Curso Práctico de Iniciación al uso del Entorno de Alta Computación

BU-ISCIII

Práctica 1: Comandos básicos de Linux

Índice

Descripción

En esta práctica se usarán los comandos más básicos para trabajar desde la terminal.

El objetivo es realizar, desde la línea de comandos, aquellas tareas cotidianas que normalmente se hacen desde un entorno gráfico, como crear, copiar, mover, visualizar o editar archivos, pero aplicadas al entorno de alta computación.

Notas importantes

- Usa el tabulador para guiarte en la terminal y autocompletar nombres de ruta, archivos y comandos (el tabulador es tu mejor aliado).
- Usa las flechas del teclado para moverte por el historial de comandos ejecutados (podrás reutilizarlos sin volver a escribirlos).
- No es aconsejable usar espacios, tildes ni caracteres especiales como la "ñ" en los nombres de archivos o directorios.
- Comandos básicos que siempre debes recordar: `pwd`, `cd`, `ls`, `mkdir`, `mv`, `rm`, `rmdir`, `less`, `nano`.

Cheats

Directorios	Archivos
mkdir	touch
mv	mv
cp -r	cp
rmdir	rm
cd	
	less
	cat
	nano

Ejercicios

> Nota: En este archivo hay un esquema con los cambios en las rutas que se van ralizando a lo largo de la práctica.

Abrimos una terminal en nuestro ordenador. En Linux se puede pulsar `Ctrl + Alt + T` para abrir una terminal.

▶ PREGUNTA: ¿Qué información nos muestra el prompt?

Entre corchetes `[ ]` se muestra primero tu usuario, después el `@`, seguido del nombre del ordenador al que estás conectado y, finalmente, la ruta en la que te encuentres en ese momento. Fuera de los corchetes aparecerán los permisos con los que estás conectado, normalmente `$` si no tienes permisos de administrador.

Para poder conectarte al Entorno de Alta Computación (HPC) del ISCIII tienes que ejecutar el siguiente comando en la terminal que se explicará más adelante:

```
ssh -p 32122 <usuario>@portutatis.isciii.es
```

Escribes la contraseña de tu usuario del HPC

> Nota: No se muestra la contraseña a medida que escribes ni se indica ningún tipo de caracter para indicarte que estás escribiendo, así que asegurate de escribirla bien.

▶ PREGUNTA: ¿Qué información nos muestra el prompt una vez conectados al HPC??

Entre corchetes `[]` te proporciona primero tu usuario, después el `@`, seguido el nombre de la máquina a la que estás conectado, en el HPC es el nodo de acceso que se llama portutatis03. Finalmente se muestra la ruta en la que te encuentres en ese momento, que cuando accedes al HPC es el home de tu usuario `/home/<nombre de usuario`, que se abrebia con el simobolo de `~`. Fuera de los corchetes te mostrará los permisos con los que estás conectado, que en el HPC es `$`, sin permisos de administrador.

1. Muestra tu directorio de trabajo actual

```
pwd
```

▶ PREGUNTA: ¿En qué directorio te encuentras?

`/home/<usuario>`. Siempre que te conectes al HPC el directorio al que entras por defecto es el directorio personal de tu usuario, que SIEMPRE se encuentra en la ruta `/home/<tu nombre de usuario>`.

▶ PREGUNTA: ¿Lo que nos muestra `pwd` es ruta absoluta o relativa?

El comando `pwd` muestra una ruta absoluta.

> Nota: El resultado del comando `pwd` debe coincidir con la ruta que se muestra en el prompt.

2. Listar los archivos de tu carpeta actual.

1. Lista el contenido del directorio de tu home (`/home/usuario`).
2. Lista el contenido del directorio de tu home (`/home/usuario`) en formato largo
3. Lista el contenido del directorio de tu home (`/home/usuario`) mostrando los archivos ocultos
4. Lista el contenido del directorio de tu home (`/home/usuario`) mostrando los archivos ocultos y en formato largo separando los parámetros.
5. Lista el contenido del directorio de tu home (`/home/usuario`) mostrando los archivos ocultos y en formato largo juntando los parámetros.

```
pwd
ls
ls -l
ls -a
ls -a -l
ls -la
```

ALGUNOS (no todos) parámetros se pueden juntar en un único parámetro, como es el caso de `-a` y `-l` del comando `ls` que se pueden juntar en `-la` para que sea más rápido a la hora de programar.

