

## Tarea 1 Teoría de Sistemas Operativos

### Melanie Pérez Santibáñez

a) Estudie y explique para qué sirven los comandos **ls**, **cat**, **chmod**, **echo**, **grep**, **cp**, **mv**, **rm** y **wc**. Dé ejemplos de uso. Para el comando **ls** averigüe para qué sirven las opciones **-l**, **-t** y **-a**. De ejemplo de uso para cada uno de esas opciones y la combinación de ellas. Ayuda: use el comando **man**, ej. **man ls**.

1. El comando **ls** sirve para listar el contenido del directorio actual donde nos encontramos o se puede especificar el directorio que queremos listar.

Aquí un ejemplo de cómo utilizarlo:

```
melanie@melanie:~$ ls
casagatos  enlace-simbolico  manchas.txt  repeticiones.txt  tarea1  wampo.txt
melanie@melanie:~$ ls casagatos
manchas.txt  nuevodir  wampo.txt
```

- El parámetro **ls -l** sirve para listar los archivos de una manera más detallada, indicando permisos, números rígidlos, el nombre del propietario, grupo al que pertenece, su tamaño en bytes, y fecha de última actualización.

```
melanie@melanie:~$ ls -l
total 12
drwxrwxr-x 2 melanie melanie 4096 Oct  1 01:58 casagatos
lrwxrwxrwx 1 melanie melanie  11 Oct  1 01:54 enlace-simbolico -> manchas.txt
-rwxrwxrwx 1 melanie melanie  37 Oct  1 01:05 manchas.txt
-rw-rw-r-- 1 melanie melanie  32 Oct  1 01:51 wampo.txt
melanie@melanie:~$ _
```

- El parámetro **ls -t** sirve para ordenar los archivos por tiempo de modificación en vez de alfabéticamente.

```
melanie@melanie:~$ ls
casagatos  enlace-simbolico  manchas.txt  wampo.txt
melanie@melanie:~$ ls -t
casagatos  enlace-simbolico  wampo.txt  manchas.txt
melanie@melanie:~$
```

- El parámetro **ls -a** lista todos los archivos incluyendo los que inician con un “.”

```
melanie@melanie:~$ ls -a
.  .bash_logout  .cache  enlace-simbolico  manchas.txt  wampo.txt
.. .bashrc      casagatos  .local           .profile
```

2. El comando **cat** sirve para concatenar, fusionar o imprimir archivos en la pantalla de salida o en otro archivo. Por ejemplo en la siguiente imagen se muestra el contenido de dos archivos.

```
melanie@melanie:~$ cat manchas.txt
hola, soy un gato y me llamo manchas
melanie@melanie:~$ cat wampo.txt
hola, soy wampo y soy otro gato
melanie@melanie:~$
```

3. El comando **chmod** se utiliza para cambiar los permisos de archivos o directorios. En la imagen que se muestra se puede ver que en un comienzo el archivo `manchas.txt` tiene asignados todos los permisos y luego con el comando `chmod 0755` se cambian esos permisos para otorgar sólo los permisos de lectura y ejecución para el usuario dueño del archivo.

```
melanie@melanie:~$ ls -l manchas.txt
-rwxrwxrwx 1 melanie melanie 37 Oct  1 01:05 manchas.txt
melanie@melanie:~$ chmod 0755 manchas.txt
melanie@melanie:~$ ls -l manchas.txt
-rwxr-xr-x 1 melanie melanie 37 Oct  1 01:05 manchas.txt
```

4. El comando **echo** se utiliza para imprimir un texto en pantalla, y es utilizado en pequeños programas llamados scripts. En la imagen se puede ver cómo funciona el comando.

```
melanie@melanie:~$ echo hola mundo
hola mundo
melanie@melanie:~$ echo -e "\bhola \bmundo \bsoy \bmanchas \bel \bgato"
holamundosoymanchaselgato
melanie@melanie:~$ echo -e "\nhola\nmundo\nsoy\nmanchas\nel\ngato"

hola
mundo
soy
manchas
el
gato
```

