

at

Formato: at hora [fecha] [+ incremento]

comando(s)

[<CTRL-D>]

at -l [trabajo(s)]

at -r trabajo(s)

Descripción: El comando at toma una lista de comandos introducidos por teclado y los ejecuta en el instante (fecha y hora) que especifica el comando. No es preciso estar conectado. Se puede redireccionar la entrada con un fichero de comandos. La salida estándar es enviada por correo a no ser que se redireccione hacia un fichero.

Opciones:

-l Lista los números de trabajo para tareas planificadas por at.

-r Retira los números de trabajo indicados desde la cola de trabajos planificados por at.

cal

Formato: cal [[mes] año]

Descripción: Presenta en pantalla el calendario del año elegido. Si, además, se da el número de mes aparecerá ese mes. Si no se le indica ningún mes ni año, aparecerá el actual.

cat

filtro

Formato: cat [opción(es)] [fichero(s)]

Descripción: El comando cat lee la entrada de una lista de ficheros y/o de la entrada estándar, y copia todas esas entradas, por orden, en la salida estándar.

Muestra el contenido de ficheros.¹

Opciones

-v Muestra Los caracteres no imprimibles. Los caracteres de control aparecen usando un ^ inicial, por ejemplo; ^H representa un <CTRL-H>. El carácter DEL se representa como ^?. No aparecen tabuladores ni caracteres de nueva línea ni avance de página. Los caracteres no ASCII, se representan con la forma M-c, donde c es el carácter especificado por los siete bits.

cd

Formato: cd [directorio]

Descripción: Para cambiar de directorio actual, tecleando solo cd nos lleva a nuestro propio directorio, cd usa la variable HOME para ello.

chgrp

Formato: chgrp grupo fichero(s)

Descripción: El comando chgrp cambia el ID (identificación) de grupo de los ficheros seleccionados por el ID de un grupo específico, el grupo puede ser identificado por un ID decimal de grupo o por un nombre de grupo.

chmod

Formato:

chmod modo-absoluto fichero(s)

chmod modo-simbólico fichero(s)

Descripción: para cambiar los permisos sobre los ficheros.

Únicamente el superusuario y el propietario de un fichero pueden cambiar sus permisos.

Los permisos se distribuyen de la siguiente forma

– rwx rwx rwx

tipo de archivo derechos de usuario derechos de grupo derechos de otros

para cambiar los derechos de un archivo se puede hacer de dos formas

–con cifras

r=4 w=2 x=1 => 7

dándose un número por tipo de derecho (usuario, grupo, otros)

–extenso

r = read -> lectura

w = write -> escritura

x = exec -> ejecutable

" = todos los permisos

u = user -> usuario

g = group -> grupo

o = other -> otros

" = todas las clasificaciones

+ -> se da el permiso

" -> se da el permiso

- -> se quita el permiso ()

= -> el permiso permanece igual

chown

Formato: chown propietario fichero(s)

Descripción: El comando chown cambia la posesión del fichero indicado a la del propietario que se especifique. Para poder utilizar chown debe poseer un fichero, a no ser que sea el superusuario.

Opciones:

-R cambia todos los archivos o directorios con sus subdirectorios de manera recursiva

cmp

Formato: cmp [-1] [-s] fichero1 fichero2

Descripción: El comando cmp compara dos ficheros. Por defecto, informa sobre la localización de la primera diferencia, si es que existe, y guarda silencio si los dos ficheros son idénticos.

Opciones:

-l Imprime el número de bytes (en forma decimal) y los valores de los distintos bytes

(en octal) de cada diferencia.

-s No imprime nada en caso de ser ficheros diferentes (normalmente utilizados en procedimientos shell que miran el valor de salida del comando).

comm

Formato: comm [-123] fichero1 fichero2

Descripción: El comando comm examina dos ficheros, que se deberían clasificar de acuerdo con la secuencia ASCII. Así comm mostrará tres columnas, la primera contiene las líneas que se encuentran únicamente en el primer fichero, la segunda las del segundo fichero y la tercera las que están en ambos ficheros.

Las opciones le permiten suprimir la impresión de determinadas columnas. Puede usar un guión como nombre de fichero para indicar la entrada estándar.

Opciones:

-n Omite la columna n (n=1 a 3).

Las combinaciones como -12 omiten más de una columna.

CONTROL DE PROCESOS

<ctrl.>-z suspender trabajo actual

bg reanudar en modo subordinado el trabajo que estaba suspendido

fg reanudar trabajos suspendidos en primer plano

jobs lista los trabajos en ejecución (mas adelante)

kill determina la ejecución de un trabajo (mas adelante)

stop detiene temporalmente la ejecución de un trabajo

Formato: nombre de orden %n° de proceso|%nombre de orden (excepto ^z y ps)

cp

Formato:

cp fichero nuevofichero

cp fichero(s) directorio

Descripción: El comando cp puede copiar un único fichero en uno nuevo o copiar uno o más ficheros en un directorio dado. Los permisos del fichero también se copian.

cron

Formato: /etc/cron

Descripción: El comando cron es un comando demonio. El sistema ejecuta el comando y almacena los resultados en ficheros crontab. Los usuarios pueden crear sus propios ficheros crontab utilizando el comando crontab

cut

filtro

Formato:

cut -fn [-d char] [fichero]

En este caso n identifica el número o número de columnas y char el delimitador.

