UD1

COMANDOS BÁSICOS DE LINUX

1. SISTEMAS DE FICHEROS DE LINUX

- Todos los sistemas operativos estructuran sus objetos en una estructura jerárquica en forma de árbol que contiene información sobre los diferentes objetos (directorio, archivos, links).
- Esta estructura permite contener distintos objetos en distintas localizaciones con el mismo nombre.
- Un directorio es un objeto destinado a contener otros objetos.
- Un archivo es un objeto destinado a contener información.
- El origen del sistema de archivos de LINUX se encuentra en el directorio root o "/"

1. DIRECTORIOS DE LINUX

/bin /sbin

Estos directorios contienen programas ejecutables que forman parte del sistema operativo. El directorio /sbin contiene las utilidades del sistema de archivos, particiones e inicio del sistema. El directorio /bin contiene el resto de comandos de la consola y utilidades varias.

/boot

Este directorio contiene la información necesaria para poder arrancar el sistema operativo. Entre otros archivos, aquí se encuentran los núcleos del sistema que se pueden iniciar y la configuración de gestor de arranque.

/etc

Este directorio contiene todos los archivos de configuración de nuestro sistema GNU/LINUX. Este directorio posee distintos subdirectorios que se utilizan para la configuración de los distintos elementos o servicios del sistema operativo.

1. DIRECTORIOS DE LINUX

/dev

Este directorio contiene archivos de dispositivos que permiten la comunicación con los distintos elementos hardware que tengamos instalados en el sistema; discos duros (como /dev/hda o /dev/sda), particiones de discos duros (como /dev/hda3 o /dev/sda6), unidades de CD-ROM (como /dev/scd0), disqueteras (como /dev/fd0), impresoras (como /dev/lp0), puertos serie (como /dev/ttyS0 o /dev/cua0), puerto PS2 (como /dev/psaux), tarjetas de sonido (como /dev/audio), etc.

/lib

Contiene las librerías que son necesarias durante el inicio del sistema operativo. La ventaja de usar librerías reside en que no es necesario integrar su código en los programas que las usan, reduciendo así el tamaño de los ejecutables. Cuando un programa necesita alguna de sus funciones, se carga la librería en memoria y puede ser usada por cualquier otro programa que la necesite, sin necesidad de volver a cargarla en memoria.

1. DIRECTORIOS DE LINUX

/mnt

Este directorio es típico de las distribuciones RedHat, y puede no estar presente en otras distribuciones. Su misión consiste en agrupar en un mismo lugar los puntos de montaje de diversos dispositivos. Este directorio contiene un subdirectorio adicional para cada una de las particiones o dispositivos disponibles en el sistema. Cuando accedemos a estos subdirectorios estamos accediendo a los dispositivos.

/home

Este directorio contiene los directorios personales "home" de todos los usuarios del sistema (menos el root). Los usuarios convencionales únicamente pueden escribir en su directorio "home".

/root

Este es el directorio personal del usuario root o súper usuario. Contiene básicamente la misma información que los directorios personales de los usuarios del sistema, pero orientada única y exclusivamente al usuario root.

1. DIRECTORIOS DE LINUX

/var

Su nombre procede de variable, y esa es la naturaleza de la información que contienen sus subdirectorios y archivos, como colas de impresión (/var/spool/lpd), correo electrónico, o archivos de registro creados por los distintos procesos del sistema.

/usr

Su nombre proviene de user y contiene una réplica de otros directorios del sistema operativo orientados a usuarios en lugar de al propio sistema operativo.

/usr/X11R6

Contiene todos los elementos que componen el entorno gráfico X Windows; binarios (/usr/X11R6/bin), librerías (/usr/X11R6/lib), manuales, etc.

/usr/bin

En este directorio se guardan los binarios o ejecutables de todas las aplicaciones orientadas al usuario.