5. El comando **grep** lee una entrada estándar o una lista de archivos, e imprime las líneas que contengan coincidencias para la expresión buscada. En el ejemplo se creó un archivo que contenía la palabra "palabras" en 3 ocasiones y las destaca, y luego con el comando `grep -l palabras ./` nos indica en qué archivo se encuentra la palabra "palabras".

```
melanie@melanie:~$ grep palabras repeticiones.txt
este es un archivo para repetir palabras, donde aparecerán muchas palabras repetidas para demostrar
lo que hace el comando grep, que es buscar palabras
melanie@melanie:~$ grep -l palabras ./
grep: ./casagatos: Is a directory
./repeticiones.txt
```

6. El comando **cp** sirve para copiar un archivo o una carpeta. En el ejemplo se copia el archivo wampo.txt dentro del directorio casagatos.

```
melanie@melanie:~$ cp wampo.txt ./casagatos/  
melanie@melanie:~$ ls casagatos  
manchas.txt wampo.txt
```

7. El comando **mv** es usado para mover o renombrar archivos o directorios. En el ejemplo utilizamos el comando mv para renombrar el archivo manchitas.txt como manchas.txt.

```
casagatos enlace-simbolico manchitas.txt repeticiones.txt wampo.txt  
melanie@melanie:~$ mv manchitas.txt manchas.txt  
melanie@melanie:~$ ls  
casagatos enlace-simbolico manchas.txt repeticiones.txt wampo.txt
```

8. El comando **rm** se utiliza para eliminar archivos y directorios de archivos. Este comando se debe utilizar con cuidado porque no pide confirmación para eliminar el archivo por lo que se debe tener certeza de la ubicación y nombre del archivo a eliminar. Para el ejemplo se creó un archivo llamado eliminame.txt que se muestra en un comienzo en el directorio y luego de utilizar el comando desaparece.

```
melanie@melanie:~$ ls  
casagatos eliminame.txt enlace-simbolico manchas.txt wampo.txt  
melanie@melanie:~$ rm eliminame.txt  
melanie@melanie:~$ ls  
casagatos enlace-simbolico manchas.txt wampo.txt  
melanie@melanie:~$ _
```

9. El comando **wc** se utiliza para contabilizar palabras, caracteres o saltos de líneas. Para el ejemplo se utilizó el comando **wc -m** para contabilizar caracteres y el comando **wc -w** para contabilizar palabras, ambos se utilizaron en el archivo repeticiones.txt

```
melanie@melanie:~$ wc -m repeticiones.txt  
152 repeticiones.txt  
melanie@melanie:~$ wc -w repeticiones.txt  
24 repeticiones.txt
```

b) Explique qué son los metacaracteres y dé ejemplos de uso de ellos.

Los metacaracteres son un conjunto de caracteres especiales utilizados para realizar búsquedas u otras operaciones en ficheros o directorios. Por ejemplo el signo de exclamación ! se utiliza para negar o excluir caracteres, otro ejemplo es el asterisco \* que es utilizado para sustituir cualquier carácter o conjunto de caracteres. En la imagen se ejemplifica el uso del carácter \* para buscar el archivo donde se encuentra la palabra “palabras” dentro del directorio.

```
melanie@melanie:~$ grep palabras repeticiones.txt
este es un archivo para repetir palabras, donde aparecerán muchas palabras repetidas para demostrar
lo que hace el comando grep, que es buscar palabras
melanie@melanie:~$ grep -l palabras ./*
grep: ./casagatos: Is a directory
./repeticiones.txt
```

c) Explique en qué consiste la expansión por paréntesis de conjunto (Brace Expansion). Con esta herramienta, resuelve el siguiente problema:

En un directorio, se quieren crear subdirectorios para que almacenen respaldos diarios de todo un año. Debe tener la siguiente estructura :

```
directorio_actual/
+ 2021-01-01/
+ 2021-01-02/
+ 2021-01-03/
. . .
+ 2021-09-18/
. . .
+ 2021-10-18/
. . .
+ 2021-12-31/
```

Suponga que todos los meses tienen 31 días. Debe ejecutar UN sólo comando para crear la estructura de directorios solicitada.

