

## Manejar archivos

touch	Actualiza tiempos de acceso y modificación de un archivo. Si no existe, lo crea (Al archivo) a menos que se indique lo contrario.
↔	touch [OPCIÓN]... ARCHIVO
ls	Lista el contenido de un directorio. Entre sus opciones se encuentra permitir ver los permisos del listado de directorios y/o archivos.
↔	ls [OPCIÓN]... [ARCHIVO]...
rm	Borra ligas a archivos. Un archivo es eliminado cuando todas sus ligas han sido borradas, así que borrar la única liga a un archivo, lo elimina.
↔	rm [OPCIÓN]... [ARCHIVO]...
mv	Mueve uno o más archivos de un directorio (SOURCE) a otro (TARGET).
↔	mv [OPCIÓN]... SOURCE... TARGET
cp	Copia uno o más archivos.
↔	cp [OPCIÓN]... SOURCE... TARGET
cd	Cambia el directorio actual (Te mueve a un nuevo lugar en el sistema de archivos)
↔	cd [OPCIÓN] [DIRECTORIO]
mkdir	Crea un directorio.
↔	mkdir [OPCIÓN]... DIRECTORIO...
rmdir	Borra ligas a directorios.
↔	rmdir [OPCIÓN]... DIRECTORIO...
cat	Concatena archivos e imprime la concatenación en la salida estándar.
↔	cat [OPCIÓN]... [ARCHIVO]...
more	Permite examinar uno o más archivos en la terminal, una página a la vez.
↔	more [OPCIÓN]... ARCHIVO...
less	Hace lo mismo que <code>more</code> , pero añade capacidades de navegación como avanzar y retroceder.
↔	less [OPCIÓN] ARCHIVO...
find	Busca en partes especificadas del sistema de archivos de Unix, estos tienen que cazar con un cierto criterio.
↔	find DIRECTORIO [OPCIÓN] CRITERIO

wc	Cuenta el número de caracteres, palabras o líneas en un archivo o más. El resultado dependerá de las opciones que se pasen.
↔	wc [OPCIÓN]... [ARCHIVO]...

## Otros comandos

1

Otros comandos que se utilizan a menudo (Aunque no para manejar archivos) son los siguientes:

pwd	Imprime la ruta del directorio actual.
↔	pwd [OPCIÓN]...
top	Muestra los procesos que están corriendo actualmente.
↔	top [OPCIÓN]...
kill	Termina un proceso.
↔	kill [OPCIÓN]...
diff	Compara archivos línea por línea.
↔	diff [OPCIÓN]... [ARCHIVO]...
tree	Lista los contenidos de directorios en formato de árbol.
↔	tree [OPCIÓN]... [DIRECTORIO]...
uname	Imprime información del sistema.
↔	uname [OPCIÓN]...
grep	Imprime las líneas que hacen match con el patrón.
↔	grep [OPCIÓN]... PATRONES [ARCHIVO]...
exit	Se usa para cerrar la sesión actual del shell.
↔	exit [n]
tar	Se usa para crear y extraer archivos en Linux. Puede comprimir y almacenar varios archivos en uno solo.
↔	tar [OPCIONES PRINCIPALES]... [OPCIÓN]... [RUTA]...
echo	Muestra una línea de texto.
↔	echo [OPCIÓN]... [CADENA]...

<sup>1</sup>Algunos comandos tienen que descargarse con su manejador de paquetes

# Universidad Nacional Autónoma De México

## Facultad De Ciencias

### Propedéutico

#### Referencias de los comandos más usados en Unix

## Comandos básicos

	(Pipe) redirige la salida de un comando.
sudo COMANDO	Ejecuta el COMANDO en modo de superusuario.
man COMANDO	Despliega la ayuda para el COMANDO, aquí se pueden ver las OPCIONES del COMANDO.
>> archivo	Concatena al ARCHIVO preservando el contenido.
> ARCHIVO	Sobreescribe el contenido del ARCHIVO.
!!	Repite el último comando.

## Atajos de teclado

<b>ctrl-z</b>	Suspende la tarea actual.
<b>ctrl-c</b>	Termina la tarea actual.
<b>ctrl-l</b>	Limpia la pantalla.
<b>ctrl-r</b>	Nos permite buscar en el historial.

Para navegar por la línea de comandos (Como ir al inicio, avanzar por letras o palabras) existen atajos, que son muy parecidos a los de `emacs`, por lo que no los listaremos aquí.