1. DIRECTORIOS DE LINUX

/usr/src
Este directorio contiene el código fuente del núcleo del sistema GNU/LINUX y aplicaciones instaladas.
/tmp
Este directorio contiene diversos archivos temporales que son usados por los programas del sistema operativo.
/proc
Contiene los archivos de proceso. No son verdaderos archivos sino una forma de acceder a las propiedades de los distintos procesos que se están ejecutando en nuestro sistema. Para cada proceso en marcha existe un subdirectorio /proc/<número de proceso> con información relativa a ese proceso.

2. I-NODOS

- Estructura de datos propia de los sistemas Unix/Linux.
- Contiene información de los objetos del sistema de archivos (archivo regular, directorio, enlaces simbólicos):
  - Permisos de usuario
  - Fechas última modificación
  - Ubicación en el disco (NO el nombre)
- Cada inodo queda identificado por un número entero, único dentro del sistema de ficheros, y los directorios recogen una lista de parejas formadas por un número de inodo y nombre identificativo que permite acceder al archivo en cuestión: cada archivo tiene un único inodo, pero puede tener más de un nombre en distintos o incluso en el mismo directorio para facilitar su localización.

2. LINKS EN LINUX

Un link o enlace es un archivo especial que crea un atajo al archivo original situado en cualquier parte del sistema de archivos. Existen dos tipos de enlaces:

- **Links Simbólicos**. Es un pequeño archivo que contiene un puntero al archivo apuntado. Cuando se abre un enlace simbólico, Linux lee el puntero y abre el archivo apuntado.

- Los enlaces simbólicos pueden apuntar a sistemas de archivos diferentes, dispositivos diferentes o incluso a otros ordenadores conectados en red.

- La orden ls –l muestra una "l" en los links e informa a que archivo apunta el link.

- Cuando un archivo que posee un enlace simbólico se borra del sistema, el link no apunta a nada (link "stale")

2. LINKS EN LINUX

- **Links Hardware.** En un enlace hardware, un mismo objeto (con un único i-nodo) posee dos o más referencias.

Las referencias tienen diferentes nombres pero apuntan al mismo i-nodo, es decir al mismo objeto.

- Los enlaces hardware tienen dos importantes limitaciones:
  - 1º como comparten i-nodo, el archivo y el enlace tienen que estar en el mismo sistema de archivo.
  - 2º no pueden apuntar a directorios.

- Son menos fáciles de manejar y menos versátiles que los enlaces simbólicos. Por esta razón la mayoría de enlaces en sistemas UNIX son links simbólicos.

3. COMANDOS DE GESTIÓN DE FICHEROS Y DIRECTORIOS

Los objetos del sistema de archivos (archivos, directorios y links) son constantemente creados, leídos, modificados, copiados, movidos y borrados. La gestión de estos objetos es una tarea de las más importantes del administrador del sistema. A continuación se presentan las órdenes básicas del intérprete de comandos que nos permiten gestionar estos objetos.

- **pwd** (print working directory)

Sintaxis: pwd

La orden pwd muestra la ruta de acceso del directorio actual.

3. COMANDOS DE GESTIÓN DE FICHEROS Y DIRECTORIOS

- **cd** (change directory)

Sintaxis: cd directory

La orden cd cambia al directorio especificado en directory

Ejemplos:

$cd apache ⇗ cambia al directorio apache que se encuentra en el directorio actual.

$cd /apache ⇗ cambia al directorio apache que se encuentra en el directorio raíz o root de la jerarquía de directorios del sistema operativo.

$cd /apache/web ⇗ cambia al directorio web situado dentro apache.

$cd .. ⇗ cambia al directorio anterior.

$cd (sin argumento) /$cd ~ ⇗ cambia al directorio personal del usuario actual.

$cd ~alumno cambia al directorio personal del usuario alumno.

3. COMANDOS DE GESTIÓN DE FICHEROS Y DIRECTORIOS

- **ls (**list)

Sintaxis: ls [options] directory

Opciones:

-A Lista todos los archivos, incluidos los ocultos (En UNIX los archivos ocultos son aquellos cuyo nombre empieza con un "."), excepto los archivos "." y ".."