La función Brace Expansion se traduce al español como expansión de llaves y se utiliza para generar combinaciones entre los caracteres que se ingresen dentro de las llaves, y se ordenan de izquierda a derecha.

Para resolver lo que nos piden crear un subdirectorio donde se almacenen los respaldos diarios de todo un año. Para realizar esta tarea utilizamos la función de expansión de llaves para crear combinaciones entre los días y meses del año, y antes de ellas se asigna el año al que pertenecen (esto se puede realizar con cualquier otro tipo de carácter pero para efectos de la tarea se utilizan números para asignar las fechas que se nos piden).

```
melanie@melanie:~$ mkdir tarea1
melanie@melanie:~$ ls
casagatos  enlace-simbolico  manchas.txt  repeticiones.txt  tarea1  wampo.txt
melanie@melanie:~$ cd tarea1
melanie@melanie:~/tarea1$ mkdir 2021-{1..12}-{1..31}
```

```
2021-10-2  2021-11-3  2021-1-25  2021-2-7  2021-4-21  2021-5-8  2021-7-22  2021-8-9
2021-10-20 2021-11-30 2021-12-5  2021-2-8  2021-4-22  2021-5-9  2021-7-23  2021-9-1
2021-10-21 2021-11-31 2021-1-26  2021-2-9  2021-4-23  2021-6-1  2021-7-24  2021-9-10
2021-10-22 2021-1-14  2021-12-6  2021-3-1  2021-4-24  2021-6-10  2021-7-25  2021-9-11
2021-10-23 2021-11-4  2021-1-27  2021-3-10  2021-4-25  2021-6-11  2021-7-26  2021-9-12
2021-10-24 2021-1-15  2021-12-7  2021-3-11  2021-4-26  2021-6-12  2021-7-27  2021-9-13
2021-10-25 2021-11-5  2021-1-28  2021-3-12  2021-4-27  2021-6-13  2021-7-28  2021-9-14
2021-10-26 2021-1-16  2021-12-8  2021-3-13  2021-4-28  2021-6-14  2021-7-29  2021-9-15
2021-10-27 2021-11-6  2021-1-29  2021-3-14  2021-4-29  2021-6-15  2021-7-30  2021-9-16
2021-10-28 2021-1-17  2021-12-9  2021-3-15  2021-4-3  2021-6-16  2021-7-31  2021-9-17
2021-10-29 2021-11-7  2021-1-3  2021-3-16  2021-4-30  2021-6-17  2021-7-31  2021-9-18
2021-10-3  2021-1-18  2021-1-30  2021-3-17  2021-4-31  2021-6-18  2021-7-4  2021-9-19
2021-10-30 2021-11-8  2021-1-31  2021-3-18  2021-4-4  2021-6-19  2021-7-5  2021-9-2
2021-10-31 2021-1-19  2021-1-4  2021-3-19  2021-4-5  2021-6-2  2021-7-6  2021-9-20
2021-10-4  2021-11-9  2021-1-5  2021-3-2  2021-4-6  2021-6-20  2021-7-7  2021-9-21
2021-10-5  2021-1-2  2021-1-6  2021-3-20  2021-4-7  2021-6-21  2021-7-8  2021-9-22
2021-10-6  2021-1-20  2021-1-7  2021-3-21  2021-4-8  2021-6-22  2021-7-9  2021-9-23
2021-10-7  2021-1-21  2021-1-8  2021-3-22  2021-4-9  2021-6-23  2021-8-1  2021-9-24
2021-10-8  2021-12-1  2021-1-9  2021-3-23  2021-5-1  2021-6-24  2021-8-10  2021-9-25
2021-10-9  2021-12-10 2021-2-1  2021-3-24  2021-5-10  2021-6-25  2021-8-11  2021-9-26
2021-1-1  2021-12-11 2021-2-10  2021-3-25  2021-5-11  2021-6-26  2021-8-12  2021-9-27
2021-1-10 2021-12-12 2021-2-11  2021-3-26  2021-5-12  2021-6-27  2021-8-13  2021-9-28
2021-1-11 2021-12-13 2021-2-12  2021-3-27  2021-5-13  2021-6-28  2021-8-14  2021-9-29
2021-1-11-1 2021-12-14 2021-2-13  2021-3-28  2021-5-14  2021-6-29  2021-8-15  2021-9-3
2021-1-10 2021-12-15 2021-2-14  2021-3-29  2021-5-15  2021-6-3  2021-8-16  2021-9-30
2021-1-11 2021-12-16 2021-2-15  2021-3-3  2021-5-16  2021-6-30  2021-8-17  2021-9-31
2021-1-12 2021-12-17 2021-2-16  2021-3-30  2021-5-17  2021-6-31  2021-8-18  2021-9-4
2021-1-13 2021-12-18 2021-2-17  2021-3-31  2021-5-18  2021-6-4  2021-8-19  2021-9-5
2021-1-14 2021-12-19 2021-2-18  2021-3-4  2021-5-19  2021-6-5  2021-8-2  2021-9-6
2021-1-15 2021-1-22 2021-2-19  2021-3-5  2021-5-2  2021-6-6  2021-8-20  2021-9-7
2021-1-16 2021-12-2  2021-2-2  2021-3-6  2021-5-20  2021-6-7  2021-8-21  2021-9-8
2021-1-17 2021-12-20 2021-2-20  2021-3-7  2021-5-21  2021-6-8  2021-8-22  2021-9-9
2021-1-18 2021-12-21 2021-2-21  2021-3-8  2021-5-22  2021-6-9  2021-8-23
2021-1-19 2021-12-22 2021-2-22  2021-3-9  2021-5-23  2021-7-1  2021-8-24
2021-1-12 2021-12-23 2021-2-23  2021-4-1  2021-5-24  2021-7-10  2021-8-25
2021-1-2  2021-12-24 2021-2-24  2021-4-10 2021-5-25  2021-7-11  2021-8-26
melanie@melanie:~/tarea1$
```