▶ PREGUNTA: ¿Qué ficheros ves?

▶ PREGUNTA: ¿Qué carpetas ves?
▶ PREGUNTA: ¿Qué hace el parámetro `-l`?

El parámetro `-l` nos muestra la lista de archivos en formato largo, lo que significa que nos muestra la información de permisos de usuario, la información de modificación, espaico de almacenamiento, etc.

▶ PREGUNTA: ¿Qué hace el `-a`?

El parámetro `-a` sirve para mostrar los archivos y directorios ocultos (aquellos que empiezan por `.`. Estos suelen ser archivos o directorios de configuración que necesitan algunos softwares para funcionar correctamente.)

3. Crear directorios

```
ls
mkdir practica_comandos
ls
```

▶ PREGUNTA: ¿Ves tu carpeta nueva que antes no estaba?

4. Moverte entre directorios

Antes de moverte a un directorio diferente, comprueba siempre donde estás y que archivos hay en tu carpeta con `pwd` y `ls`. Haz lo mismo siempre que cambies de carpeta.

```
pwd
ls
cd practica_comandos
pwd
ls
```

▶ PREGUNTA: ¿Que contiene la carpeta a la que te has movido?

5. Crear directorios, moverse y crear archivos vacíos dentro.

1. Desde `practica_comandos` crear 2 directorios, uno se llamara 'dir1' y el otro 'dir2'
2. Acceder dentro del directorio `dir1` y crear dos archivos de 'texto' vacios, uno llamado `archivo1.txt` y otro llamado `archivo2.txt`.
3. Volver al directorio de inicio (/home/alumno)

```
ls
mkdir dir1 dir2
ls
pwd
cd dir1
ls
```

```
> archivo1.txt
ls
touch archivo2.txt
ls
cd
pwd
ls
```

Vemos que tanto `>` como `touch` permiten crear archivos que antes no estaban.

▶ PREGUNTA: ¿Cuál sería el path absoluto al archivo `archivo1.txt` que acabas de crear?

`/home/<usuario>/practica_comandos/dir1/archivo1.txt`

▶ PREGUNTA: ¿Y el path relativo a la carpeta en la que te encuetras actualmente (`/home/usuario`)?

`./practica_comandos/dir1/archivo1.txt` o `practica_comandos/dir1/archivo1.txt`

▶ PREGUNTA: ¿Cómo crearías el archivo `archivo1.txt` desde tu home sin moverte de carpeta?

Con los comandos:

Ruta relativa: `touch practica_comandos/dir1/archivo1.txt` o `> practica_comandos/dir1/archivo1.txt`

Ruta absoluta: `touch /home/<usuario>/practica_comandos/dir1/archivo1.txt` o `> /home/<usuario>/practica_comandos/dir1/archivo1.txt`

▶ PREGUNTA: ¿Que hace el comando `cd` sin argumento?

Te mueve al directorio de tu home ubicado siempre en `/home/<tu nombre de usuario>`

