



1 DE OCTUBRE DE 2018

**COMANDOS**  
SISTEMAS INFORMÁTICOS

JORDAN PERUCHO

1 DAM

# INDICE

Concepto de rutas .....	3
Comandos sobre ficheros y directorios de Linux.....	3
LS .....	3
CD .....	3
PWD .....	3
MKDIR.....	3
MV .....	4
CP.....	4
RM .....	4
Comodines en Linux .....	4
Comandos para visualizar y editar en Linux .....	5
CAT.....	5
MORE / LESS .....	5
TAIL / HEAD .....	5
ECHO.....	5
Comandos de búsqueda y patrones .....	6
EGREP .....	6
FIND .....	6
Comandos de información y filtrado .....	7
SORT .....	7
UNIQ.....	7
CUT .....	7
TR.....	7
WC .....	8
Comandos de permisos .....	8
CHMOD.....	8
CHOWN.....	8
CHGRP.....	8
USERADD .....	8
USERMOD .....	8
USERDEL .....	9
GROUPADD.....	9
GROUPDEL.....	9

Ejercicios Comandos.....	10
Ejercicios de Permisos .....	15

## Concepto de rutas

RUTA RELATIVA	RUTA ABSOLUTA
<b>1DAM/Asignaturas/Sistemas</b>	<b>/home/BIG/1DAM/Asignaturas/Sistemas</b>
Una ruta relativa es aquella que parte de la ubicación actual	Una ruta absoluta es aquella que parte desde la raíz del sistema de archivos

## Comandos sobre ficheros y directorios de Linux

### LS

Permite listar el contenido de un directorio.

**Syntax:** *ls [ opciones ] [ directorio | fichero ]*

Opciones:

-l	Muestra el contenido en formato bloque o lista
-R	Lista directorios y subdirectorios, de forma recursiva
-a	Lista también los ficheros o directorios ocultos
-h	Hace legible el tamaño de los ficheros
-i	Muestra el identificador del i-nodo asociado a cada elemento

### CD

Se utiliza para navegar entre directorios.

**Syntax:** *cd [ ruta ]*

### PWD

Muestra la ruta absoluta del directorio en el que estamos ubicados.

**Syntax:** *pwd*

### MKDIR

Crea directorios.

**Syntax:** *mkdir [ opciones ] [ ruta ]*

Opciones:

-p	En caso de no existir la ruta previa a la carpeta que se va a crear, también crea el resto de carpetas padre.
----	---

## MV

Mueve ficheros, pero también es usado para renombrar ficheros o directorios.

**Syntax:** *mv [ opciones ] [ ruta fuente ] [ ruta destino ]*

Opciones:

- I Ejecuta de forma interactiva, pregunta antes de sobrescribir el destino
- u Actualiza el destino con el fuente solo si este es mas reciente

## CP

Permite copiar un fichero o varios ficheros en un directorio.

**Syntax:** *cp [ opciones ] [ ruta fuente ] [ ruta destino ]*

Opciones:

- R Copia recursivamente un directorio
- i Forma interactiva, pregunta si sobrescribir el destino en caso de existir
- l Crea un enlace a los ficheros en vez de copiarlos (Accesos directos)

## RM

Borra ficheros o directorios.

**Syntax:** *rm [ opciones ] [ ficheros | directorios ]*

Opciones:

- r Borra recursivamente un directorio
- i Lo ejecuta de forma interactiva

## Comodines en Linux

- \* Sustituye el carácter por cualquier cadena de combinaciones posibles
- ? Sustituye un solo carácter
- [] Se comporta como el ? pero se le pueden indicar varios valores. Por ejemplo, el rango del 1 al 9, [1-9] o 5 caracteres concretos, [aeiou]

# Comandos para visualizar y editar en Linux

## CAT

Permite concatenar ficheros y imprime por pantalla su contenido

**Syntaxis:** `cat [ fichero ] [ opcional++ ]`

## MORE / LESS

Paginan uno o varios ficheros y los muestran por la salida estándar.

**Syntaxis:** `more [ fichero ] | less [ fichero ]`

## TAIL / HEAD

Retorna las líneas de un documento según le pidas, tail retorna desde bajo del documento y head desde el principio

**Syntaxis:** `tail -X [ fichero ] | head -X [ fichero ]`

## ECHO

Permite escribir una serie de cadenas y imprimirlas por la salida estándar.

