**EJERCICIOS SISTEMAS**

**CP IN**

0) Teclear los siguientes comandos que generarán los ficheros y los

directorios que se usaran a lo largo de esta práctica:

touch Fichero-1.txt

ln -s Fichero-1.txt Fichero-3.txt

touch Fichero-5.txt

chmod u-w Fichero-5.txt

mkdir Directorio-uno

touch Directorio-uno/Fichero-14.txt

touch Directorio-uno/Fichero-36.txt

mkdir Directorio-tres

1) Crear una copia de Fichero-3.txt con el nombre Fichero-4.txt

El Fichero-3.txt era en realidad un "enlace simbólico" a Fichero-1.txt

pero, el nuevo Fichero-4.txt que es, ¿ un enlace o un fichero real ?

cp Fichero-3.txt Fichero-4.txt

Es un fichero real.

2) Si en el punto anterior se hubiese deseado crear una copia del "enlace

simbólico", ¿ como se habría tenido que invocar al comando cp ?

cp -s Fichero-3.txt Fichero-6.txt

cp -d Fichero-3.txt Fichero-6.txt no funciona.

cp -P Fichero-3.txt Fichero-6.txt no funciona.

**CD IN**

1) Teclear los siguientes comandos que generarán el directorio y el

fichero que se usaran para los siguientes apartados:

alumno@pc03:~$mkdir Directorio-uno

alumno@pc03:~$touch Directorio-uno/Fichero-1.txt

alumno@pc03:~$ln -s Directorio-uno Directorio-dos/

2) Cambiar al Directorio-uno y listar su contenido. Cambiar al

Directorio-dos y listar su contenido. ¿ Qué ficheros contienen cada uno de

ellos ? ¿ por qué ?

alumno@pc03:~$cd Directorio-uno

alumno@pc03:~/Directorio-uno$ls

--> Fichero-1.txt

alumno@pc03:~/Directorio-uno$cd Directorio-dos

alumno@pc03:~/Directorio-dos$ls

--> Fichero-1.txt

Porque anteriormente se hizo una copia del Directorio-uno con el

nombre de Directorio-dos.

3) Cambiate nuevamente a tu carpeta personal y acontinuación cambiate al Directorio-dos usando la opción -P del comando cd ¿ en que directorio nos sitúa según el sistema operativo ?

alumno@pc03:~/Directorio-dos$cd

alumno@pc03:~$cd -P Directorio-dos

-P al cambiar de directorio ignora los enlaces simbólicos

alumno@pc03:~/Directorio-uno$

**IN**

1) Crear en el directorio personal un fichero llamado Original.txt con el

contenido dado más abajo y "enlazarlo" de forma "blanda" (o simbólica) con los

nombres Fichero-1.txt, Fichero-5.txt, Fichero-14.txt, Fichero-36.txt,

Fichero-72.txt, Fichero-103.txt

Según el comando ls, ¿ cuantos enlaces tiene el fichero Original.txt ? ¿ Por

qué ? (Sugerencia: el número que aparece en un listado realizado con ls -l

justo después de los permisos representa el número de enlaces)

ln -s original.txt Fichero-1.txt. Y así sucesivamente con todos los

demás.

ls -l original.txt. Según esto tiene 1 enlace. Porque un enlace

blando crea ficheros que solo contienen el nombre y la ruta de otro

fichero. Pero la máquina entiende que solo existe un fichero con 6

nombres distintos.

2) Eliminar el fichero Original.txt creado en el apartado anterior. ¿ Es

posible eliminarlo o previamente hay que eliminar todos los enlaces "blandos"

-simbólicos- que se le crearon ? Si se intenta hacer cat Fichero-1.txt, ¿ que

mensaje da el sistema operativo ? Crear nuevamente el fichero Original.txt

con el contenido dado más abajo. Si ahora se intenta hacer cat Fichero-1.txt,

¿ es necesario actualizar los enlaces "blandos" anteriores o siguen

funcionando como si nada hubiese ocurrido ? Si el contenido para el fichero

Original.txt hubiese sido otro, ¿ que hubiese ocurrido con los enlaces

"blandos" ? ¿ Por qué ?

rm original.txt

No hace falta borrar los enlaces blandos.

cat: Fichero-1.txt: No existe el fichero o el directorio

Sale el contenido del fichero:

Este fichero será el fichero

original que se usará para

realizar los ejercicios relativos

a enlaces con el comando ln

Sigue funcionando como si nada hubiera ocurrido.

Al poner cat hubiese salido el contenido del fichero original.txt, a

los ficheros blandos no le afecta el contenido del fichero principal.

3) Eliminar todos los enlaces "blandos" usados para los

apartados anteriores y crear dos enlaces "duros" y uno "blando" para el fichero

Original.txt El primer enlace "duro" se hará desde el propio directorio

personal y se llamará Original-A.txt

El segundo enlace "duro" se hará desde el directorio /tmp y se llamará

Original-B.txt

El enlace "blando" se hará desde el directorio personal y se llamará

Enlace.txt

Según el comando ls ¿ cuantos enlaces tiene el fichero Original.txt ? ¿ Y el

fichero Original-A.txt ? ¿ Y el fichero Original-B.txt ? ¿ Y el fichero

Enlace.txt ?