6. Modificar y visualizar el contenido de los archivos

1. Moverse al directorio `practica_comandos/dir1/` Usando el editor nano, añadir texto al archivo `archivo1.txt`, guardar (pulsar ctrl + o) y salir del editor (ctrl + x).
2. Visualizar en pantalla el contenido de `archivo1.txt`.
3. Usando el editor nano, añadir texto DISTINTO al archivo `archivo2.txt`, guardar (pulsar ctrl + o) y salir del editor (ctrl + x).
4. visualizar en pantalla el contenido de `archivo1.txt` y `archivo2.txt` con una sola instrucción.
5. Por último, guardar el contenido de los 2 ficheros en uno nuevo y llámalo `juntar_ficheros.txt`.
6. Visualizar en pantalla.

```
pwd
ls
cd practica_comandos/dir1/
pwd
ls
nano archivo1.txt
# escribe algo
# (ctrl + o) para guardar cambios: Nos pregunta: "File Name to Write", hay
```

```
que confirmar con "Intro" el nombre del archivo a escribir.
# (ctrl + x) para salir
cat archivo1.txt
```

```
nano archivo2.txt
# escribe algo diferente
# (ctrl + o) para guardar cambios
# (ctrl + x) para salir
cat archivo2.txt
```

```
cat archivo1.txt archivo2.txt
cat archivo1.txt archivo2.txt > juntar_ficheros.txt
cat -n juntar_ficheros.txt
```

▶ PREGUNTA: Antes usamos > para algo diferente, ¿qué hace exactamente el >?

El simbolo > sirve para redireccionar el standard output de un comando a un archivo. En este caso el standard output del comando `cat archivo1.txt archivo2.txt` que es el contenido de los dos ficheros uno después de otro, es redireccionado al archivo `juntar_ficheros.txt`

▶ PREGUNTA: ¿Qué hace el parámetro `-n` en el comando `cat`?

El parámetro `-n` permite visualizar el número de linea de un archivo que estás leyendo con `cat`.

7. Parámetros de un mismo comando

1. Comprobar en qué directorio estás
2. Listar el contenido en formato largo del directorio `dir1`
3. Probar con otros parámetros del comando ls (-t, -S, -r).

```
pwd
ls -l
ls -t
ls -S
ls -r
```

▶ PREGUNTA: ¿Qué hace el parámetro `-t`?

El parámetro `-t` minúscula (distinto de `-T` mayúscula) lista los archivos en orden de tiempo, poniendo el más reciente el primero.

▶ PREGUNTA: ¿Qué hace el parámetro `-S`?

El parámetro `-S` mayúscula (distinto de `-s` minúscula) lista los archivos ordenandolos por tamaño, poniendo primero el más grande.

▶ PREGUNTA: ¿Qué hace el parámetro `-r`?

El parámetro `-r` minúscula (distinto de `-R` mayúscula) te hace un listado en orden inverso. Por defecto se lista por orden alfabético por lo que el parámetro `-r` por sí solo listará en orden alfabético inverso. Si se juntara con el parámetro `-t` haría un listado por orden de tiempo reverso.

8. Copiar archivos y directorios

1. Copiar el archivo `archivo1.txt` a otro archivo y lee el contenido de ambos.
2. Copiar el archivo `juntar_ficheros.txt` a otro directorio y lee el contenido de ambos.
3. Muevete al directorio `practica_comandos` y copia el directorio `dir2` a `dir2_copia` y lista el contenido de ambos.

```
pwd
ls
cp archivo1.txt archivo_copiado1.txt
ls
cat archivo1.txt
cat archivo_copiado1.txt
```

```
pwd
ls
cp juntar_ficheros.txt ../dir2/archivo_copiado2.txt
ls
ls ../dir2/
cat juntar_ficheros.txt
cat ../dir2/archivo_copiado2.txt
```

```
pwd
ls
cd ..
pwd
ls
cp dir2 dir2_copia
ls
cp -r dir2 dir2_copia
ls
ls dir2 dir2_copia
```

A los comandos que requieren archivos o directorios como argumentos, se les puede proporcionar tanto la ruta relativa como absoluta a esos ficheros, de forma que se pueden crear, leer, modificar, etc. archivos o directorios que no están en tu directorio actual.

▶ PREGUNTA: ¿Cuál es la ruta ABSOLUTA a los archivos `archivo_copiado1.txt` y `archivo_copiado2.txt`?

`archivo_copiado1.txt`: `/home/<nombre usuario>/practica_comandos/dir1/archivo_copiado1.txt` `archivo_copiado2.txt`: `/home/<nombre usuario>/practica_comandos/dir2/archivo_copiado2.txt`

▶ PREGUNTA: ¿Qué diferencia hay entre `cp` y `cp -r`?