## Permisos

`chmod` Cambia los permisos de lectura, escritura y ejecución.  
 ↪ `chmod [OPCIÓN]... ARCHIVO`  
`chown` Cambia el archivo de dueño y de grupo.  
 ↪ `chown [OPCIÓN]... ARCHIVO`  
`chgrp` Cambia el archivo de grupo.  
 ↪ `chgrp [OPCIÓN]... GRUPO ARCHIVO`

### Más sobre permisos

Antes de comenzar, necesitamos ejecutar el siguiente comando `ls -l`, que lo único que hace es listar los archivos que se encuentran en el directorio en el que estamos parados, y `-l` es una bandera (Que nos permite visualizar con más detalle nuestros archivos como la última fecha de modificación, los permisos, el propietario del archivo y el tamaño). Los permisos son los que nos interesan.

El comando nos arroja algo como lo siguiente:

```
drwxr-xr-x 2 propedeutico users 4096 jul 5 20:07 Imágenes
-rw-r--r-- 1 propedeutico users 12 nov 5 2019 Tarea1.pdf
```

Los permisos se dividen en cuatro bloques, el primero no siempre es necesario, solo indica el tipo de archivo:

- Indica que es un archivo.
- d Indica que es un directorio.
- I Indica que es un enlace a un archivo.

El segundo, tercer y cuarto bloque indican el propietario, el grupo y otros que se asocian al archivo. Estos bloques están ordenados en `rwX`, los cuales significan:

[ ] indica que es opcional. Al momento de escribir el comando, no son necesarios.  
 ... indica que se permite uno o más.

`r` (Read) Permite leer el contenido de un archivo.  
`w` (Write) Permite escribir contenido en un archivo, crear uno nuevo, listar archivos, renombrar y borrar archivos de una carpeta.  
`x` (Execute) Permite ejecutar cualquier archivo o script, también permite el acceso a carpetas y archivos.  
 - No se puede.

Puede estar - que quiere decir que no se tienen permisos de lectura, escritura o ejecución, o alguna combinación es estas. En el ejemplo del comando que se encuentra arriba `Tarea1.pdf` el segundo bloque que tiene los siguientes permisos `rw-`, quiere decir que el propietario (propedeutico) tiene permisos de lectura y escritura, pero no de ejecución.

### Roles de usuario y permisos

Arriba mencionamos que el segundo, tercer y cuarto bloque. Los sistemas Linux cuentan con tres roles, que justamente son estos.

Usuario Propietario.  
 Grupo Todos los miembros del grupo.  
 Otros Resto de usuarios.

### Modificación de permisos

Se utilizan dos tipos de notaciones, la simbólica y la octal, la diferencia es que la *simbólica* hace uso del alfabeto (letras) y la *octal* usa números.

**Simbólica** Definimos a los roles con las siguientes letras

`u` Usuario.  
`g` Grupo.  
`o` Otros.

Y a los permisos con las siguientes letras:

`r` Permiso de lectura.  
`w` Permiso de escritura.  
`x` Permiso de ejecución.

Por ejemplo, si quisiéramos cambiar los permisos para la `Tarea1.pdf`, lo haríamos de la siguiente manera.

```
chmod u=rwx,g=rx,o=w Tarea1.pdf
```

Donde le estamos dando permisos al propietario de lectura, escritura y ejecución, al grupo le damos permisos de lectura y ejecución y a otros le damos permisos de escritura. Podemos solo actualizar por partes, esto quiere decir, no poner algún rol.

**Octal** Haciendo uso de esta notación, cada permiso tiene un número predefinido.

Lectura (r) 4.  
 Escritura (w) 2.  
 Ejecución (x) 1.

Las combinaciones que podemos usar son las siguientes:

$7 = 4+2+1$  (`rwX`) (Read + Write + Execute).  
 $6 = 4+2$  (`rw-`) (Read + Write).  
 $5 = 4+1$  (`r-x`) (Read + Execute).  
 $4 = 4$  (`r-`) (Read).  
 $3 = 2+1$  (`-wx`) (Write + Execute).  
 $2 = 2$  (`-w-`) (Write).  
 $1 = 1$  (`-x`) (Execute).  
 $0 = 0$  (`-`) (None).

Si quisiéramos cambiar los permisos a `Tarea1.pdf` para que quedaran como en el ejemplo de arriba, haríamos lo siguiente.

```
chmod 752 Tarea1.pdf
```