

Comandos básicos Linux I

Comandos Básicos I

- \$ date
- \$ cal \$ cal 12 1970 \$cal -3 12 1970
- Encadenado de comando: \$ date;pwd;cal
- Visualizar texto por pantalla:
 - \$ echo –e “Hola\nme\llamo\nPepe”
 - \n: Salto de línea. \\: barra
 - \t: Tabulación horizontal.
 - \b: carácter atrás.

Comandos Básicos I

- Comando internos y externos:
 - Comando externos: son los programas binarios almacenados en HD y cargados en memoria, para poder ejecutarlos.
 - Comando internos: son los que se ejecutan dentro de la Shell.
 - \$ type comando

Comandos Básicos I

- Atajos para comandos:
 - [Ctrl] C: Interrupción del programa, se termina.
 - [Ctrl] Z: Para el programa.
 - \$ find /
 - \$ ps -aux
 - [Ctrl] D: interrupción de datos en un prompt >
 - \$ mail

Comandos Básicos I

- Histórico de comandos:
 - \$ history
 - \$!numero comando: ejecuta ese comando.
 - Se almacena en ~/.bash_history
 - \$ history -c: ¿borra histórico?

Comandos Básicos I

- Las rutas:
 - El nombre de la ruta o Path name de un fichero es la concatenación, desde la raíz /, de todos los directorios que se deben cruzar para acceder a él, estarán separados cada uno por el carácter /
 - /home/usuario/practicas/pract1.txt
 - Una ruta absoluta no tiene ni . ni ..

Comandos Básicos I

- Ruta relativa:
 - Partimos de la posición en el directorio actual.
 - Entradas particulares de directorios:
 - . Representa el directorio actual. Implícito.
 - .. Representa el directorio de nivel superior.
 - La ruta relativa:
 - Describe una ruta desde la posición actual.
 - Puede contener . O ..

Comandos Básicos I

- Documentos/personal
- ./Documentos/personal
- Donde: . Representa el directorio activo
- /home/usuario/documentos/./

Comandos Básicos I

- Alias del directorio personal: ~ (Virgulilla)
- Las rutas pueden ser relativas a la virgulilla, pero esta no debe ir precedida por carácter alguno.
 - \$ cd ~/docu
 - \$ /~ Error

Comandos Básicos I

- `cd`:
 - Change Directory.
 - `cd .` No se mueve.
 - `cd ..` Sube un nivel.
 - `cd` sin argumento vuelve al directorio personal del usuario.
 - `cd ../home`
 - `cd ../../usuario`

Comandos Básicos I

- Listado de ficheros y directorios:
 - ls
 - ls -l es la opción más completa.

Ejercicio:

Utiliza man ls para ver las opciones.

Anota las opciones más importantes.

¿Qué ficheros son los que comienzan por un punto .?.

Comandos Básicos I

```
drwxr-xr-x 5 usuario usuario 4096 oct  3 18:38 pruebas
-rw-r--r-- 1 usuario usuario 62241 oct  3 18:22 resultado
```

- 1 -: ordinario, d: directorio, l: vínculo simbólico.
- 2 contador de vínculos.
- 3 Propietario del fichero que suele ser el creador.
- 4 El grupo al cual pertenece
- 5 El tamaño del fichero en Bytes.
- 6 Fecha de la última modificación.
- 7 Nombre del fichero.

Comandos Básicos I

- Crear ficheros vacíos:
 - `$ touch nombre`
 - ¿Qué otra finalidad tiene el comando?.
- Crear directorios
 - `mkdir nombre`
 - `mkdir -p dir1 dir2 dir3`
- Suprime directorios:
 - `rmdir`

Comandos Básicos I

- `rmdir` no suprime una estructura.
- El directorio a eliminar no puede tener ni ficheros ni directorios, aunque los subdirectorios están vacíos.
- Para suprimir una estructura utilizar:
 - `$ rm.`

Comandos Básicos I

- `rm` : Suprime uno o varios fichero y estructuras completas.
- ALTAMENTE PELIGROSO.
- `rm [opciones] fic1 fic2 fic3`
- `-i`: Confirmación para cada uno de los ficheros a eliminar.
- `-r`: siguiente parámetro debe ser un directorio. La supresión es recursiva. Se suprimen todos los niveles inferiores.
- `-f`: fuerza la supresión.
- Muy práctico pero peligroso combinar: `-rf`

Comandos Básicos I

- Copiar ficheros:
 - `$ cp fich1 [fich2 fich3] Destino`
 - `-i`: Pide confirmación de escritura para cada fichero.
 - `-r`: recursivo, copia un directorio y todo su contenido. `$ cp -r dir1 dir2`
 - `-p`: se preservan los permisos y las fechas.
 - `-f`: forzar la copia.