-l Lista los archivos en formato largo y muestra información detallada sobre ellos.

-R Lista de forma recursiva los contenidos de los subdirectorios.

-i Muestra el número de i-node de cada fichero.

-s Muestra el tamaño en KiloBytes junto a cada archivo.

-u Clasifica por fecha y hora del último acceso.

-t Clasifica por fecha y hora de la última modificación.

3. COMANDOS DE GESTIÓN DE FICHEROS Y DIRECTORIOS

- **cp** (copy)

Sintaxis: cp [options] file1 file2 ⇗ Copia file1 a file2. Si file2 existe y el usuario tiene los permisos apropiados el archivo será remplazado.

cp [options] files directorio ⇗ Copia uno o más archivos en directorio. Si no existe se mostrará un mensaje de error.

Opciones:

-f (force) Fuerza a sobrescribir los archivos existentes en el destino.

-i (interactive) Pregunta antes de sobrescribir cualquier archivo.

-p Mantiene toda la información del archivo; propietario, grupo propietario, permisos, hora y fecha. Sin esta opción, el archivo o archivos copiados tendrán la fecha y hora actual, los permisos, propietario y grupo propietario por defecto.

-R (recursive) Si en file1 se especifica un directorio, la opción –r o –R copia toda la jerarquía del directorio en el destino especificado.

-v (verbose) Muestra el nombre de cada archivo mientras se copia.

3. COMANDOS DE GESTIÓN DE FICHEROS Y DIRECTORIOS

- **mkdir** (make directory)

Sintaxis: mkdir [options] directory

Crea un directorio. El usuario tiene que poseer permisos de escritura en el directorio donde se creara el directorio.

Opciones:

-p Crea los directorios intermedios si estos no existen.

- **mv** (move)

Sintaxis: mv [options] source target

Mueve o renombra archivos y directorios. Si target no existe, source es renombrado. Si target existe, será sobrescrito. Si target es un directorio, source será movido dentro de ese directorio.

Opciones:

-f Fuerza a no preguntar si el target existe, eliminado los mensajes de advertencia.

-i Fuerza a preguntar antes de mover cualquier archivo.

3. COMANDOS DE GESTIÓN DE FICHEROS Y DIRECTORIOS

- **rm** (remove)

Sintaxis: rm [options] files

Elimina uno o más archivos del sistema. Para eliminar un archivo es imprescindible que el usuario tenga permiso de escritura en el directorio que contiene el archivo, pero no necesita permiso de escritura en el archivo. El comando rm también puede borrar directorios cuando se usan las opciones -r o –R.

Opciones:

-f Fuerza a no preguntar al borrar archivos sin permiso de escritura.

-i Fuerza a preguntar al borrar cada archivo.

-r Si file es un directorio, elimina recursivamente el contenido completo del directorio, incluidos los subdirectorios.

3. COMANDOS DE GESTIÓN DE FICHEROS Y DIRECTORIOS

- **rmdir** (remove directory)

Sintaxis: rmdir [options] directory

Borra directorios vacíos.

Opciones

-p Borra los directorios intermedios si estos están vacíos como resultado de la orden.

- **touch**

Sintaxis: touch [options] files

Cambia la fecha del último acceso o/y modificación de files. Si no se especifica ninguna opción se actualizarán ambas fechas (acceso y modificación). Sin opciones crea un archivo vacío.

Opciones:

-a Actualiza únicamente la fecha del último acceso del archivo.

-m Actualiza únicamente la fecha de modificación del archivo.

3. COMANDOS DE GESTIÓN DE FICHEROS Y DIRECTORIOS

-t No utiliza la fecha actual, sino el especificado a continuación mediante el formato de [[CC]YY]MMDDhhmm[.ss].

Ejemplos:

Modifica la fecha del último acceso al 12 de enero de 2001 a las 18 horas, 45 minutos.