La entrada estándar se utiliza cuando se omite la lista de ficheros. La salida estándar puede ser un fichero.

date

Formato:

date [+ formato]

date mmddhhmm[yy] (Sólo superusuario).

Descripción: Cuando el comando date se usa sin argumentos, visualiza en la pantalla la fecha y la hora actuales. El superusuario puede dar un argumento de fecha y hora (mmddhhmm\yy\) para reinicializar el reloj del sistema y cualquier otro usuario puede proporcionar al sistema un argumento formato para especificar la manera como se presentará en pantalla la fecha.

%d numero del día (1–31)

%m número del mes (1–12)

%y últimos dos dígitos del año (00–99)

%D fecha en formato mm/dd/yy.

%H hora (00–23)

%M minutos (00–59)

%S segundos (00–59)

%T hora en formato HH:MM:SS.

%r hora en notación AM/PM

%a nombre del día (domingo–lunes)

%h nombre del mes (enero–diciembre)

%w día de la semana (0–6; el domingo es 0)

%n inserta un carácter de nueva línea

%t inserta un carácter tabulador

%j día del año (001–366)

echo

Formato: echo [argumento(s)]

Descripción: El comando echo muestra sus argumentos por la salida estándar (por defecto, el terminal).

Cada argumento, cuando aparece en pantalla, está separado del siguiente por un espacio, y la salida completa se termina con una nueva línea. El comando echo reconoce algunas notaciones especiales.

/c Imprime argumentos hasta ese punto, eliminando la nueva línea e ignorando los argumentos restantes.

/n Carácter de nueva línea (ir al comienzo de la nueva línea).

/b black space : borra un carácter

/f para a la siguiente pagina

/r return

/t tabulador

/v tabulación vertical

// imprime la /

/n° octal imprime un símbolo

egrep

Formato: egrep [opcion(es)] [patron(es)] [fichero(s)]

Descripción: El comando egrep saca normalmente aquellas líneas de un fichero que coinciden con el patrón o patrones. Si se busca en más de un fichero, el nombre de fichero se mostrará en pantalla con la línea coincidente. Un patrón puede contener caracteres especiales. Los patrones podían especificarle en la línea de comandos o en un fichero aparte.

Si no se da el nombre del fichero que hay que buscar, el comando egrep leerá de la entrada estándar.

La salida del comando egrep se envía a la salida estándar.

Opciones: -v Muestra en pantalla todas la líneas coincidentes.

-c Muestra en pantalla sólo un contador de líneas coincidentes; si hay varios ficheros se tratarán individualmente.

-l Muestra en pantalla únicamente los nombres de los ficheros que tienen líneas coincidentes

-n Muestra en pantalla el número de línea antes de cada una

-i No hace distinción entre mayúsculas y minúsculas al realizar comparaciones

-f Toma una lista de patrones de un fichero.

+ Permite la repetición de 1 o mas veces el carácter precedente ()

? Identifica la repetición de 0 o mas veces el carácter precedente

| Identifica el buscar cualquiera de 2 o mas elementos

() Tratar un texto como si fuera un grupo

EXPRESIONES REGULARES

Descripción: Son un conjunto de caracteres con un significado especial para la shell

. representa cualquier carácter

* representa la repetición de 0 o mas veces el carácter precedente

[] representa cualquiera de los caracteres representados entre los corchetes

[-] representa cualquiera de los caracteres representados entre el intervalo

[^] representa la negación de los caracteres representados dentro de los corchetes

^ representa comienzo de línea

\$ representa final de línea

exit

Formato:

Exit

<ctrl>D

Descripción: nos permite abandonar el sistema, desconectándonos del sistema o de la sesión iniciada

fgrep

Formato: fgrep [opción(es)] [cadena(s)] [fichero(s)]

Descripción: Normalmente el comando fgrep muestra en pantalla aquellas líneas que en un fichero (o ficheros) coincidan con una cadena dada o un conjunto de cadenas. Si se busca en más de un fichero, también se visualizará el nombre del fichero. Se puede buscar mas de una cadena; la lista de las cadenas buscadas se almacenara en un fichero. Si no se dan nombres de ficheros en la línea de comandos, se utilizará la entrada estándar. Así podrá emplearse un redircccionamiento. A diferencia de grep, este comando no reconoce los caracteres especiales de las expresiones regulares.

Opciones: -f Toma la cadena de fichero y no de la línea de comando.

find

Formato: find lista_de_vías_de_acceso expresión_de_búsqueda.