Comandos Básicos I

- Ejercicio:
 - Crear un fichero de nombre:-i
 - Tratar de borrarlo.

Comandos Básicos I

- Mover y renombrar ficheros:
 - \$ mv
 - \$ mv fich1 fich.old
 - \$ mv fich2 dir/fich2
 - \$ mv fich2 dir/fich2.old

Comandos Básicos I

- Comodines:
 - Son caracteres de sustitución.
 - *: Sustituye una cadena de longitud variable.
 - ?: Sustituye un único carácter.
 - [a-g]: Un carácter entre la serie o rango de caracteres.
 - [!a-g]: Inversión de la búsqueda.
 - [^a-g]: Inversión de la búsqueda.
 - {a,g,h}: Un carácter de los especificados.

Comandos Básicos I

- Comodines:
 - Pueden ser utilizados dentro de todos los argumentos que representen ficheros o rutas.
 - Se deben cerrar algunos caracteres especiales.
 - \ permite cerrar un único carácter.
 - \$ ls fichero\ * . Se pone \ para que interprete el carácter en blanco.
 - Las comillas “”, permiten la interpretación de caracteres especiales o de variables, dentro de una cadena.
 - Los apóstrofes ’ cierran todos los caracteres especiales de una cadena o fichero.

Comandos Básicos I

- Vínculos Simbólicos (SoftLink):
 - Es un fichero especial que contiene como información la ruta a otro fichero.
 - Es por tanto como un “Alias”.
 - \$ ln -s fichero vínculo.
 - Los vínculos se comportan como el fichero al que apunta, con los mismos permisos y propiedades.

Comandos Básicos I

- Vínculos Simbólicos (SoftLink):
 - Si el fichero apuntado es un programa, ejecutar el vínculo ejecutará el programa.
 - Si el fichero apuntado es un fichero especial, periférico, se ve el vínculo como periférico.
 - Si el fichero apuntado es un directorio, un cd llevará a la entrada en ese directorio.

Comandos Básicos I

- Vínculos Simbólicos (SoftLink):
 - La eliminación del vínculo no implica la eliminación del fichero apuntado.
 - La eliminación del fichero apuntado no implica la eliminación del vínculo. Vinculo apunta al vacío.
 - Los derechos del fichero apuntado prevalecen sobre los del vínculo.

Comandos Básicos I

- Vínculos Simbólicos (SoftLink):
 - `$ touch prueba1`
 - `$ ln -s prueba1 vinculo prueba1`
 - `ls -l`
 - ¿Qué derechos tiene tanto el vínculo como el fichero apuntado?.

Comandos Básicos I

- Redirecciones:
 - Capacidad de redireccionar la visualización de la pantalla hacia:
 - Fichero.
 - Impresora.
 - Cualquier otro periférico.
 - Capacidad de redireccionar los mensajes de error hacia otro fichero.
 - Sustituir la introducción vía teclado por un fichero.

Comandos Básicos I

- Redirecciones:
 - Cualquier flujo de datos pasa por un canal y podrá ser desviado hacia otro destino u otra fuente.
 - Canal de entrada por defecto es el teclado.
 - Canal de salida por defecto es la pantalla.
 - Los errores se direccionan por defecto a la pantalla.

Comandos Básicos I

- Redirección de salida: (>,>>)
 - \$Comando > fichero
 - \$ls -l > listado.txt
 - Si no existe listado.txt, se creará.
 - Si existe listado.txt, se sobrescribirá, incluso si el comando introducido fuera incorrecto.
 - El Shell en primer lugar crea el fichero y posteriormente introduce el resultado del comando en listado.txt.