`cp` por sí solo sin parámetros permite copiar archivos pero no directorios, si lo intentamos se queja con el siguiente mensaje `cp: -r not specified; omitting directory 'dir2'`. El parámetro `-r` le indica al comando `cp` que funcione de forma recursiva y permite copiar también directorios. Recursivo, dicho especialmente de un proceso, signfica que se aplica de nuevo al resultado de haberlo aplicado previamente.

9. Mover y renombrar archivos

1. Mueve el `archivo_copiado1.txt` de `dir1` a `dir2`
2. Renombra `dir2` a `directorio_copias`
3. Muevete a `dir2` y renombra `archivo_copiado2.txt` a `juntar_ficheros.txt`

```
pwd
ls
ls dir1 dir2
mv dir1/archivo_copiado1.txt dir2/archivo_copiado1.txt
ls dir1 dir2
```

```
pwd
ls
mv dir2 directorio_copias
ls
ls dir2 directorio_copias
```

```
pwd
ls
cd directorio_copias
pwd
ls
mv archivo_copiado2.txt juntar_ficheros.txt
ls
```

▶ PREGUNTA: ¿Qué hace el comando `mv`?

El comando `mv` permite mover y renombrar tanto directorios como archivos. Es lo que hacen las órdenes cortar y pegar.

10. Limpiar pantalla

```
pwd
ls
cat juntar_ficheros.txt
clear
```

▶ PREGUNTA: ¿Qué hace el comando `clear`?

El comando `clear` permite limpiar (aclarar) la pantalla de la terminal para poder ver todo más limpio.

11. WildCards

1. Lista todos los archivos que empiecen por la palabra archivo
2. Lista todos los archivos que terminen por la palabra .txt
3. Lee el contenido de todos los archivos que empizan por la palabra archivo
4. Lee el contenido de todos los archivos que terminen por la palabra .txt

```
pwd
ls
ls archivo*
ls *.txt
```

```
cat archivo*
cat *.txt
```

12. Leer archivos

1. Renombrar `archivo1.txt` a `archivo_importante.txt`
2. Leer el contenido de `archivo_importante.txt`
3. Leer y redireccionar el contenido del archivo `/etc/passwd` a `archivo_importante.txt`
4. Leer el contenido de `archivo_importante.txt` con distintos comandos

```
pwd
ls
mv archivo_copiado1.txt archivo_importante.txt
cat archivo_importante.txt
ls
cat /etc/passwd > archivo_importante.txt
ls
cat archivo_importante.txt
less archivo_importante.txt # Pulsar q para salir de less
more archivo_importante.txt # Pulsar q para salir de more
```

▶ PREGUNTA: ¿Por qué usar `less` o `more` teniendo `cat`?

Los comandos `less` y `more` nos permiten visualizar los archivos fuera del prompt. Si leyeramos un archivo muy grande con `cat`, se nos mostraría en el prompt y no podríamos visualizar el principio del archivo, porque la terminal tiene un límite de lineas que nos puede mostrar. Si usamos `less` o `more`, podemos ir visualizando el archivo poco a poco.

▶ PREGUNTA: ¿Que diferencia hay entre `less` y `more`?

Con el comando `less` al salir no nos quedará el texto leído en la pantalla, pero con el comando `more` una vez que salgamos al haber encontrado el texto de interés, nos aparecerá en el terminar para poder usarlo.

13. Eliminar archivos

1. Elimina el archivo `archivo1.txt`
2. Elimina el archivo `archivo2.txt`
3. Muevete al directorio `dir1`
4. Elimina todos los archivos del directorio

```
pwd
ls
rm archivo1.txt
rm -f archivo1.txt
rm archivo_importante.txt
ls
```

```
pwd
ls
cd ../dir1/
pwd
ls
rm *
ls
```

▶ PREGUNTA: ¿Qué ha pasado con el archivo `archivo1.txt`?