¿ Que permisos de acceso tienen los ficheros Original.txt, Original-A.txt y

Original-B.txt ? ¿ Y el fichero Enlace.txt ? ¿ Siempre tienen los ficheros de

enlaces "blandos" los mismos permisos ?

rm Fichero-1.txt , con todos los demás ficheros se hace lo mismo.

ln original.txt Original-A.txt

cd /tmp

ln /home/usuario/original.txt /home/usuario/Original-B.txt

cd

ln -s original.txt Enlace.txt

ls -l original.txt

-rw-r--r-- 3 alumno users 119 nov 13 16:41 original.txt

original.txt tiene tres enlaces.

ls -l Original-A.txt

-rw-r--r-- 3 alumno users 119 nov 13 16:41 Original-A.txt

Original-A.txt tiene tres enlaces.

ls -l Original-B.txt

-rw-r--r-- 3 alumno users 119 nov 13 16:41 Original-B.txt

Original-B.txt tiene tres enlaces.

ls -l Enlace.txt

lrwxrwxrwx 1 alumno users 12 nov 13 17:01 Enlace.txt ->

original.txt

Sí, siempre tienen el mismo permiso.

4) Desde el punto de vista de UNIX los ficheros se identifican por números (lo

que se llama inodo o nodo índice) y en realidad no le importa que un fichero

tenga el nombre que sea o que "cuelgue" de tal o cual directorio. ¿ Como se

usaría el comando ls para ver el número de "inodo" que le corresponde al

fichero Original.txt del apartado anterior ? ¿ Y para ver el número de

"inodo" que le corresponden a los ficheros Original-A.txt y Original-B.txt ?

¿ Tienen el mismo número de "inodo" ? ¿ Por qué ?

Y el fichero Enlace.txt ¿tiene el mismo número de "inodo" ? ¿ Por qué ?

alumno@pc03:~/prueba > ls -i original.txt

226951 original.txt

alumno@pc03:~/prueba > ls -i Original-A.txt

226951 Original-A.txt

alumno@pc03:~/prueba > ls -i Original-B.txt

226951 Original-B.txt

Original-A.txt y Original-b.txt si tienen los mismos inodos porque

son el mismo contenido del fichero con distintos nombres.

alumno@pc03:~/prueba > ls -i Enlace.txt

226969 Enlace.txt

No porque se trata del mismo contenido, pero en distinta ruta.

5) Usar el comando df -i y anotar la cantidad exacta de "inodos" que

existen (libres y ocupados) en el sistema de ficheros.

Eliminar el fichero Original-B.txt ¿ Que variación experimenta el

número de "inodos" ? Eliminar el fichero Enlace.txt ¿ Que variación

experimenta el número de "inodos" ?

df -i

Filesystem Inodes IUsed IFree IUse% Mounted on

/dev/hda6 1101056 79598 1021458 7% /

/dev/hda1 4016 21 3995 1% /boot

rm Original-B.txt

df -i

Filesystem Inodes IUsed IFree IUse% Mounted on

/dev/hda6 1101056 79598 1021458 7% /

/dev/hda1 4016 21 3995 1% /boot

El número de inodos no cambia ya que existen otros dos ficheros con

el mismo contenido.

rm Enlace.txt

alumno@pc03:~/prueba > df -i

Filesystem Inodes IUsed IFree IUse% Mounted on

/dev/hda6 1101056 79597 1021459 7% /

/dev/hda1 4016 21 3995 1% /boot

6) Crear dentro del directorio personal otro directorio creando nuevamente el fichero llamado Original.txt con el

contenido dado más abajo y "enlazarlo" de forma "dura" con los nombres enlaceduro-1.txt,

enlaceduro-2.txt y enlaceduro-3.txt.

Según el comando ls, ¿ cuantos enlaces tiene el fichero Original.txt ? ¿ Por

qué ? (Sugerencia: el número que aparece en un listado realizado con ls -l

justo después de los permisos representa el número de enlaces)

ln Original.txt enlaceduro-1.txt. Y así sucesivamente con todos los

demás.

ls -l Original.txt. Según esto tiene 4 enlaces. Tenemos un único fichero

con 4 nombres distintos.

7) Queremos eliminar ahora el fichero Original.txt. ¿Qué tendríamos que hacer?

¿Cómo eliminaríamos el fichero completamente del sistema de archivos?

(Sugerencia: usar el comando find para localizar todos los nombres del mismo fichero)

Primero tenemos que obtener el inodo del fichero Original.txt: ls -i Original.txt

1844224 Original.txt

Segundo, buscamos todos los ficheros desde el raíz que tengan ese inodo:

sudo find ~ -inum 1844224

Otra forma mucho más rápida y sencilla sería utilizar find de la siguiente forma:

sudo find ~ -samefile ./Original.txt -exec rm {} \;

De esta forma no tenemos que buscar el inodo del fichero, por lo que nos

ahorramos un paso.

**TODAS PROBADAS Y HE TENIDO PROBLEMAS CON ALGUNOS YA QUE ME RESULTAN DIFÍCILES PERO LAS PUDE SOLUCIONAR AL FINAL**