Descripción: El comando find busca a través de los directorios indicados y de todos sus subdirectorios, los ficheros que coinciden con un criterio de búsqueda. Este criterio puede ser un nombre de fichero concreto, un patrón general para nombres de ficheros, una propiedad del fichero (como tamaño o fecha de la última

modificación), o una combinación. El comando especificará que se muestren en pantalla los nombres de los ficheros coincidentes o que algún otro comando.

Opciones: -o para negar condiciones.

-name fichero Verdadero si fichero coincide con el nombre del fichero actual. Se puede utilizar la notación de meta caracteres, por ejemplo, -name '*.c'.

-perm onum Verdadero si el indicador de permiso de fichero coincide con el número en octal onum, que describe los permisos en el formato visto en chmod Si onum lleva el prefijo de signo menos, hará que los bits adicionales sean comparados, coincidiendo con el formato utilizado por la llamada de sistema stat().

-type c Verdadero si el fichero es de tipo c. donde c es un fichero b (bloque especial), c carácter especial), d (directorio), p (pipe ofifo) o f (fichero normal).

-user nombre Verdadero si el fichero pertenece al usuario nombre. Si nombre es un numero y ese numero no es un nombre de conexión de un usuario, entonces nombre se tomará como un número ID de usuario del fichero /etc/passwd

-group nombre Verdadero si el fichero pertenece al grupo nombre. Si nombre es un número y ese número no es un nombre de grupo, entonces nombre se tomara como un número de ID de grupo del fichero /etc/group.

-size n[c] Verdadero si el fichero tiene una longitud de n bloques, donde cada bloque son 512 bytes. Si n va seguida de una c, se tomará como el número de caracteres (un carácter es un byte).

-atime n Verdadero si se ha accedido al fichero hace n días. Nota: Los directorios en la lista de vías de acceso tienen su fecha cambiada por find

-mtime n Verdadero si el fichero fue modificado hace n días.

-print Verdadero si está presente; hace que se visualice por pantalla el nombre de vía de acceso actual. -a && ; not

-exec cmd Verdadero si el comando ejecutado (cmd) devuelve un estado de salida cero (ejecución con éxito). Se pueden utilizar llaves ({ }) para representar el nombre de vía de acceso actual. El final de cmd debe marcarse con un punto y coma con secuencia de escape (\,). Por ejemplo, -exec ls -l { } \; significa "listado de la información de estado de cada fichero que se encuentre".

-ok cmd Como -exec, excepto que la línea de comando deseada se visualiza primero con una interrogación. El comando se ejecutará únicamente si el usuario responde tecleando y.

finger

Formato: finger nombre_de_usuario

Descripción: nos da información acerca de un usuario aunque este no esté conectado al sistema

grep

filtro

Formato: grep [opción(es)] patrón [fichero(s)]

Descripción: El comando grep normalmente muestra en pantalla aquellas líneas de un fichero que coinciden con un patrón de búsqueda dado. El patrón puede ser una palabra, o más frecuentemente una cadena (secuencia de caracteres). El patrón puede incluir caracteres especiales de especificación de patrones. Si se revisa más de un fichero, grep mostrara en pantalla el nombre del fichero junto con las líneas coincidentes.

El comando grep procesa aquellos ficheros que están listados en la línea de comando grep

Si no-se dan argumentos de nombres de ficheros, grep leerá la entrada estándar, esto permitirá que este acepte entradas de un pipe (redirección).

Opciones:

- v Visualiza sólo aquellas líneas que no coinciden con el patrón.
- c Visualiza sólo el número de líneas coincidentes; los ficheros múltiples se tratan de manera individual.
- l Visualiza sólo los nombres de ficheros con líneas coincidentes, no las líneas.
- n Visualiza el número de línea antes de cada línea coincidente.
- i Evita que grep distinga entre mayúsculas y minúsculas

head

Formato: head -n fichero

Descripción: nos muestra las n primeras líneas de un documento

jobs

Definición: muestra los procesos

join

Formato: join [opción(es)] fichero1 fichero2

Descripción: El comando join toma los ficheros como argumentos. Normalmente, el primer campo de cada fichero se considera un "campo de unión". El comando join muestra en pantalla el campo de unión seguido del resto de la línea del primer fichero y a continuación del resto de la línea del segundo fichero. Por ejemplo, si el primer fichero contiene la línea

john pickiess chips

y el segundo la línea

john dip pretzeis

el campo de unión entonces será john y la línea de salida será

john pickiess chips dip préízeis

La opción -j indica el campo por el que ordenar.

Los campos se delimitan o terminan con separadores de campos; los separadores por defecto son los caracteres de espacio, tabulador y nueva línea. En la salida, se utilizará un blanco cuando se visualicen los campos.

kill

Formato: kill [-señal] PID(s)

Descripción: El comando kill envía una señal específica al "proceso" indicado. La señal se indica con un entero y el proceso con un número PID. Podemos encontrar el PID si usamos el comando ps. La señal de terminar (SIGTERM, en forma simbólica) se manda cuando no se especifica ninguna señal. Sólo los superusuarios pueden enviar señales a procesos que no sean los suyos.