**Syntaxis:** `echo "cadena de texto"`

-n	No añade el retorno al final del echo
-e	Activa la interpretación de los caracteres de barra invertida
\\b	Espacio hacia atras
\\c	Suprime la salida posterior
\\n	Línea nueva
\\r	Retorno de carro o de linea
\\t	Tabulador
\\v	Tabulador Vertical
\\	Escribe barra invertida

# Comandos de búsqueda y patrones

## EGREP

Permite filtrar contenido a base de patrones

**Syntax:** *egrep [opcion] [expresion]*

<b>-c</b>	Devuelve la cantidad de líneas que contienen el patrón
<b>-i</b>	Ignora diferencias entre mayúsculas y minúsculas
<b>-H</b>	Imprime el número de líneas y el nombre del fichero donde se encuentra el patrón
<b>-l</b>	Imprime el nombre del fichero donde se encuentra el patrón, pero no el numero de líneas
<b>-v</b>	Devuelve las líneas que no contienen el patrón
<b>-r</b>	Busca un directorio de forma recursiva
<b>-n</b>	Imprime el número de cada línea que contiene el patrón

## FIND

Permite buscar de forma recursiva en un directorio todos los ficheros que cumplan ciertas condiciones.

**Syntax:** *find [ruta] [opcion]*

<b>-name</b>	Permite especificar nombre de búsqueda
<b>-iname</b>	Permite especificar nombre de búsqueda ignorando mayúsculas/minúsculas
<b>-type</b>	Indica el tipo de fichero
<b>-size</b>	Indica el tamaño máximo y/o mínimo del fichero
<b>-perm</b>	Indica si tiene permisos dados
<b>-exec</b>	Define un comando a ejecutarse cada resultado de busqueda

# Comandos de información y filtrado

## SORT

Ordena las líneas de un fichero.

**Syntax:** *sort [opciones] [fichero]*

-t	Indica delimitador
-k	Indica columna a ordenar
-r	Ordena al revés
-f	Ignora diferencia de Mayúsculas y Minúsculas
-n	Ordena numericamente

## UNIQ

Elimina líneas repetidas. Va unido de la mano del sort

**Syntax:** *uniq [opcion]*

-u	Solo imprime las líneas que no esten duplicadas
-d	Solo imprime las líneas duplicadas

## CUT

Recorta líneas según un delimitador

**Syntax:** *cut [opciones] [fichero]*

-d	Indica delimitador
-f	Indica columna a cortar
-c	Selecciona solamente esos caracteres

## TR

Traduce expresiones

**Syntax:** *tr [opcion1] [opcion2]*



## WC

Imprime el número de líneas, palabras y bytes de uno o varios ficheros.

**Syntax:** *wc [opciones] [ficheros]*

-l	Cuenta líneas
-m	Cuenta caracteres
-c	Cuenta bytes
-w	Cuenta palabras

## Comandos de permisos

### CHMOD

Cambia los permisos de los ficheros

**Syntax:** *chmod [opcion] [usuario]*

### CHOWN

Cambia el propietario de un directorio o fichero

**Syntax:** *chown [opcion] [usuario]*

### CHGRP

Cambia el grupo de un directorio o fichero

**Syntax:** *chgrp [opcion] [usuario]*

### USERADD

Añade usuarios al sistema.

**Syntax:** *useradd [usuario]*

### USERMOD

Modifica el usuario.

**Syntax:** *usermod [opcion] [usuario]*

-s	Cambia la Shell por defecto
-G	Señala grupos secundarios
-e	Indica la fecha de expiración del usuario

## USERDEL

Elimina un usuario del sistema.

**Syntax:** *userdel [usuario]*

## GROUPADD

Añade grupos al sistema.

**Syntax:** *groupadd [grupo]*

## GROUPDEL

Elimina un grupo del sistema.