Como se puede ver al hacer el `ls` antes de lanzar el `rm`, el archivo `archivo1.txt` no existía porque lo habíamos renombrado. Al no existir el archivo que queremos eliminar, el comando `rm` se "queja" con este mensaje `rm: cannot remove 'archivo1.txt': No such file or directory` porque no ha podido eliminar un archivo que no existe.

▶ PREGUNTA: ¿Qué hace el parámetro `-f`?

El parámetro `-f` le dice al comando `rm` que ignore los archivos que no existen, por lo que aunque `archivo1.txt` no existe, el comando `rm` ya no se queja y no te avisa.

14. Eliminar directorios

1. Muevete a `practica_comandos`

2. Lista el contenido de `dir1`
3. Elimina el directorio `dir1`
4. Lista el contenido de `directorio_copias`
5. Elimina el directorio `directorio_copias`
6. Muevete al home
7. Elimina el directorio `practica_comandos`

```
pwd
ls
cd ..
pwd
ls
ls dir1
rm dir1
rmdir dir1
ls
```

```
ls directorio_copias
rm directorio_copias
ls
rmdir directorio_copias
ls
rm -r directorio_copias
ls
```

```
pwd
ls
cd ..
pwd
ls
rm -rf practica_comandos
```

▶ PREGUNTA: ¿Que pasa con `rm` si no le pasas argumentos y quieres borrar directorios?

El comando `rm` no permite borrar directorios como tal y nos da este error `rm: cannot remove 'dir1': Is a directory`. Al igual que pasa con el comando `cp` hay que indicarle que trabaje de forma recursiva para poder trabajar con directorios. La alternativa es el uso de comando `rmdir`.

▶ PREGUNTA: ¿Que pasa con `rmdir` al usarlo sobre `dir2`?

`dir2` no es un directorio vacío, y `rmdir` SOLO nos permite borrar directorios vacíos, y nos da este error: `rmdir: failed to remove 'directorio_copias': Directory not empty`.

15. Historial

```
pwd
ls
history
```

16. Usuarios

```
who
whoami
id
uname -a
```

▶ PREGUNTA: ¿Que hace el comando `who`?

Muestra qué usuarios están actualmente conectados al sistema y desde dónde:

- Quien está conectado, tu nombre de usuario
- El nombre del terminal o sesión
- La fecha y hora de la conexión

▶ PREGUNTA: ¿Que hace el comando `whoami`?

Te dice con qué usuario estás conectado, que es tu nombre de usuario.

▶ PREGUNTA: ¿Que hace el comando `id`?

Muestra el UID, GID y grupos a los que pertenece el usuario actual.

- UID: identificador único del usuario
- GID: identificador del grupo principal
- groups: lista de grupos a los que perteneces

▶ PREGUNTA: ¿Que hace el comando `uname -a`?

Muestra información del sistema: kernel, nombre de host, arquitectura y más.

- Sistema operativo
- Nombre del host
- Versión del kernel
- Arquitectura
- Sistema base

17. Super usuario

```
sudo su
```

▶ PREGUNTA: ¿Que ha ocurrido?

En el HPC no tenemos permisos de root / sudo, por lo que no podemos conectarnos como administrador. Esto supone un "problema" de cara a instalar software, pero veremos como trabajar con ello a lo largo del curso.

Nota final

- Podéis practicar estos ejercicios en cualquier ordenador, ya que estos comandos son universales y funcionan en toda máquina linux y similares, incluyendo macs y WSL.
- La manera más sencilla de practicarlos sin instalar nada es vía http://www.webminal.org/. En esta web puedes crearte un usuario de forma gratuita y abrir una terminal en una máquina remota, todo a través de vuestro explorador web. También contiene tutoriales complementarios que os pueden servir para afianzar lo aprendido hoy o repasar los comandos cuando tengáis necesidad de usarlos.
- Como alternativa podéis usar http://copy.sh/v86/?profile=archlinux, aunque esta carece de tutoriales.

```
Visita Webminal o copy.sh para practicar en casa: http://www.webminal.org/
o http://copy.sh/v86/?profile=archlinux
```