El valor de PID 0 tiene un significado especial: significa "señalar todos los procesos en el grupo de procesos".
()

Opciones:

9 -SIGKILL Aborta (no puede ser ignorada).

15 SIGTERM Señal de terminar del software.

ln

Formato: ln [-f] nombre1 nombre2

ln [-f] nombre(s) de directorio

Descripción: UNIX identifica los ficheros internamente con un número de "i-nodo". El usuario, sin embargo, identifica los ficheros por el nombre. Cuando se crea un fichero, este se asocia con un nombre, o se une con su valor correspondiente de i-nodo. El comando lo permite unir nombres adicionales al mismo fichero. El comando puede usarse para asignar un nombre nuevo a un fichero dado o para unir el fichero al mismo nombre en un directorio distinto. Normalmente, si el nuevo nombre elegido existe ya y pertenece a un fichero protegido contra escritura, el modo permiso se mostrará en pantalla y preguntará si se desea continuar.

lock

Descripción: Bloquea el terminal.

logname

Formato: logname

Descripción: El comando logname da el nombre de conexión del usuario.

ls

Formato: ls [opcion(es)] [fichero(s)] [directorio(s)]

Descripción: El comando ls por si mismo lista los ficheros del directorio actual. Las opciones aportan mas información sobre los ficheros. Se puede listar un sólo fichero u otro directorio, dando su nombre como argumento.

Opciones:

–F Si un fichero contiene un programa ejecutable, se colocará un asterisco (*) después del nombre del fichero. Si el fichero es un directorio, se pondrá una barra (/) después del nombre.

–R Lista recursivamente un directorio; esto es, lista todos los ficheros y subdirectorios en cualquier subdirectorio que se encuentre.

–a Lista todas las entradas; por otra parte, los nombres que empiecen con un punto (.), como .profile, no se listarán.

–d Si un argumento es un directorio, listará únicamente su nombre y no su contenido. Esta opción a menudo se utiliza con –l para ver los modos de permiso del directorio.

–i Antepone a cada nombre de fichero su número de i-nodo; el sistema interno de numeración se utiliza para identificar ficheros.

–l Lisia en el formato largo, dando el modo, número de enlaces, propietario, grupo, tamaño en bytes y hora de la última modificación de cada fichero. Si el fichero es especial, el campo tamaño coincidirá con el número mayor y menor de dispositivo.

–r Invierte el orden de clasificación

–s Da el tamaño del fichero en unidades de 512 bytes.

–t Ordena de acuerdo con la fecha y hora de la última modificación

–u Ordena utilizando la hora del último acceso.

–x Muestra en pantalla la salida de multicolumna con entradas ordenadas a lo largo de la página en lugar de hacia abajo.

–q Visualiza una ? en lugar de un carácter no imprimible

–b Visualiza los caracteres no imprimibles con su valor en octal

–l Visualiza un fichero por línea

mail

Formato:

mail [-t] nombre(s)

mail [-epqr] [-f fichero]

Descripción: El comando mail es el comando básico de UNIX para el correo electrónico. Si le llama sin argumentos, los mensajes se mostrarán en pantalla de modo que el último en entrar será el primero en salir. Para enviar correo se llamará a mail con el nombre de conexión del usuario.

Opciones:

-t Para enviar el mismo mensaje a una lista de usuarios.

+ siguiente mensaje

- mensaje anterior

numero de mensaje actual

a mensajes recibidos durante el correo

d borrar mensaje actual

m reenviar mensaje al usuario especificado

p imprime el mensaje actual

q o <ctrl.>-d salir de mail

u [n] no borrar mensaje especificado

r responder al remitente y borrar el mensaje

s [archivo] guardar el mensaje en el archivo especificado o en \$HOME/mbox

w [archivo] guardar el mensaje en el archivo especificado o en ./mbox

x deja el correo como está y sale

? Muestra la ayuda

;orden ejecuta la orden especificada y vuelve a mail

man

Formato: man orden

Descripción: Nos proporciona ayuda acerca de las distintas ordenes de UNIX y funciones C, proporcionándonos sintaxis variables y función

more

filtro

Formato: orden |more

Descripción: Permite mostrar una salida estándar de una función o documento en pantalla de forma paginada

Opciones

<return> muestra la pagina siguiente

<space bar> muestra la siguiente línea

d avanza media pagina

b retrocede media pagina

q sale de more

mkdir

Formato: mkdir [-mp] [nombre del directorio(s)]

Descripción: El comando mkdir toma uno o mas nombres como argumentos y crea directorios con esos nombres. Usted tiene que tener permiso de escritura en un directorio para poder crear un nuevo subdirectorio en el. Hay dos entradas estándar que se hacen con cada nuevo directorio: . para el propio directorio y .. para el directorio padre.

mv

Formato: mv [-f] nombre1 nombre2

mv [-f] nombre(s) directorio

Descripción: El comando mv cambia el nombre de un fichero. Al cambiar el nombre de vía de acceso se cambiará también el directorio en el que está el fichero. El comando mv se usa de dos maneras. En la primera, nombre1 puede ser el nombre de un directorio o un fichero normal y nombre2, su nuevo nombre. En la segunda, nombre(s) representa los nombres de ficheros normales y directorio representa el nombre del directorio en el que se colocaran esos ficheros. Si se dan más de dos argumentos, el argumento final debe ser el nombre de un directorio ya existente. Si el argumento final es un nombre de directorio, mv asumirá que debe utilizarse de la segunda forma.

