

REPASO DE LINUX. CAPÍTULO 3: FICHEROS Y DIRECTORIOS II

REPASO DE LINUX. CAPÍTULO 3: FICHEROS Y DIRECTORIOS II

Caracteres comodín

Comandos para copia y borrado de ficheros

Comando `cp`

Comando `mv`

Comando `rm`

Comandos para copia y borrado de directorios

Comando `cp`

Comando `mv`

Comando `rm`

Ejercicios

Caracteres comodín

Se pueden crear patrones utilizando **símbolos comodín** para no tener que escribir todos y cada uno de los ficheros.

- Para mostrar cada uno de los ficheros que comienzan por `docu` seguido de un número del 1 al 6 se puede utilizar un patrón:

```
$ cat fich[1-6]
```

- Para mostrar el contenido de todos los ficheros que comienzan por `fich` se puede hacer:

```
$ cat fich*
```

dónde `*` representa cualquier combinación de caracteres, incluso la cadena vacía.

- El carácter `*` se puede colocar en cualquier lugar. Por ejemplo, para mostrar todos los ficheros que comienzan por la letra "a" y terminan por la letra "s" dentro del directorio `/usr/bin`:

```
$ ls /usr/bin/a*s
```

- El símbolo `?` representa un carácter cualquiera. De esta forma, la siguiente sentencia muestra todos los ficheros del directorio `/usr/bin` cuyo nombre comienza por "g", sigue cualquier carácter, a continuación sigue una "o" y termina con cualquier cadena de caracteres incluida la cadena vacía.

```
$ ls /usr/bin/g?o*
```

- Los **corchetes** se utilizan de forma parecida al carácter `?` aunque, a diferencia de éste, permiten especificar un poco más. Por ejemplo, `[adfg]` significa cualquiera de los caracteres `a`, `d`, `f` o `g`. `[Hh]ola` es un patrón que encaja tanto con "Hola" como con "hola". `[a-z]*` representa cualquier cadena de caracteres que comienza con una letra minúscula.

Comandos para copia y borrado de ficheros

Comando `cp`

Sirve para copiar ficheros. Se puede copiar un único fichero o muchos. Se pueden copiar tanto ficheros como directorios. Por supuesto, se pueden utilizar símbolos comodín.

En el proceso de copia intervienen 3 factores: lo que se copia, la ruta de origen y la ruta de destino.

Por ejemplo, la siguiente sentencia copia el fichero `hosts`, que se encuentra en el directorio `/etc` al directorio `/home/alumno/pruebas/`.

```
$ cp /etc/hosts /home/alumno/pruebas/
```

Si no se especifica ningún directorio origen, se toma por defecto el directorio actual. El siguiente ejemplo copia todos los archivos con la extensión `.odt` del directorio actual al directorio `textos`.

```
$ cp *.odt textos/
```

Cuando se quiere especificar como directorio destino el directorio actual se utiliza el carácter `.`

El siguiente ejemplo copia todos los ficheros del directorio `/usr/bin` que comienzan por la letra "g" al directorio actual.

```
$ cp /usr/bin/g*
```

Comando `mv`

Sirve para dos cosas, para mover y para cambiar de nombre. Se puede hacer cualquiera de las dos cosas por separado o las dos cosas al mismo tiempo. Por ejemplo:

```
$ mv mi_texto.txt carta.txt
```

Le cambia el nombre a `mi_texto.txt` y pasa a llamarse `carta.txt`

En cambio:

```
$ mv carta.txt Documentos/
```

mueve `carta.txt` al directorio `Documentos`

Se pueden hacer dos cosas a la vez, mover y cambiar el nombre:

```
$ cd Documentos/  
/Documentos$ mkdir correspondencia  
/Documentos$ mv carta.txt correspondencia/carta01.txt
```

Comando `rm`

Se utiliza para borrar ficheros, aunque estos no se envían a la papelera así que **no se pueden recuperar una vez borrados**. Por ejemplo:

```
$ rm *.txt
```

Borra todos los archivos con la extensión `txt` del directorio actual.

```
alejandra@cliente24:~$ rm *.txt  
alejandra@cliente24:~$
```

Comandos para copia y borrado de directorios

De la misma forma que se copian, borran o mueven ficheros, se puede hacer lo mismo con directorios. Hay que tener en cuenta que los directorios pueden contener muchos ficheros y otros directorios que a su vez pueden contener otros ficheros y directorios.