$touch –ta 200101121845 file

- **ln** (link)

Sintaxis: ln [options] file link

ln [options] files directory

Crea enlaces entre archivos. En la primera forma se crea un enlace llamado link que apunta al archivo file. En la segunda forma, se crea un enlace dentro del directorio directory para cada uno de los archivos especificados en files.

Opciones:

-f Fuerza a sobrescribir los enlaces si existen previamente.

3. COMANDOS DE GESTIÓN DE FICHEROS Y DIRECTORIOS

-i Pregunta antes de crear cada enlace.

-s Crea un enlace simbólico. Por defecto crea enlaces hardware.

- **cat**

Sintaxis: cat [options] file

Muestra el contenido del archivo file.

Opciones:

-b Numera todas las líneas de salida que no están en blanco.

-n Numera todas las líneas de salida.

-s Reemplaza por una línea en blanco varias líneas en blanco adyacentes.

3. COMANDOS DE GESTIÓN DE FICHEROS Y DIRECTORIOS

- **head /tail**

Sintaxis: head [options] file / tail [options] file

Muestra las n primeras/últimas líneas del archivo file.

Opciones:

-n Indica las n primeras/últimas líneas del archivo

- **uniq**

Sintaxis: uniq [options] file

Compara las líneas y busca líneas únicas. Si las líneas son iguales, muestras sólo una de ellas.

Opciones:

-n Ordena los campos numéricos por su valor numérico.

-r Realiza una ordenación inversa (de mayor a menor).

3. COMANDOS DE GESTIÓN DE FICHEROS Y DIRECTORIOS

- **cut**

Sintaxis: cut [options] file

Recorta líneas especificando el número de caracteres:

Opciones:

-n Ordena los campos numéricos por su valor numérico.

-r Realiza una ordenación inversa (de mayor a menor).

4. METACARACTERES Y SELECCIÓN DE ARCHIVOS

Al manipular archivos con el intérprete de comandos, a menudo es necesario realizar una determinada operación con muchos archivos. Por ejemplo, en el desarrollo de un programa en C, es necesario usar la orden touch para forzar la compilación de todos los archivos.

Para realizar este tipo de operaciones de una manera rápida y simple, el intérprete de comandos dispone de varios metacaracteres. En vez de especificar el nombre de cada archivo, los metacaracteres substituyen parte del nombre del archivo. Los metacaracteres disponibles en GNU/LINUX se resumen en la siguiente tabla.

4. METACARACTERES Y SELECCIÓN DE ARCHIVOS

metacarácter	descripción
*	Sustituye cualquier número de caracteres, incluido cero caracteres. Por ejemplo, x* se corresponde con los archivos o directorios x, xy, xyz, x.txt, xy.txt,...
?	Sustituye únicamente un carácter. Por ejemplo, x? se corresponde con xx, xy, xz, pero no con "x" ó "xyz".
[caracteres]	Sustituye un único carácter que este listado entre los corchetes. Por ejemplo, x[yz] se corresponde con "xy" ó "xz".
[!caracteres]	Sustituye un único carácter que **NO** este listado entre los corchetes. Por ejemplo, x[!yz] se corresponde con todos los archivos o directorios cuyo segundo carácter no es ni "y" ni "z".
[a-z]	Sustituye un único carácter que este dentro del rango especificado entre los corchetes. Por ejemplo, x[0-3] se corresponde con x0, x1, x2 y x3, pero no se corresponde con "xx" ó "x4".
[!a-z]	Sustituye un único carácter que **NO** este dentro del rango especificado entre los corchetes.