**Syntax:** *groupdel [grupo]*

## Ejercicios Comandos

1. Crear un directorio ejercicio1 en vuestro home, y copiar en el mismo todos los ficheros con extensión conf del directorio /etc. Realizad esta acción con un solo comando, y posicionados en el directorio home del usuario.

```
mkdir ejercicio1 && cp /etc/*.conf ejercicio1/
```

2. Crear un directorio ejercicio2 dentro de un directorio prueba dentro de un directorio control dentro de un directorio alumno dentro de vuestro home. Realizad esta acción con un solo comando y posicionados en el directorio home de vuestro usuario.

```
mkdir -p alumno/control/prueba/ejercicio2
```

3. Mover todos los ficheros que contengan una letra a en su nombre del directorio ejercicio1 al directorio ejercicio2. Un solo comando y posicionados en la raíz de vuestro árbol.

```
mv ejercicio1/*a* alumno/control/prueba/ejercicio2
```

4. Copiar el directorio alumno de vuestro home, y todo lo que contiene, a un directorio copia alumno que debe crearse en vuestro directorio home. Un solo comando para la copia.

```
cp -r alumno "copia alumno"
```

5. Borrar el directorio alumno de vuestro home con un solo comando y sin que pida confirmación.

```
rm -rf alumno
```

### FICHERO: amigos.txt

```
Pepe:Garcia:Garcia:h:33:canals
Paco:Dominguez:Cots:H:16:Xativa
Ana:Garcia:Pous:m:22:Alzira
Maria:Alonso:Alonso:M:18:Algemesi
Lola:Beltran:Osca:m:17:Moixent
ana:Benavent:Boil:M:44:Gavarda
pepe:Botella:Velasco:h:42:Alzira
paco:Tormo:Colomer:H:36:Xativa
Pepe:Romero:Casanova:H:55:Xativa
Paco:Alonso:Alonso:h:17:Guadassuar
pepe:Casasus:Domingo:H:24:Xativa
adela:Alabau:Sais:M:33:Antella
Herminia:Pastor:Pepe:M:100:Antella
Marc:Albentosa:Alabau:H:28:Guadassuar
```

1. Listar el nombre y apellidos de los amigos/as

```
cut -d: -f1-2 amigos.txt | tr ":" " "
```

2. Cuantos amigos/as hay en el IES.

```
Sort amigos.txt | wc -l
```

3. Cuantos amigos (chicos) hay en el IES.

```
cut -d: -f4 amigos.txt | egrep [Hh] | wc -l
```

4. Cuantas amigas (chicas) hay en el IES.

```
cut -d: -f4 amigos.txt | egrep [Mm] | wc -l
```

5. Cuantos amigos tienen de nombre pepe (cuidado con mayúsculas o minúsculas).

```
cut -d: -f1 amigos.txt | egrep -i "pepe" | wc -l
```

6. Muestra el nombre, apellidos y edad de amigos/as que tienen entre 20 y 29 años.

```
cut -d: -f1,2,5 amigos.txt | egrep [2][0-9]$ | tr ":" " "
```

7. Mostrar el nombre, los apellidos y la edad de los amigos (chicos) menores de edad, en la FP son los de 16 y 17 años (sin los dos puntos)

```
sort amigos.txt | egrep -i ":h:" | cut -d: -f1,2,5 | egrep [1][0-9]$ | tr : " "
```

8. Listado de los nombres diferentes que hay.

```
cut -d: -f1 amigos.txt | tr [A-Z] [a-z] | sort | uniq
```

9. Listado de los nombres de los amigos (chicos) y sin repeticiones.

```
cut -d: -f1,4 amigos.txt | egrep -i "h"$ | cut -d: -f1 | tr [A-Z] [a-z] | sort | uniq
```

10. Listado de los amigos/as (nombre, ape1, ape2, edad) ordenado por edad de menor a mayor (sin los dos puntos).

```
cut -d: -f1-3,4 amigos.txt | sort -t: -k4 -n | tr : " "
```

11. Listado de los nombres y edad de las amigas (chicas) ordenados por edad de mayor a menor (sin los dos puntos).

```
cut -d: -f1,4,5 amigos.txt | egrep -i ":m:" | cut -d: -f1,3 | sort -t: -k2 -n -r | tr : " "
```

12. Muestra el nombre, la edad y la población de la amiga (chica) de menor edad (los campos tabulados).

```
sort -t: -k5 -n amigos.txt | egrep -i ":m:" | head -1 | cut -d: -f1,5,6 | tr : "\t"
```