Comando `cp`

Sirve para copiar directorios. Si se quiere copiar un fichero completo con todo lo que incluye en su interior hay que indicarlo con la opción `-R` ("copiar de forma recursiva"). Por ejemplo:

```
$ mkdir multimedia2
$ cp multimedia/* multimedia2
```

Se hace una copia del contenido del directorio `multimedia` al directorio `multimedia2`.

Comando `mv`

Funciona de forma análoga a `cp`, pero mueve en lugar de copiar. Cuando se trata de renombrar, funciona igual que con ficheros. Por ejemplo:

```
$ mv multimedia2 multimedia_copia
```

```
alejandra@cliente24:~$ mv multimedia2 multimedia_copia
alejandra@cliente24:~$
```

Esto cambia el nombre del directorio `multimedia2` a `multimedia_copia`. El contenido de ese directorio permanece intacto.

Comando `rm`

Se pueden borrar directorios. Por ejemplo:

```
$ rm multimedia_copia/
```

```
alejandra@cliente24:~$ rm multimedia_copia/
rm: no se puede borrar 'multimedia_copia/': Es un directorio
alejandra@cliente24:~$
```

Se obtiene un error porque hay que borrar el contenido de forma recursiva:

```
$ rm -rf multimedia_copia/
```

```
alejandra@cliente24:~$ rm -Rf multimedia_copia/  
alejandra@cliente24:~$
```

Además de la opción `-R` se incluye la opción `-f` que hace que nos pida confirmación por cada elemento que se quiera borrar.

Ejercicios

1. Muestra todos los archivos del directorio actual que son imágenes `jpg`.

```
$ ls *.jpg
```

2. Muestra todos los archivos del directorio `/usr/bin` que empiecen por la letra `j`.

```
$ ls /usr/bin/j*
```

```
alejandra@cliente24:~$ ls /usr/bin/j*  
/usr/bin/join          /usr/bin/jsongdiff      /usr/bin/json_pp  
/usr/bin/journalctl    /usr/bin/jsonpatch      /usr/bin/jsonschema  
/usr/bin/jpgicc        /usr/bin/json-patch-jsongdiff  
/usr/bin/jq            /usr/bin/jsonpointer
```

3. Muestra los archivos que empiecen por `k` y tengan una `a` en la tercera posición, dentro del directorio `/usr/bin`.

```
$ ls /usr/bin/k?a*
```

4. Muestra los archivos del directorio `/bin` que terminen en `n`.

```
$ ls /bin/*n
```

```
alejandra@cliente24:~$ ls /bin/*n
/bin/aptdcon          /bin/hp-align        /bin/prezip-bin
/bin/bluemoon         /bin/hp-clean        /bin/pw-mon
/bin/btmon            /bin/hp-plugin       /bin/py3clean
/bin/catman           /bin/hp-scan         /bin/rpcgen
/bin/chcon            /bin/ibus-daemon     /bin/runcon
/bin/chfn             /bin/join            /bin/sbsign
/bin/chown            /bin/kmodsign        /bin/sbvarsign
/bin/column           /bin/ldapmodrdn      /bin/session-migration
/bin/cpan             /bin/linux-version   /bin/setupcon
/bin/dbus-daemon      /bin/ln              /bin/slogin
/bin/dbus-run-session /bin/login           /bin/splain
/bin/dbus-uuidgen     /bin/man             /bin/ssh-keygen
/bin/dh_bash-completion /bin/mapscrn        /bin/ssh-keyscan
/bin/eqn              /bin/mimeopen        /bin/systemd-run
/bin/fc-pattern       /bin/neqn            /bin/tracker3-daemon
/bin/fc-scan          /bin/open            /bin/uuidgen
/bin/fzputtygen       /bin/pkaction        /bin/vcs-run
/bin/gapapplication  /bin/pkcon           /bin/xcursorgen
/bin/geqn             /bin/pkmon           /bin/xdg-desktop-icon
/bin/gnome-keyring-daemon /bin/pod2man        /bin/xdg-open
/bin/gnome-session    /bin/pon             /bin/xman
```