Opciones: -f Normalmente, si el argumento final es un fichero cuyo modo prohíbe la escritura, mv imprimirá el modo del fichero y preguntará si el usuario desea actuar. Con la opción -f, mv no hará preguntas y comenzará a actuar, si ello es posible.

passwd

Formato: passwd

Descripción: El comando passwd le permite crear o cambiar su palabra de paso de forma interactiva.

paste

filtro

Formato:

paste [-d lista] ficheros(s)

paste -s [-dlista] ficheros(s)

Descripción: El comando paste tiene dos modos. El primero (fusión paralela) toma una lista de ficheros de entrada como argumentos, fusiona las líneas correspondientes de cada fichero y envía el resultado a la salida estándar (por defecto, la pantalla). Las líneas de entrada se fusionan con el carácter tabulador (por defecto)

para formar la línea de salida.

El segundo modo (fusión serie) se invoca cuando es empleada la opción `-s`. En éste, cada línea de entrada se añade al final de la línea anterior, creando una línea más larga. Nuevamente, el carácter se usa por defecto para unir líneas.

El guión (`-`) puede emplearse como un nombre de fichero que represente la entrada estándar.

Opciones:

`-d` lista Define los caracteres usados para unir líneas. Si lista incluye más de un carácter, el primero se utilizará para la primera unión, el segundo para la segunda, y así sucesivamente. Si la lista se acaba antes de alcanzar la última unión de la línea, se usará se de nuevo empezando en su primer carácter. Los caracteres con significado especial shell, como `*` o un espacio, deben ponerse entre comillas.

`-s` Se usa para la función serie en vez de la paralela. Se colocan primero todas las líneas del primer fichero de entrada y después las del siguiente.

pr

Formato: `pr [opción(es)] [fichero(s)]`

Descripción: El comando `pr` envía el contenido del fichero nombrado a la salida estándar (por defecto la pantalla). Si no hay opciones, el texto se organizará en páginas de 66 líneas, cada página comienza con una cabecera de 5 líneas y termina con otras cinco líneas de pie de página.

Estas últimas constan de líneas en blanco, la cabecera consta de dos líneas en blanco, una línea con el número de página, la fecha y hora y el nombre de fichero. Después vienen dos líneas más en blanco, las líneas que sean demasiado largas se dividirán en dos, la entrada estándar se utilizará cuando no se especifique fichero o cuando se usa un guión como nombre de fichero.

Algunas opciones producen columnas separadas. Por defecto, estas columnas tendrán el mismo ancho e irán separadas al menos por un espacio.

Opciones:

`-d` Doble espacio en la salida.

`-lk` Especifica la longitud de página en `k` líneas. El valor por defecto es 66. Si no quedan suficientes líneas para incluir la cabecera y pie de página, estos serán suprimidos.

`-p` Cuando la salida está dirigida a un terminal, se detiene antes del comienzo de cada página. El comando `pr` hará que el terminal genere un pitido y esperará hasta que usted teclee `<INTRO>` antes de continuar.

`-h` cabecera Sustituye el nombre de fichero de la cabecera por la cadena cabecera.

`-t` Suprime las cinco líneas de cabecera y las cinco de pie de página. Termina la presentación después de la última línea de un fichero en vez de rellenar el final de la página con líneas vacías.

`-wk` Especifica el ancho de línea a `k` posiciones de carácter para salida multicolumna. El valor por defecto es 72.

–nck Provoca una salida no numerada. Si el entero k está presente, se emplearán k dígitos para cada número. El valor por defecto es 5. Si el carácter no numérico c está presente, se utilizará para separar los números del texto siguiente. El valor por defecto es el carácter fabulador. Las primeras k+1 posiciones de carácter de cada columna de salida se usarán para el número, y para la salida –m la línea se numerará por completo.

ps

Formato: ps [opción(es)]

Descripción: Dado que UNIX es un sistema operativo multiusuario y multitarea, lo normal es que atienda varias tareas de forma concurrente, cada una de las cuales se denomina un "proceso activo"

Opciones:

–u listauid Limita el listado a los procesos cuyos números de identificación de usuario ID o nombres de conexión vengan dados en la lista listauid. El número UID se visualizara en el listado salvo que esté activada la opción –f, en cuyo caso, aparecerá el nombre de conexión.

