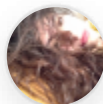


WUOLAH



emewinchester

www.wuolah.com/student/emewinchester



600

Practica 2 APSO resuelta.pdf

PRÁCTICAS RESUELTAS (incluye teoría)



2º Administración y Programación de Sistemas Operativos



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingeniería
UHU - Universidad de Huelva

5

**CONSIGUE UN 5
PELAO EN INGLÉS
Y GANA UN VIAJE
A NUEVA YORK**

NO TE LIES
VIAJA

**MY
ENGLISH
COOL**
EASIER,
CHEAPER...
BETTER

PARTICIPA Y GANA

APSO: PRÁCTICA 2

MANEJO DE LA SHELL (II)

Marina Delgado Pérez
GRADO INGENIERÍA INFORMÁTICA | HUELVA

ENTRA EN LA HISTORIA DE MIXTA

CO-CREA LA NUEVA LATA LIMITED EDITION.
TU DISEÑO SALDRÁ EN TODOS LADOS Y GANARÁS 5.000 EUROS.



PARTICIPA EN
WWW.SABEAMIXTA.COM
#MixtaLimitedEdition

Contenido

1.- COMANDOS.....	2
grep	2
wc	2
cal	2
date	2
sort	2
ps	2
tty	2
write	2
mesg	2
id.....	2
kill	3
tail o head.....	3
df	3
top	3
2.- COMANDO FIND.....	3
3.- ATRIBUTOS DE FICHEROS Y PERMISOS	3
4.- CREACIÓN DE ENLACES	5
HARD LINK	5
SOFT LINK	5

PRELUDIO TEÓRICO

1.- COMANDOS

Los comandos que veremos en esta práctica son los siguientes:

grep

Busca, dentro de un fichero, un patrón. Muestra líneas que concuerdan con un patrón. Se busca por conjunto de caracteres.

wc

Contabiliza. Siempre muestra por pantalla un número. Puede contar líneas, caracteres o palabras de un fichero. Si no nos dicen lo contrario, cada caracter es un byte de información (así calculamos lo que ocupa un fichero).

cal

Calendario oficial de UNIX. Se consulta cualquier fecha, pasada, presente o futura. Consultar man para ver más opciones.

date

Imprime por pantalla la fecha y hora actual del sistema.

sort

Ordena líneas de archivos de texto. Hace ordenaciones de todo tipo en función de las opciones que se especifiquen.

ps

Muestra los procesos activos del sistema y su estado, del terminal en el que estamos. Para poder ver todos los procesos activos que tiene un usuario, independientemente del terminal que esté usando, habrá que especificarlo mediante opciones.

tty

Imprime por pantalla el nombre del terminal que se tiene abierto. (/dev/pts/numero).

write

Permite escribir un mensaje.

mesg

Activa/desactiva los mensajes. Opciones: (y/n).

id

Imprime por pantalla qué usuario soy y a qué grupo pertenezco. Pide al sistema las características del usuario.



kill

Permite enviar un mensaje a un proceso. Si el proceso no está preparado, muere. Nos permite matar el proceso que nosotros queramos. Para poder matar un proceso, hace falta ser propietario de ese proceso. En caso contrario, no podremos enviar el mensaje.

tail o head

Muestra líneas desde arriba de un fichero (head) o desde abajo (tail).

df

Muestra la información sobre el disco, la utilización del espacio en sistema de ficheros.

top

Muestra ocupación de cpu de los procesos del sistema, de forma continuada y a tiempo real. Es un comando informativo.

2.- COMANDO FIND

Comando que sirve para buscar en todo el sistema lo que nosotros queramos, desde una ruta especificada. Muy importante especificar siempre la ruta, incluido si es desde el directorio en el que nos encontremos (en ese caso, la ruta sería punto (.)). Las búsquedas se hacen por patrones. Al ser tan extenso el comando, mirar `man find` para más información.

SINTAXIS:

`find especificar_ruta_busqueda opciones_busqueda`

3.- ATRIBUTOS DE FICHEROS Y PERMISOS

Para consultar los atributos de ficheros y sus permisos tan solo hay que ejecutar el comando `ls -l`.