13. Muestra el nombre, la edad y la población del amigo (chico) de mayor (los campos tabulados).

```
sort -t: -k5 -n -r amigos.txt | egrep -i ":h:" | head -1 | cut -d: -f1,5,6 | tr : "\t"
```

14. Listado de los nombres de pueblos diferentes que hay.

```
cut -d: -f6 amigos.txt | tr [A-Z] [a-z] | sort | uniq
```

15. Presenta los nombres de pueblos que no se repiten.

```
cut -d: -f6 amigos.txt | tr [A-Z] [a-z] | sort | uniq -u
```

16. Presenta los nombres de pueblos que si se repiten.

```
cut -d: -f6 amigos.txt | tr [A-Z] [a-z] | sort | uniq -d
```

17. Muestra los nombres de los pueblos que empiecen por vocal (sin repeticiones).

```
cut -d: -f6 amigos.txt | tr [A-Z] [a-z] | sort | uniq | egrep -i ^[aeiou]
```

18. Muestra los nombres de los pueblos que empiezan por consonante (sin repeticiones).

```
cut -d: -f6 amigos.txt | tr [A-Z] [a-z] | sort | uniq | egrep -i ^[^aeiou]
```

19. Muestra los tres primeros caracteres de los nombres de los pueblos (sin repeticiones)

```
cut -d: -f6 amigos.txt | tr [A-Z] [a-z] | sort | uniq | cut -c 1-3
```

#### FICHERO: vecinos.txt

```
Alfonso:garcia:garcia:h:33:camionero:1200:11111111Q:ene
carla:bou:bou:m:18:administrativa:900:22222222M:Ene
alfonso:rus:terol:h:55:excalcalde:12000:33333333G:Jul
nacho:vidal:vidal:H:44:actor:11000::Mar
robert:pous:pous:H:22:Actor:2000:44444444F:sep
susana:cabri:cabra:M:25:administrativa:1500:55555555Q:dic
Elena:canales:garcia:M:25:administrativa:1500:66666666Q:may
Erika:año:company:M:25:Administrativa:1500:99999999Q:feb
Ana:camps:otero:m:6:jugar:40:1212Q:may
rosa:boix:colomer:M:5:Jugar:20:1234Q:feb
Enelda:vives:clos:m:100:Jubilada:900:11222333M:Jul
```

1. Lista el nombre, apellidos y profesión de todos (espacio en blanco entre los campos)

```
cat vecinos.txt | cut -d":" -f1-3,6 | tr ":" " "
```

2. Cuantas vecinas tenemos

```
cat vecinos.txt | cut -d":" -f4 | egrep "[Mm]" | wc -l
```

3. Profesiones diferentes de vecinos/as

```
cat vecinos.txt | cut -d":" -f6 | sort | uniq -i | wc -l
```

4. Mostrar los vecinos/as (nombre, apellidos y la edad ordenada de mayor a menor sin :) y todos los campos tabulados (tr ":" "\t")

```
sort -n -t":" -k5 vecinos.txt | cut -d":" -f1-3,5 | tr ":" "\t"
```

5. Tenemos que elegir al presidente y hacer una votación, necesitamos saber cuáles son los tres vecinos más jóvenes para formar la mesa de las votación (el nombre, apellidos y su edad)

```
sort -n -t":" -k5 vecinos.txt | cut -d":" -f1-3,5 | head -3 | tr ":" "\t"
```

6. Al final no se hace la votación y se decide que sea presidente el que más sueldo cobre y que aparezca su nombre y apellidos en mayúscula

```
sort -n -t":" -k7 vecinos.txt | cut -d":" -f1-3,7 | tail -1 | tr "[a-z]" "[A-Z]" | tr ":" "\t"
```

7. Listado de los vecinos entre 30 y 49 años (nombre, apellidos y su edad sin ":" y tabulado)

```
sort -n -t":" -k5 vecinos.txt | cut -d":" -f1-3,5 | tr ":" "\t" | egrep [34][0-9]
```

8. Hombres cuyo nombre empiece por vocal (nombre, apellidos y su profesión sin :)

```
cat vecinos.txt | cut -d":" -f1-4 | tr ":" "\t" | egrep "[Hh]$" | egrep -i "^[aeiou]"
```