Comandos Básicos I

- Redirección de salida: (>,>>)
 - Si lo que se quiere es añadir en el fichero se empleará >>
 - `$ls -l >> listado.txt`
 - De esta forma se añadirán nuevos datos a listado.txt sin sobrescribirlo.
 - `$ifconfig eth0 > milP.txt`

Comandos Básicos I

- Redirección de entrada: (<)
 - Los comandos que esperan datos o parámetros pueden también recibirlos desde un fichero usando <.
 - \$wc < fichero.txt
 - \$tr ' ' '\t' <prueba.txt
 - \$mail usuario < carta.txt

Comandos Básicos I

- Redirección de entrada: (<<)
 - Herescript o Here Document, permite la inserción de un texto hasta un punto especificado y el envío de su resultado a un comando o un filtro.
 - \$tr ' ' '\t' <<END
 - Ahora introducimos un texto. Al terminar de introducir el texto pondremos END, se le pasa al comando tr el texto y sustituirá espacios en blanco por tabulaciones.

Comandos Básicos I

- Redirección de entrada: (<<)
 - Herescript o Here Document
 - \$tr '[a-z]' '[A-Z]' <<END
 - >En un lugar de la Mancha.
 - >END

Comandos Básicos I

- Canales estándares:
 - Canal de entrada estándar stdin descriptor 0
 - Canal de salida estándar stdout descriptor 1
 - Canal de error estándar descriptor 2.
 - Se puede redireccionar el canal de error estándar hacia otro fichero.
 - `$ls t* y*`
 - `$ls t* y* 2> error.txt`
 - No se muestra por pantalla los errores sino que los reencaminamos al fichero error.txt

Comandos Básicos I

- Canales estándares:
 - Podemos redireccionar los dos canales de salida a un único fichero, utilizando &>.
 - `$ls z* y* > resultados.log`
 - Muestra los errores en pantalla y en resultados.log lo acertado de ls.
 - `$ls z* y* 2> errores.log`
 - Muestra los resultado acertado de ls por pantalla, y se introduce en errores.log los errores de ls.

Comandos Básicos I

- Canales estándares:
 - `$ls z* y* > resulconError.log 2>&1`
 - En pantalla no se muestra nada estando todos los resultados acertado como errores de ls en resulconError.log
 - En primer lugar se indica que la salida de error se desvíe a la salida estándar, posteriormente se redirecciona el resultado de la salida estándar del comando ls hacia el fichero resulconError.log. El fichero tendrá tanto la salida estándar y la salida de error.

Comandos Básicos I

- Canales estándares:
 - `$wc < fichero.txt > resultado.txt`
 - `$cat resultado.txt`
- `$ls -l > /dev/null 2>&1`
- `$cat /dev/null`

Comandos Básicos I

- Apertura de canales:
 - Los canales estándares son los numerados como 0,1,2.
 - `exec` permite abrir otros siete canales, 3-9.
 - El número de canales en total son del 0-9.
 - Los canales están abiertos para la entrada y para la salida.
 - `$exec 5>canal5.txt`
 - `$ls -l >&5`
 - `$cat canal5.txt`
 - Todo lo que se escriba por el canal 5 irá a canal5.txt
 - `$exec 5>&-` #Cierro el canal 5

Comandos Básicos I

- Apertura de canales:
 - `$exec <6canal6.txt`
 - `$cat <&6`
 - `$exec 6>&-` #Cerramos el canal.
- Para cerrar el canal lo redireccionamos a un seudocanal (canal de cierre -)

Comandos Básicos I

- Tuberías o Pipelines:
 - Permiten redireccionar el canal de salida de un comando hacia el canal de entrada de otro comando.
 - Carácter |
 - `$ps -aux | grep gimp*`
 - `$ifconfig eth0 | grep 192.168.1.100`

Comandos Básicos I

- Búsqueda de ejecutables:
 - \$ whereis nombre
 - \$ which comando
 - \$ locate fichero

Comandos Básicos I

- Búsqueda de ejecutables:
 - \$ whereis nombre
 - Busca en rutas de binarios -b
 - Entradas en el manual -m
 - Fuentes de los ficheros -s