Los permisos funcionan en UNIX de la siguiente manera:

PERMISOS
Lectura (R)
Escritura (W)
Ejecución (X)

Los permisos se aplican sobre 3 grupos de usuarios:

PROPIETARIO	GRUPO DEL PROPIETARIO	OTROS
-------------	-----------------------	-------

EJEMPLO: `-rw-r--r-- f1.txt`

`drwxr-xr-x` **prac1** (la d del principio indica que es un directorio).

Los permisos pueden darse (o quitarse) a cualquier fichero.

El permiso de lectura sobre un directorio permite ver lo que hay dentro de ese directorio.

Si un fichero no tiene permiso de escritura (W), no se puede borrar ni modificar. Lo mismo ocurre para los directorios.

Los ficheros tienen permiso de ejecución (X) si pueden ser ejecutados, es decir, si contienen código ejecutable en su interior.

El permiso de ejecución (X) en un directorio permite hacer búsquedas en él. En el caso de estar desactivado, no podríamos ejecutar el comando `find`.

La única persona que puede dar o quitar permisos sobre unos directorios o ficheros es el propietario de los respectivos directorios y ficheros.

El comando `chmod` se utiliza para dar o quitar permisos sobre ficheros y archivos. Su sintaxis es la siguiente:

chmod	u (propietario)	dar -> +	R W X
	g (grupo)		
	o (otros)	quitar -> -	
	a (todos)		

En una misma línea de comando se pueden o dar o quitar permisos, pero no dar y quitar a la vez.

EJEMPLO:

`chmod uo-x f1.txt` (quita permiso X al propietario y a otros).

Otra manera de dar o quitar permisos es la forma global (solo se utilizará cuando se nos indique en las prácticas).

EJEMPLO: Si quisiéramos que `f1.txt` tuviera los permisos que se muestran a continuación, ejecutaríamos la siguiente línea de comandos.

`f1.txt` `RWX R-- R-X`  `chmod 745 f1.txt`
 `111 100 101`

4.- CREACIÓN DE ENLACES

En UNIX, cada archivo en el sistema está representado por un inodo. Un inodo no es más que un bloque que almacena información de los archivos, de esta manera, a cada inodo podemos asociarle un nombre. En principio, un mismo archivo no podría tener más de un nombre, pero debido a los enlaces es posible asignar más nombres a archivos.

HARD LINK

El enlace físico no es más que un nombre para un fichero, una etiqueta o nuevo nombre asociada a un fichero existente. En otras palabras, una forma de identificar el mismo contenido con diferentes nombres. Es mucho más rápido que el soft link, pero menos seguro.

Este enlace **NO** es una copia separada del archivo anterior, sino un nombre diferente para exactamente el mismo contenido.

El contenido del inodo no se eliminará mientras haya un enlace físico que le haga referencia. Esto puede tener ventajas, pero también puede complicar la tarea de seguimiento de los archivos. Un enlace físico tampoco puede usarse para hacer referencia a directorio o a archivos en otros equipos.

Para crear un enlace físico en UNIX del archivo `archivo.txt` a `nuevo_nombre.txt`, ejecutamos:

```
ln archivo.txt nuevo_nombre.txt
```

El enlace aparecerá como otro archivo más en el directorio y apuntará al mismo contenido de `archivo.txt`. Cualquier cambio que se haga se reflejará de la misma manera tanto para `archivo.txt` como para `nuevo_nombre.txt`.

SOFT LINK

Un enlace simbólico también puede definirse como una etiqueta o un nuevo nombre asociado a un archivo, pero a diferencia de los enlaces físicos, el enlace simbólico no contiene los datos del archivo, simplemente apunta al registro del sistema de archivos donde se encuentran los datos (se asemeja a un acceso directo en Windows).

Sintaxis para crear enlace simbólico en UNIX:

```
ln -s archivo.txt nuevo_nombre.txt
```

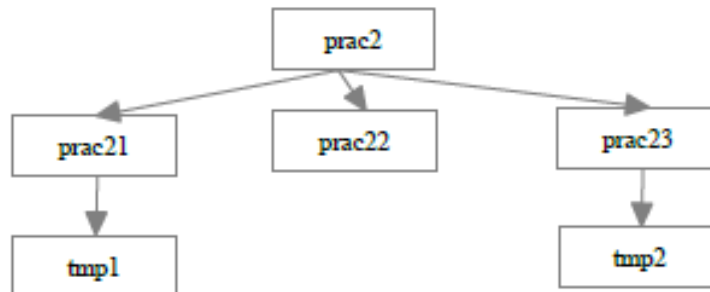
Este enlace también aparecerá como otro archivo más en el directorio y apuntará al mismo contenido de `archivo.txt`, reflejando todos los cambios que se hagan tanto para `archivo.txt` como para `nuevo_nombre.txt`.