9. nombre de los vecinos/as que empiece por c

```
cat vecinos.txt | egrep "^c" | cut -d":" -f1
```

10. nombre de los vecinos/as que empiece por vocal

```
cat vecinos.txt | egrep -i "^[aeiou]" | cut -d":" -f1
```

11. nombre de los vecinos/as que acabe en consonante

```
cat vecinos.txt | cut -d":" -f1 | egrep -i "[^aeiou]$"
```

12. nombre de los vecinos/as que empiece y acabe en vocal

```
cat vecinos.txt | cut -d":" -f1 | egrep -i ^[aeiou] | egrep -i [aeiou]$
```

13. nombre, apellidos y mes de nacimiento de los vecinos que nacieron en el primer trimestre del año (ene, feb y mar), además los vecinos tiene que salir ordenados por mes de menor a mayor y sin el separador ":" y tabulado, y además todas las letra en mayúsculas.

```
cat vecinos.txt | cut -d":" -f1-3,9 | egrep -i ":(ene|feb|mar)"
```

14. Que vecinos tiene un sueldo de 5 dígitos (nombre, apellidos y sueldo tabulados) y ordenados de mayor a menor

```
sort -n -t":" -k7 vecinos.txt | cut -d":" -f1-3,7 | tr ":" "\t" | egrep [0-9]....$
```

15. Que vecinos/as tienen el DNI correcto formato 8 dígitos numéricos y una letra xxxxxxxxL(mostrar el nombre, apellidos y su DNI, separado por espacios en blanco)

```
cut -d":" -f1-3,8 vecinos.txt | egrep -i "[0-9].....[a-z]$" | tr ":" "\t"
```

16. Que vecinos/as tienen el DNI incorrecto (mostrar el nombre, apellidos y su DNI tabulados y todo en mayúscula)

```
cut -d":" -f1-3,8 vecinos.txt | egrep -iv "[0-9]{8}[a-z]" | tr ":" "\t" | tr [a-z] [A-Z]
```

#### FICHERO: red.txt

```
pc01-192.168.0.33-1A:22:33:44:55:1F-Jun-2009
pc2-172.168.0.44-1B:22:33:44:55:1G-Feb-2008
pc03-100.168.0.55-1C:22:33:44:55:1F-Ene-2010
pc04-192.168.0.66-1D:22:33:55:1F-Ene-2009
pc10-172.168.0.7a7-1F:22:33:44:55:1F-Jul-2018
pc11-192.168.0.88-1Z:22:33:44:55:1F-Abr-2012
pc20-172.168..99-A1:22:33:44:55:1F-May-2011
pc24-102.168.0.101-A2:22:33:44:FF-Dic-2009
pc46-142.168.a.112-A3:22:33:66:FF:2B-Sep-2008
pc662-192.168.0.116-A4:22:33:88:4F:3D-Mar-2015
pcabc-192.168.0.112-B1:22:33:88:4F:3D-FEB-2014
```

1. PCs que hayan sido instalados en el primer trimestre.

```
cut -d":" -f1,4 red.txt | egrep -i "Sep|Oct|Nov|Dic"
```

2. Listado tabulado del nombre, mes y año de instalación de los PCs ordenados por año.

```
cut -d":" -f1,4,5 red.txt | sort -t":" -k3 -n | tr ":" "\t"
```

- El nombre del PC tiene que seguir el siguiente formato: pcx, pcxx o pcxxx donde x es un dígito numérico. Comprueba que todos los nombres de los Pcs siguen este formato. Los nombres que no sean correctos guárdalos en el fichero renombrar.txt.

```
cut -d"-" -f1 red.txt | egrep -v "pc[0-9]|pc[0-9][0-9]|pc[0-9][0-9][0-9]" >> renombrar.txt
```

- Muestra la MAC que sean correctas en su formato XX:XX:XX:XX:XX:XX donde XX puede ser cualquier número (0-9) y letras (A-F).

```
cut -d"-" -f3 red.txt | egrep -i "[0-F][0-F]:[0-F][0-F]:[0-F][0-F]:[0-F][0-F]:[0-F][0-F]:[0-F][0-F]"
```

- Muestra la MAC que no sean correctas.

```
cut -d"-" -f3 red.txt | egrep -iv "[0-F]{2}:[0-F]{2}:[0-F]{2}:[0-F]{2}:[0-F]{2}:[0-F]{2}"
```