5. Muestra todos los archivos que hay en `/etc` y todos los que hay dentro de cada subdirectorio, de forma recursiva.

```
$ ls -R /etc
```

```
alejandra@cliente24:~$ ls -R /etc
/etc:
adduser.conf      host.conf         ppp
alsa              hostname          printcap
alternatives      hosts            profile
anacrontab        hosts.allow      profile.d
apg.conf          hosts.deny       protocols
apm              hp               prueba.txt
apparmor          ifplugd          pulse
apparmor.d        init             python3
appport           init.d           python3.12
apt              initramfs-tools  rc0.d
avahi             inputrc          rc1.d
bash.bashrc       insserv.conf.d   rc2.d
bash_completion  ipp-usb          rc3.d
bindresvport.blacklist  iproute2         rc4.d
binfmt.d          issue            rc5.d
bluetooth         issue.net        rc6.d
brlapi.key        kernel           rcS.d
brltty            kerneloops.conf  resolv.conf
brltty.conf       krb5.conf.d      rmt
ca-certificates   ldap             rpc
ca-certificates.conf  ld.so.cache     rsyslog.conf
```

6. Crea un directorio en tu directorio de trabajo con nombre `prueba`. Copia el archivo `gzip` del directorio `/bin` al directorio `prueba`. Crea un duplicado de `gzip` con nombre `gzip2` dentro de `prueba`.

```
$ mkdir prueba
$ cp /bin/gzip prueba/
$ cp prueba/gzip prueba/gzip2
```

```
alejandra@cliente24:~$ mkdir prueba
alejandra@cliente24:~$ cp /bin/gzip prueba/
alejandra@cliente24:~$ cp prueba/gzip prueba/gzip2
```

7. Cambia el nombre de `prueba` a `prueba2`. Crea `prueba3` en el mismo nivel que `prueba2` y mueve todos los ficheros de `prueba2` a `prueba3`. Borra `prueba2`.

```
$ mv prueba prueba2
$ mkdir prueba3
$ mv prueba2/* prueba3/
$ rmdir prueba2
```

```
alejandra@cliente24:~$ mv prueba prueba2
alejandra@cliente24:~$ mkdir prueba3
alejandra@cliente24:~$ mv prueba2/* prueba3/
alejandra@cliente24:~$ rmdir prueba2
```

8. Crea un fichero vacío con nombre `*?Hola caracola?*`. ¿Se puede? En caso de que se pudiera, ¿sería recomendable poner nombres así? Razona la respuesta.

```
$ touch '*?Hola caracola?*'
```

```
alejandra@cliente24:~$ touch '*?Hola caracola?*'

```

Sí se puede, si se pon el nombre entre comillas, aunque no es recomendable.

9. Crea un directorio con nombre `multimedia_pruebas` y copia en él todo el contenido del directorio `multimedia`. A continuación crea en `multimedia/video/` dos ficheros, uno con nombre `peliculas.txt` y otro con nombre `actores.txt`. Edita el fichero `peliculas.txt` e introduce el nombre de tu película favorita. A continuación, crea en `multimedia_pruebas/video/` otro fichero que también tenga por nombre `peliculas.txt`, edítalo y esta vez escribe el nombre de tus cinco películas favoritas. Ahora haz una copia de todo el contenido de `multimedia` en `multimedia_prueba` de tal forma que sólo se copien los contenidos nuevos, es decir, si hay coincidencia en el nombre de un archivo se respetará el que se haya modificado más recientemente. Para comprobar que se ha hecho todo correctamente, basta mirar si en `multimedia_prueba/video` está el archivo vacío `actores.txt` y además el archivo `peliculas.txt` debe contener 5 películas y no 1.

```
$ mkdir multimedia_pruebas
$ cp multimedia/* multimedia_pruebas/
$ cd multimedia/video/
$ mkdir peliculas.txt actores.txt
$ nano peliculas.txt
$ cd ..
$ cd multimedia_pruebas/video/
$ mkdir peliculas.txt
$ nano peliculas.txt
$ cd ..
$ cp multimedia/* multimedia_pruebas/
```

10. Borra el directorio `multimedia/imagenes/otras`. El sistema debe pedir al usuario que confirme el borrado.

```
$ rm -ri multimedia/imagenes/otras
```

```
alejandra@cliente24:~$ rm -ri multimedia/imagenes/otras
rm: ¿borrar el directorio 'multimedia/imagenes/otras'? (s/n) s
```

11. Mueve el archivo `peliculas.txt`, que está dentro de `multimedia/video`, al directorio que está justo a un nivel superior. Ahora el archivo debe llamarse `mis_peliculas.txt` en lugar de `peliculas.txt`.

```
$ mv multimedia/video/peliculas.txt multimedia/mis_peliculas.txt
```