5. CORRESPONDENCIA COMANDOS LINUX vs MSDOS

DOS	Comando Linux
cd directorio	cd directorio
dir, dir/w	ls, ls -l
chdir (directorio actual)	pwd
del (borra un archivo)	rm
deltree (borra un directorio y su contenido)	rm -r
copy	cp
xcopy (copia todo el contenido de un directorio)	cp -R
rename, move	mv
type (imprime el contenido de un archivo a la pantalla)	cat
help, [comando] /?	man
cls (limpia la pantalla)	clear
find (busca por una palabra(s) en un determinado archivo)	grep
edit nombre-de-archivo	gedit nombre-de-archivo
mem (muestra la memoria disponible)	free, top
scandisk	fsck
pkzip (crea un paquete de archivos)	tar, utilizado en conjunto con gzip para compresión
ipconfig (visualiza dirección IP y configuración de red)	ifconfig
route print (muestra tablas de ruteo)	route -n

PRACTICA UD 1. GESTIÓN DE ARCHIVOS

Paso 1. Abrir un terminal. Examinar el directorio en el cual nos ha situado el sistema por defecto. Este directorio recibe el nombre de personal y en principio es donde tenemos permisos para crear carpetas y dejar nuestros documentos.

Paso 2. Visualizar el contenido del directorio personal.

Paso 3. Acceder desde el directorio personal a los siguientes directorios

/etc /root /home /boot /bin

- de forma directa o absoluta (todo el path)

- de forma relativa

Paso 4. Explicar el contenido de esos directorios del S.O.

Paso 5. Desde cualquier lugar del sistema de directorios, acceder al directorio personal del usuario actual, utilizando el carácter ~.

PRACTICA UD 1. GESTIÓN DE ARCHIVOS

Paso 6. Crear en el directorio personal del usuario actual los siguientes directorios: test y practica_de_gestion_de_archivos

NOTA: En UNIX los nombres de archivo y directorio no pueden contener espacios en blanco.

Paso 7. Entrar en el directorio test y crear el directorio linux.

Paso 8. Situarse en el directorio personal del usuario. Con una sola orden crear el directorio test.1 y dentro de test.1 crear el directorio test.2.

Paso 9. Crear los siguientes archivos mediante la orden del sistema operativo ls –al > "archivo" dentro del directorio practica_de_gestion_de_archivos.

test testa testA testB TestBa testC
testCa testCb testCc TestDa
testDb testDc testDd testDx TestDy

PRACTICA UD 1. GESTIÓN DE ARCHIVOS

Paso 10. Cambiar el directorio actual de trabajo a test. Sin cambiar de directorio copiar todos los archivos del directorio practica_de_gestion_de_archivos de 6 caracteres terminados en a, al directorio test.2. Indica los ficheros copiados. Especificar la opción verbose y force en el comando.

Paso 11. Situarse en el directorio personal del usuario actual. Copiar del directorio practica_de_gestion_de_archivos a test.2 los archivos de 6 caracteres donde el quinto carácter no sea ni una a ni una b. Indica los ficheros copiados. Especificar la opción interactive de la orden.

Paso 12. Borra los ficheros copiados y copia sólo ahora los archivos de 5 caracteres donde el quinto carácter no sea ni una a ni una b. Indica los ficheros copiados

Paso 13. Listar el contenido (incluido los archivos ocultos) de los directorios practica_de_gestion_de_archivos y test.2 sin salir del directorio personal del usuario.

PRACTICA UD 1. GESTIÓN DE ARCHIVOS

Paso 14. Mover del directorio practica_de_gestion_de_archivos al directorio linux todos los archivos que NO terminen en a, b, c y d usando el metacarácter de intervalo. Especificar la opción force y verbose de la orden.

Paso 15. Cambiar el directorio actual de trabajo a practica_de_gestion_de_archivos. Renombrar el archivo test a test.txt. Si no existe test, crearlo mediante el editor vi.

Paso 16. Cambiar la fecha de modificación del archivo test.txt a 20 de diciembre de 1973 11 horas 35 minutos de la mañana.

Paso 17. Crear un enlace simbólico a test.txt que se llame test.txt.link.

Paso 18. Visualizar el fichero test.txt y test.txt.link mediante la orden cat. ¿Hay alguna diferencia al visualizar los dos archivos?

Paso 19. Eliminar el contenido de los directorios test, test.1 y practica_de_gestion_de_archivos. Especificar la opción verbose, interactive, forzado y recursive del comando.