- Muestras las IPs que sean correctas en su formato XXX.XXX.XXX.XXXX donde XXX puede ser cualquier número entre 0 y 9, y puede haber una X o dos XX o tres XXX Ej: 192.44.1.88.

```
egrep -i "[0-9]{1,3}\.){3}([0-9]{1,3}\-)" red.txt | cut -d- -f2
```

- Muestras las IPs que no sean correctas.

```
egrep -iv "[0-9]{1,3}\.){3}([0-9]{1,3}\-)" red.txt | cut -d- -f2
```

## Ejercicios de Permisos

- Crea el directorio prueba y prueba1 y compara sus permisos.

```
mkdir prueba prueba1  
ls -l | cut -d" " -f1,9  
drwxr-xr-x prueba  
drwxr-xr-x prueba1
```

- Crea el archivo prueba.txt y escribe algo dentro de él.

```
echo "hola" > prueba.txt
```



3. Compara los permisos que tienen el directorio prueba y el archivo prueba.txt ¿Porque los directorios tienen diferentes permisos que los archivos?

```
ls -l
drwxrwxr-x 2 big big 4096 oct 24 22:35 prueba
drwxrwxr-x 2 big big 4096 oct 24 22:35 prueba1
-rw-rw-r-- 1 big big 5 oct 24 22:39 prueba.txt
```

La diferencia de permisos esta dada por lo siguiente, para empezar el primer carácter indica el tipo de objeto. "d" de Directorio y "-" de fichero. Después para cada grupo tiene unos permisos diferentes porque para los directorios la "x" de ejecución significa acceso al directorio y para los ficheros ejecución de programa. La "w" significa poder escribir en el directorio y para el fichero modificar el contenido. Y la "r" permite listar con ls la carpeta y para los archivos poder leer el contenido.

4. Modificar la umask para que cuando creemos directorios los usuarios del grupo no puedan escribir dentro de él y los otros no puedan ni escribir ni entrar. Crea el directorio prueba2

```
umask 023
mkdir prueba2
ls -l
drwxrwxr-x 2 big big 4096 oct 24 22:35 prueba
drwxrwxr-x 2 big big 4096 oct 24 22:35 prueba1
drwxr-xr-- 2 big big 4096 oct 24 22:54 prueba2
-rw-rw-r-- 1 big big 5 oct 24 22:39 prueba.txt
```

5. Modificar los permisos de prueba2 (rwx r-x r--) con letras para que sean iguales a los de prueba y prueba1 (rwx rwx r-x). Letras (u,g,o,a añadir + y quitar - r,w,x).

```
chmod g+w,o+x prueba2
ls -l
drwxrwxr-x 2 big big 4096 oct 24 22:35 prueba
drwxrwxr-x 2 big big 4096 oct 24 22:35 prueba1
drwxrwxr-x 2 big big 4096 oct 24 22:54 prueba2
-rw-rw-r-- 1 big big 5 oct 24 22:39 prueba.txt
```

6. Modificar los permisos de prueba2 (rwx rwx r-x) con números para que sean rwx r-x r--.

```
chmod 754 prueba2
ls -l
drwxrwxr-x 2 big big 4096 oct 24 22:35 prueba
drwxrwxr-x 2 big big 4096 oct 24 22:35 prueba1
drwxr-xr-- 2 big big 4096 oct 24 22:54 prueba2
-rw-rw-r-- 1 big big 5 oct 24 22:39 prueba.txt
```

7. Quitamos (con letras) al usuario propietario el permiso de lectura del directorio prueba, lista el contenido de este directorio.

```
chmod u-r prueba
ls -l prueba
ls: no se puede abrir el directorio 'prueba': Permiso denegado
```

8. Si no se ha podido realizar la acción anterior restablecer el permiso (letras) e inténtalo otra vez.

```
chmod u+r prueba
ls -l prueba
total 0
```

9. Quitamos (con letras) al usuario propietario el permiso de escritura del directorio prueba, entrar en prueba y crear el directorio prueba1.

```
chmod u-w prueba
cd prueba
mkdir prueba1
mkdir: no se puede crear el directorio «prueba1»: Permiso denegado
```

