descripción: Este comando muestra el mensaje, que recibe como argumento, por pantalla, o muestra el valor de una variable de sistema.

Ejemplos: echo Hola amigos, echo 'Hola amigos', echo \$HOME.

passwd **Cambia la clave de acceso del usuario en la máquina en la que ejecuta este comando**

Sintaxis: passwd

grep **Busca las líneas que contengan el patrón**

Sintaxis: grep <cadena> <ficheros>

4.9. Gestión de impresión

El sistema UNIX incluye una colección de programas, denominados el sistema Ip, para imprimir archivos y documentos.

El sistema Ip es en sí mismo grande y complejo, pero afortunadamente su complejidad está oculta a los usuarios. De echo las tres órdenes básicas, **Ip**, **Ipstat** y **cancel**, es todo lo que se necesita saber para usar este sistema.

La orden básica para imprimir un archivo es Ip (line printer). *Ejemplo.* Para imprimir el archivo mifichero:

Ip mifichero

La orden Ipstat nos muestra información del estado de la impresora: trabajos que hay para imprimirse, las colas de impresión, etc.

La orden cancel se utiliza para eliminar o detener trabajos de impresión.

En algunos sistemas la utilidad **Ipadmin** permite gestionar todos los recursos de impresión: creación de colas, asignación de usuarios a colas, configuración de los puertos, etc.

5. BIBLIOGRAFÍA

Stallings, W. *Sistemas operativos* Prentice Hall, 2ed., 1997

Tanenbaum, Andrew S.
Sistemas operativos: Diseño e implementación
Prentice Hall, 1990