–f Genera un listado completo.

–e Presenta información de todos los procesos, no únicamente de los asociados con su terminal.

–l Genera un listado largo.

pwd

Formato: pwd

Descripción: Este comando proporciona el nombre del directorio de trabajo actual.

read

Formato: read [variable]

Descripción: lee un valor de la entrada estándar y lo almacena en una variable.

rm

Formato: rm [–fri] fichero(s)

Descripción: El comando rm, sin opciones, borra el fichero o ficheros dados del directorio que los contiene. Con más precisión, borra un nombre de fichero especificado de la lista del directorio. Un fichero puede tener varios nombres o "enlaces", y rm borrará solamente los enlaces que se especifiquen. Cuando el enlace final de un fichero se borra, también se elimina el propio fichero.

Opciones:

–f Borra el fichero sin formular preguntas, si se posee permiso de cuenta.

–r Acepta un nombre de directorio como argumento. Se borran todos los contenidos del directorio y después el directorio mismo. Esta opción trabaja de forma recurrente, es decir, se vacían y borran todos los subdirectorios, sub-subdirectorios y así sucesivamente.

–i Borra ficheros (o directorios si está –r) de forma interactiva, es decir, solicitando confirmación para cada fichero o directorio antes de borrarlo.

rmdir

Formato: rmdir [-ps] nombre(s) de directorio

Descripción: El comando rmdir borra los directorios especificados. Deben estar vacíos.

REDIRECTORES

Descripción: Son caracteres que se encargan de redireccionar la entrada o salida estándar a otros dispositivos

Tipos:

> redirecciona la salida a un fichero

>> redirecciona la salida como añadidura al final de un fichero

< redirecciona la entrada como un fichero

- redirecciona la salida estándar de errores

| redirecciona la salida de una orden como entrada de otra

sort

filtro

Formato: sort [opción(es)] [fichero(s)]

Descripción: El comando sort toma como entrada una lista de ficheros de texto ordenándolos por líneas, y envía el resultado a la salida escogida, por defecto, la pantalla Si se suministra más de un fichero de entrada, los ficheros se fusionarán durante el proceso de clasificación.

Opciones:

–c Comprueba si el archivo de entrada está clasificado. El orden creado en la clasificación dependerá de otras opciones, si las hubiera. No habrá salida de este comando a menos que el fichero no esté en el orden probado, en cuyo caso sort le informará de que dicho fichero está sin clasificar.

–m Fusiona los ficheros clasificados, suponiendo que los ficheros de entrada ya estén ordenados.

()

–u Suprime todas las líneas con idéntica clave de clasificación, menos una. Como la clave de clasificación es, por defecto, toda la línea, esta orden suprimirá varias líneas, a no ser que se especifique otra clave de clasificación.

–o salida Envía la salida al fichero salida. Este nombre puede ser uno de los ficheros de entrada. El espacio entre el –o y el nombre del fichero es opcional

-d Utiliza el orden de "diccionario". En las comparaciones únicamente son significativos las letras, dígitos y blancos (espacios y tabuladores), la puntuación, por ejemplo, se ignora.

-f Equipara letras mayúsculas y minúsculas.

-n Ordena numéricamente; esto es por su valor numérico y no por su carácter. El orden por defecto sitúa, por ejemplo, 1342 antes que el 3. porque el primer carácter de 1342. 1, está antes que 3 en la secuencia de código. Para que la clasificación numérica se produzca, la cadena numérica debe estar al principio de la línea o, más generalmente, al principio del campo de clasificación. Una cadena numérica consta de espacios blancos opcionales al principio, que son ignorados, un signo menos opcional, y cero o más dígitos, que pueden incluir un punto decimal. La opción -n no clasifica de manera correcta los números exponenciales como 2.314E10.

-r Invierte el orden.

-b Ignora los espacios en blanco iniciales cuando determinan el principio y el final de una clave de clasificación restringida. Si se coloca la opción -b antes del primer argumento +pos1 hará que todos los argumentos +post1 se vean afectados.

También puede atribuirse a irnos argumentos +pos1 y -pos2 individuales.

-tc Emplea el carácter c como separador de campo.

su

Formato: su [-] (nombre de usuario) [arg(s)]

Descripción: El comando su (cambio de usuario) le permite convertirse en otro usuario (cambiar su UID) sin desconectarse. Si no proporciona un argumento nombre de usuario, se asume root. A menos que sea el superusuario (por ejemplo, si da el nombre de conexión root). Se le pedirá La palabra de paso del usuario que usted desee ser. El comando su cambia su ID de usuario real y efectiva (tal como es dada por el comando id) a la del usuario indicado y se comienza un nuevo shell. Use <control-D> para reasumir su anterior identidad.

tail

Formato: tail [+/-[número | lbc] [f] [fichero]

Descripción: Por defecto, el comando tail presenta en pantalla las 10 últimas líneas de un fichero. Si no se da el nombre de fichero, tail leerá la entrada estándar. Se puede elegir para el comando tail el punto de comienzo de la visualización de un fichero, alterándolo con el uso del especificador de localización.