Sobre un enlace simbólico también se pueden usar todos los comandos básicos de archivos (`rm`, `mv`, `cp`, ...), sin embargo, cuando el archivo original es borrado o movido a una ubicación diferente, el enlace dejará de funcionar (**enlace roto**).

Un enlace simbólico permite enlazar directorios y, usando NFS, también permite enlazar archivos fuera del equipo.

RESOLUCIÓN PRÁCTICA 2

1. Cree con un solo comando el siguiente árbol de directorios a partir de su directorio personal con rutas relativas.



```
mkdir prac2 prac2/prac21 prac2/prac21/temp1 prac2/prac22  
prac2/prac23 prac2/prac23/temp2
```

2. Muévase al directorio `prac2`. Copie los ficheros `f1.txt` y `f2.txt` que se encuentran en `/home/so/velez/MI` en el directorio `prac2` usando rutas absolutas para indicar la ruta a los ficheros y ruta relativa para indicar el directorio de destino.

```
cd prac2  
cp /home/so/velez/MI/f[12].txt .
```

3. Use la adición y sustracción de permisos para dejar al directorio `prac2` sin permiso de lectura y escritura, tanto para el grupo del propietario como para los que no son del grupo del propietario. El propietario debe quedarse con los permisos que ya tenía.

```
ls -l .. (para ver qué permisos tiene el directorio prac2)  
chmod go-r /home/marina.delgado536/prac2
```

4. Usando el método de colocación de permisos global, haga que los ficheros `f1.txt` y `f2.txt` tengan todos los permisos para el propietario, permiso de lectura para los usuarios del grupo del propietario y ningún permiso para los usuarios que no son del grupo del propietario.

```
chmod 740 f[12].txt
```

5. Muévase al directorio `prac21` mediante rutas absolutas. Mediante rutas relativas haga una copia de los ficheros `f1.txt` y `f2.txt` en el directorio `tmp1`. Renombre el fichero `f1.txt` del directorio `tmp1` a `f21.txt` con rutas relativas.

```
cd /home/marina.delgado536/prac2/prac21  
cp ../f[12].txt tmp1
```



*¿Estás de exámenes?
Estos podríais ser
tú y tu colega...*

MENTES BRILLANTES

MENTES
BRILLANTES

una película de
**THOMAS
LILTI**

VINCENT
LACOSTE

WILLIAM
LEBGHIL

★★★★★
"IRRESISTIBLE,
HILARANTE Y
CONMOVEDORA"
ROLLING STONE

5 DE ABRIL
EN CINES

```
mv temp1/f1.txt temp1/f21.txt
```

6. Mueva y renombre el fichero `f2.txt` del directorio `tmp1` al directorio `tmp2` con el nombre `f23.txt` usando rutas relativas y en un solo comando.

```
mv temp1/f2.txt ../prac23/temp2/f23.txt
```

7. Muévase con un solo comando al directorio `tmp2` sin usar rutas absolutas. Busque con el comando `find` los directorios que cuelgan de su directorio personal cuyo segundo carácter sea una `r` o una `t` (tanto en mayúsculas como en minúsculas), su penúltimo carácter no sea un número y su último carácter sea un número.

```
cd ../prac23/temp2
find /home/marina.delgado536/ -name ?[rRtT]*[!0-9][0-9] -
type d
```

8. Busque los ficheros que no sean directorios (pueden ser de cualquier otro tipo) y que cuelguen de su directorio personal, que sean más nuevos que el fichero `solp1.txt` que se encuentra en su directorio `prac1` y que hayan sido modificados hace menos de 3 días. Use rutas absolutas en todo momento.

```
find /home/marina.delgado536/ ! -type d -newer
/home/marina.delgado536/prac1/solp1.txt
```