10. Si no se ha podido realizar la acción anterior restablecer el permiso (letras) e inténtalo otra vez.

```
cd ..
chmod u+w prueba
cd prueba
mkdir prueba1
ls -l
drwxrwxr-x 2 big big 4096 oct 24 23:35 prueba1
```

11. Quitamos (con letras) al usuario propietario el permiso de ejecución del directorio prueba e intenta entrar dentro de él.

```
cd ..
chmod u-x prueba
cd prueba
bash: cd: prueba: Permiso denegado
```

12. Si no se ha podido realizar la acción anterior restablecer el permiso (letras) e inténtalo otra vez.

```
chmod u+x prueba
cd prueba
pwd
/home/big/prueba
```

13. Quitamos (con números) al usuario propietario el permiso de lectura del archivo prueba.txt, muestra el contenido de este archivo.

```
cd..  
chmod 264 prueba.txt  
cat prueba.txt  
cat: prueba.txt: Permiso denegado
```

14. Si no se ha podido realizar la acción anterior restablecer el permiso (con números) e inténtalo otra vez.

```
chmod 664 prueba.txt  
cat prueba.txt  
hola
```

15. Quitamos (con números) al usuario propietario el permiso de escritura del archivo prueba.txt, edita el archivo y añade o borra algo de su contenido.

```
chmod 464 prueba.txt  
echo "adios" >> prueba.txt  
bash: prueba.txt: Permiso denegado
```

16. Si no se ha podido realizar la acción anterior restablecer el permiso (con números) e inténtalo otra vez.

```
chmod 664 prueba.txt  
echo "adios" >> prueba.txt  
cat prueba.txt  
hola  
adios
```

17. ¿El archivo prueba.txt tiene permiso de ejecución? ¿Por qué? ¿Si tuviera permiso de ejecución se podría ejecutar?

No, porque no es un ejecutable. Si se podría ejecutar, pero daría error en la sintaxis del contenido.

18. Si quisiéramos ejecutar un archivo que fuera ejecutable y que no tuviera los permisos necesarios para que el usuario y el grupo lo pudieran ejecutar ¿qué comando ejecutaríamos (con números)?

```
chmod 110 fichero
```

19. En un directorio vacío (nuevo), crear 9 archivos (archiv1, archiv2, etc.) utilizando el comando touch. Quitarle todos los permisos con el comando "chmod a-rwx archiv\*".

```
touch archiv{1..9}
chmod a-rwx archiv*
ls -l
----- 1 big big 0 oct 25 00:33 archiv1
----- 1 big big 0 oct 25 00:33 archiv2
----- 1 big big 0 oct 25 00:33 archiv3
----- 1 big big 0 oct 25 00:33 archiv4
----- 1 big big 0 oct 25 00:33 archiv5
----- 1 big big 0 oct 25 00:33 archiv6
----- 1 big big 0 oct 25 00:33 archiv7
----- 1 big big 0 oct 25 00:33 archiv8
----- 1 big big 0 oct 25 00:33 archiv9
```

20. Modificar los permisos usando el operador '=' del 'chmod', para que queden de la siguiente manera:

<b>chmod u=rwx,go= archiv1</b>	archiv1 -rwx-----
<b>chmod u=rw archiv2</b>	archiv2 -rw-----
<b>chmod a=rwx archiv3</b>	archiv3 -rwxrwxrwx
<b>chmod u=rwx,g=rw,o=r archiv4</b>	archiv4 -rwxrw-r—
<b>chmod u=rwx,g=r archiv5</b>	archiv5 -rwxr-----
<b>chmod u=rx,g=rw,o=r archiv6</b>	archiv6 -r-xrw-r—
<b>chmod u=r,o=x archiv7</b>	archiv7 -r-----x
<b>chmod u=rw,go=r archiv8</b>	archiv8 -rw-r--r—
<b>chmod ug=rw,o=r archiv9</b>	archiv9 -rw-rw-r--

21. Modificar los permisos de los archivos anteriores utilizando los operadores + y - del 'chmod' para que queden de la siguiente manera (Los cambios son relativos a los archivos del ejercicio anterior):