Opciones: -n Empieza la visualización n líneas antes del final

+n Empieza la visualización n líneas desde el principio.

b Cuenta en unidades de bloques (512 bytes).

c Cuenta en unidades de caracteres.

f Cuando se añade al final de la cadena de opciones hace que tail entre en un bucle infinito.

talk

Formato: talk usuario[@host.dominio]

Descripción: permite establecer una comunicación entre dos usuarios, haciendo que lo que uno escriba en su pantalla aparezca a la vez en la pantalla del otro y viceversa

tee

filtro

Formato: tee [-i] [-a] [fichero(s)]

Descripción: El comando tee transmite la entrada estándar a la salida estándar, a la vez que realiza una copia de la entrada en un fichero. Por defecto, se sobrescribe en los ficheros ya existentes.

Si quiere obtener más de una copia, se puede listar más de un fichero de destino.

Opciones: -a Añade la salida a un fichero en lugar de escribirla sobre dicho fichero.

test

Formato: test expresión

[expresión]

La forma entre corchetes es una alternativa. Téngase en cuenta el uso de blancos para delimitar los corchetes.

Descripción: Las estructuras de comandos usadas en los procedimientos del shell, como el bucle while y la estructura de control if emplean el estado de salida de un comando test para decidir qué hacer seguidamente. Un estado de salida cero se considera como verdadero, mientras que un estado de salida no cero se considera falso.

Opciones de test

- Comparaciones para números enteros

-eq A -eq B si son iguales

-ne A -ne B si son distintos

-gt A -gt B si $A > B$

-ge A -ge B si $A \geq B$

-lt A -lt B si $A < B$

-le A -le B si $A \leq B$

- Comparaciones para cadenas de caracteres

-z -z cadena si la longitud de la cadena =0

-n -n cadena si la longitud de la cadena !=0

= cadena1=cadena2 si la cadena 1 es igual a la cadena 2

!= cadena1!=cadena2 si la cadena 1 no es igual a la cadena 2

cadena cadena si la cadena !=NULL

- Comparaciones para archivo

-a archivo si existe el archivo

-r archivo si existe el archivo y se puede leer

-w archivo si existe el archivo y se puede escribir

-x archivo si existe el archivo y se puede ejecutar

-f archivo si existe el archivo y es un archivo regular

-d archivo si existe el archivo y es un directorio

-h archivo si existe el archivo y es un enlace simbólico

-c archivo si existe el archivo y es un archivo de caracteres especiales

-b archivo si existe el archivo y es un archivo de bloques

-p archivo si existe el archivo y es un cauce con nombres

-s archivo si existe el archivo y tiene tamaño > 0

touch

Formato: touch [-amc] [mmddhhmm|aa] fichero(s)

Descripción: UNIX mantiene información en un i-nodo para cada fichero. El i-nodo incluye el momento de creación, el del último acceso y el de la última modificación. El comando touch le permite actualizar las horas de acceso y modificación. Por defecto, el comando touch actualiza ambos, utilizando la información temporal especificada, si se proporciona, o el tiempo actual del sistema en cualquier otro caso. También por defecto, si un fichero que ha sido nombrado no existe, se crea.

Opciones:

-a Actualiza solamente las horas de acceso.

-m Actualiza solamente las horas de modificación.

-c No crea un fichero si el fichero nombrado no existe.

tr

filtro

Formato: tr [-cds] [cadena1 [cadena2]]

Descripción: El comando tr copia los caracteres desde la entrada estándar a la salida estándar sustituyendo o borrando algunos de ellos en el camino. Este comando no puede leer y escribir ficheros, por lo que normalmente se emplearán redirecciones o tuberías para conectar la entrada y salida estándar a otros ficheros o programas. Aquí cadena1 consta de los caracteres que van a sustituirse y cadena2 de los caracteres de sustitución.

Opciones:

-c Usa todos los caracteres que no estén en la cadena1, es decir, se toman los caracteres cuyos códigos están comprendidos entre los números octales 001 y 377 y se hace la lista de sustituciones de aquellos que no se den explícitamente en la cadena1.

-d Borra los caracteres de cadena1 del texto de entrada antes de enviarlos a la salida

-s Cuando un carácter de cadena1 aparece dos o más veces consecutivas en la segunda, comprime la secuencia de caracteres repetidos, hasta dejarla con un solo carácter.

umask

Formato: umask [000]

Descripción: El conjunto actual de permisos sobre un fichero es el que proporciona el proceso de creación del fichero y el que modifica la máscara de creación de ficheros. La máscara indica los permisos que no están permitidos. Por defecto, la máscara suele ser 0022. El comando umask permite poner la máscara de creación al valor que usted elija. Cuando se utiliza sin argumentos, umask toma el valor actual de la máscara.