9. Consulte con el `grep` donde aparecen en los ficheros `f1.txt` y `f2.txt` del directorio `prac2` los caracteres `llama`, sea en mayúscula o en minúscula (con un solo comando). Deben aparecer 5 veces en `f1.txt` y 6 veces en `f2.txt` para estar correcto.

```
grep -c -i -e llama /home/marina.delgado536/prac2/f[12].txt
```

10. ¿Cuántos bytes ocupa el fichero `f21.txt`? Indique dos comandos distintos con los que pueda averiguar el tamaño en bytes de un fichero.

```
wc -c ../../prac21/temp1/f21.txt
(resultado= 1499 bytes)
```

```
grep -b ""
/home/marina.delgado536/prac2/prac21/temp1/f21.txt
(resultado = 1479 + 20 bytes que cuento a mano)
```

11. Consulte en que día de la semana se fundó la Universidad de Huelva (1 de Julio de 1993) con el comando `cal`. (Nota: consulte sólo el mes y averigüe el día de la semana que es).

```
cal 7 1993
(día: martes)
```

12. Muestre la hora y fecha con el siguiente formato:

Bienvenido. Son las <hora> horas y <minuto> minutos del <día de la semana en letras> <día del mes> de <mes en letras>. Ha sido un placer.

```
date "+Bienvenido. Son las %H horas y %M minutos del %A %d de %B. Ha sido un placer."
```

13. Muévase a su directorio personal. Ordene el fichero usuarios04-05.txt que se encuentra en el directorio /home/so/velez/MI de menor a mayor y guarde el fichero resultante en el directorio prac22 con el nombre usu.orden.pabajo. Haga lo mismo, pero ordenando de mayor a menor y guarde el resultado en el directorio prac22 con el nombre usu.orden.parriba.

```
cd
sort -o prac2/prac22/usu.orden.pabajo /home/so/velez/MI/usuarios04-05.txt
sort -r -o prac2/prac22/usu.orden.parriba /home/so/velez/MI/usuarios04-05.txt
```

14. Visualice el fichero f1.txt. Visualice las 2 ultimas líneas del fichero solf1.txt que se encuentra en el directorio /home/so/velez/MI para ver la solución al enigma.

```
more prac2/f1.txt
tail -n 2 /home/so/velez/MI/solf1.txt
```

(la clave es el numero de caracteres que tiene el numero que dice el portero)

15. Muévase al directorio prac2. Visualice el número de caracteres del fichero f23.txt del directorio tmp2.

```
cd prac2
wc -m prac23/temp2/f23.txt
```

16. Cree un enlace (hard link) llamado solucionprac1 en el directorio prac2 que se corresponda con el fichero solp1.txt del directorio prac1.

```
ln ../prac1/solp1.txt solucionprac1
```

17. Cree un enlace simbólico (soft link) llamado `temporal` en el directorio `prac2` que apunte al directorio `tmp2` del directorio `prac23`.

```
ln -s prac23/tmp2 temporal
```

18. Usando el comando `file` averigüe de que tipo son los ficheros `solucionprac1` y `temporal`.

```
file temporal
(resultado: sympolic link to 'prac23/tmp2')
```

```
file solucionprac1
(UTF-8 Unicode text, with CRLF, LF line terminators)
```

19. Usando el `who` visualice sólo los nombres de todas las sesiones que hay actualmente abiertas en el sistema, y el número de sesiones abiertas. Se necesita una opción.

```
who -q
```

20. Abra una nueva sesión. Edite con el `joe` un fichero con el nombre `inservible`. Averigüe con el `ps` el identificador del proceso `joe`. Mate al proceso `joe` con el comando `kill`.

```
(nuevo terminal)
joe inservible
```

```
(en el terminal de la practica)
ps -u
kill 47230 (47230 es el PID del proceso joe)
ps -u (comprobamos que el proceso joe ha sido abortado)
```

21. Escriba un mensaje con el `write` al usuario `velez` que tiene la sesión abierta a una hora más temprana.

(no se hizo en clase)

22. Deje todos los permisos menos el de ejecución al fichero `solp2.txt` y quite el resto de los permisos para todos los que no son de su grupo sobre ese fichero.

```
chmod go-r solp2.txt
```