<b>chmod o+r archiv1</b>	archiv1 -rwx---r--
<b>chmod u-w archiv2</b>	archiv2 -r-----
<b>chmod a-x archiv3</b>	archiv3 -rw-rw-rw-
<b>chmod go-r archiv4</b>	archiv4 -rwx-w----
<b>chmod g-r,o+wx archiv5</b>	archiv5 -rwx----wx
<b>chmod u+w,o-r archiv6</b>	archiv6 -rwxrw----
<b>chmod u+w,g+x,o+w-x archiv7</b>	archiv7 -rw---x-w-
<b>chmod u-rw,g-r archiv8</b>	archiv8 -----r--
<b>chmod u+x,g-rw,o-r archiv9</b>	archiv9 -rwx-----

22. Crear 9 archivos (num1, num2, etc) utilizando el comando touch.

```
touch num{1..9}
ls -l n*
-rw-rw-r-- 1 big big 0 oct 25 01:29 num1
-rw-rw-r-- 1 big big 0 oct 25 01:29 num2
-rw-rw-r-- 1 big big 0 oct 25 01:29 num3
-rw-rw-r-- 1 big big 0 oct 25 01:29 num4
-rw-rw-r-- 1 big big 0 oct 25 01:29 num5
-rw-rw-r-- 1 big big 0 oct 25 01:29 num6
-rw-rw-r-- 1 big big 0 oct 25 01:29 num7
-rw-rw-r-- 1 big big 0 oct 25 01:29 num8
-rw-rw-r-- 1 big big 0 oct 25 01:29 num9
```

23. Sobrescribir los permisos utilizando el comando `chmod` con argumento numérico (octal) para que queden de la siguiente manera:

<b><code>chmod 421 num1</code></b>	<code>num1 -r---w---x</code>
<b><code>chmod 000 num2</code></b>	<code>num2 -----</code>
<b><code>chmod 777 num3</code></b>	<code>num3 -rwxrwxrwx</code>
<b><code>chmod 564 num4</code></b>	<code>num4 -r-xrw-r--</code>
<b><code>chmod 740 num5</code></b>	<code>num5 -rwxr-----</code>
<b><code>chmod 644 num6</code></b>	<code>num6 -rw-r--r--</code>
<b><code>chmod 645 num7</code></b>	<code>num7 -rw-r--r-x</code>
<b><code>chmod 764 num8</code></b>	<code>num8 -rwxrw-r--</code>
<b><code>chmod 700 num9</code></b>	<code>num9 -rwx-----</code>

24. Con una sola instrucción, quitar permisos de lectura, escritura y ejecución para "otros" a todos los archivos del último ejercicio.

```
chmod o-rwx num*
```

25. Crear un directorio y quitarle todos los permisos de ejecución. Explicar qué pasa al intentar entrar al directorio con el comando `cd`. Explicar el significado de los permisos `r`, `w` y `x` en directorios.

```
chmod a-x ejercicio  
cd ejercicio  
bash: cd: ejercicio/: Permiso denegado
```

En directorios:

- `r`. Sirve para listar el directorio
- `w`. Sirve para escribir o modificar en el directorio
- `x`. Sirve para acceder al directorio

26. Informarse sobre los grupos a los que pertenece su usuario.

```
groups big  
big adm cdrom sudo dip plugdev lpadmin sambashare
```

27. Utilizando los comandos `chown` y `chgrp`, intentar cambiar el propietario y el grupo del archivo "num3". ¿Cuál es el problema?

**`chown root num3`**

*`chown: cambiando el propietario de 'num3': Operación no permitida`*

**`chgrp root num3`**

*`chgrp: cambiando el grupo de 'num3': Operación no permitida`*

Necesitamos permisos de administrador para realizar la operación

28. Transforma los siguientes permisos simbólicos en absolutos.

<b><code>rwXr-Xr-X</code></b>	<b>755</b>
<b><code>rw-r-----</code></b>	<b>640</b>
<b><code>r-Xr--r--</code></b>	<b>544</b>
<b><code>r-X--X--X</code></b>	<b>511</b>

29. Transforma los siguientes permisos absolutos en simbólicos

<b>644</b>	<b><code>rw-r--r--</code></b>
<b>755</b>	<b><code>rwXr-Xr-X</code></b>
<b>610</b>	<b><code>rw---X---</code></b>
<b>631</b>	<b><code>rw--XW--X</code></b>