VARIABLES DE ENTORNO

Son aquellas variables que tiene el sistema para guardar ciertos valores de variables que pueden ser de utilidad para distintas ordenes o el SHELL

CDPATH contiene la lista de directorios buscados por la orden cd, la fija el usuario y en ella se separan las direcciones con :

HOME contiene la trayectoria del directorio de trabajo del usuario y su valor lo pone inicialmente el administrador

LOGNAME contiene el nombre de usuario y se fija automáticamente al inicio de la sesión

MAIL contiene la trayectoria del directorio donde se encuentra el correo

MAILFILE contiene la trayectoria del archivo con los mensajes de mailx ()

PATH contiene la lista de directorios donde la shell buscara los archivos

PS1 indicativo primario del shell

PS2 indicativo secundario del shell

SHELL contiene la trayectoria del shell del usuario y es fijada automáticamente

TERM define el tipo de terminal utilizado, lo define el usuario y es utilizado principalmente por las ordenes de texto o las que están destinadas a trabajar con la pantalla

TZ información sobre la zona horaria utilizada, la fija el sistema

Para referirnos a una variable tenemos que preceder el nombre de la variable del signo \$

Para establecer una variable o cambiar su valor escribiremos

NOMBRE_DE_LA_VARIABLE=valor de la variable

Si la variable contiene espacios en blanco la enceraremos entre comillas.

Para definir una variable como de entorno utilizaremos la orden export

export nombre_de_variable

wait

Formato: wait [PID]

Descripción: El programa shell se encarga del interfáz entre usted y el sistema. Normalmente, cuando se usa & para ejecutar una tarea en modo subordinado, el shell reanuda el proceso de los comandos subsiguientes. El comando wait, cuando se da sin argumentos, produce la reanudación del shell únicamente después de que todos los procesos subordinados hayan terminado. Si se da un PID con el comando wait y ese PID identifica a una tarea subordinada, entonces el shell esperará hasta que aquel proceso concreto termine. Si el PID no identifica a un proceso subordinado, entonces el shell esperara a que todos los procesos estén terminados.

wc

filtro

Formato: wc [-lwc] [nombre (s) de fichero(s)]

Descripción: El comando wc sirve para contar el numero de caracteres, palabras y líneas en los ficheros que se hayan nombrado. Cuando se indica el nombre de más de un fichero, wc proporciona tanto los contadores individuales como el total; se presentan los de cada fichero etiquetados con su nombre. Aquí las palabras se definen en términos de "espacio en blanco". Un espacio en blanco es un espacio, uno fabulador o una nueva línea. Una palabra es una cadena de caracteres que no contiene espacios en blanco y que está limitada por ellos.

Opciones: ()

-l Cuenta líneas

-w Cuenta palabras

-c Cuenta caracteres.

who

Formato:

who [opción(es)] [nombre de fichero]

who am i

who i am

Descripción: El comando who lista, por defecto, los nombres de conexión, las líneas de terminales y los tiempos de conexión de los usuarios que actualmente estén conectados al sistema. Hay varias opciones que provocan la visualización de información. Normalmente, el comando who obtiene la información del fichero /etc/utmp, pero la toma de nombre de fichero si se da el nombre del mismo como un argumento. Al teclear who am i o who i am, el comando identificará al usuario que el sistema cree que es usted.

Opciones:

-T Añade el campo estado a la lista de campos. Si fuera la única opción, el resto de los campos serian los indicados en la opción -i.

-s Lista únicamente los campos nombre, línea y hora.

-q Es la opción rápida de who; sólo visualiza los nombres y el numero de usuarios. ignorando las otras opciones.

write

Formato: write usuario [terminal]

Descripción: El comando write hace que las líneas que se hayan escrito desde un terminal se presenten con eco en el terminal destinatario Asimismo, el destinatario puede utilizar el comando write para comunicarse con el terminal emisor. Para finalizar el diálogo, se usa una señal de interrupción (normalmente generada por <break> o) o la señal EOF (en general, un <control-d> al comienzo de una línea).

&

Formato: orden &

Descripción: ordena a la shell que ejecute la orden en modo subordinado

*** ? [] CARACTERES ESPECIALES**

Descripción: son aquellos caracteres que tienen un significado especial para la shell

*

?

[]

<

>>

>

\

/ anula el significado especial del siguiente carácter

anula el significado especial de los caracteres entrecomillados excepto \$ y /

` ' anula el significado especial de todos los caracteres entrecomillados

` (ejecución de la orden en la shell, devolviendo el resultado a la propia shell

() ejecución de las ordenes en un subshell, cuya salida se toma como unidad respecto a la redirección

{ } ejecución de las ordenes en la shell, cuya salida se toma como unidad, respecto a la redirección

| redirecciona la salida de un comando a la entrada de otro

METACARACTERES

* sustituye cualquier o ningún carácter excepto el punto inicial

? sustituye cualquier carácter excepto el punto inicial

[] cualquiera de los caracteres indicados

[-] cualquiera de los caracteres del rango indicado

4