b) La interconexión de comandos a través de *pipes* permite construir, de forma muy simple, nuevas herramientas. Como ejemplo considere los comandos `ls` y `wc`, que interconectados permiten contar archivos del directorio actual: `ls | wc -l`. Mediante el uso de pipes resuelva:

1. Mediante `grep`, encontrar archivos cuyo nombre contenga el carácter `i` en el directorio `/bin`.

En el ejemplo se utiliza el comando **grep -l** para encontrar los archivos del directorio cuyo nombre contiene el carácter `i`.

```
melanie@melanie:~$ ls
casagatos  enlace-simbolico  manchas.txt  pia.txt  repeticiones.txt  tarea1  wampo.txt
melanie@melanie:~$ grep -l i ./.*
grep: ./casagatos: Is a directory
./pia.txt
./repeticiones.txt
grep: ./tarea1: Is a directory
```

2. Contar los archivos con una secuencia de permisos r-x en los directorios /bin y /usr/bin.

c) Las variables de ambiente definen aspectos del entorno de programación, y los comandos set y echo (mediante el metacaracter \$) permiten ver su contenido.

1. Investigue el uso de las variables HOME, SHELL, PATH, y PWD. ¿Cómo se puede visualizar su contenido?
  - La variable HOME hace referencia a la carpeta personal del usuario.
  - La variable SHELL es la encargada de interpretar cualquier comando que se ingrese. Mayoritariamente será bash de forma predeterminada, pero se pueden establecer otros valores de acuerdo a la preferencia del usuario.
  - La variable PATH almacena una lista de directorios en los cuales linux debe buscar un comando para poder ejecutarlo.
  - La variable PWD es el directorio actual de trabajo.

Logré visualizar el contenido de las variables HOME y PATH.

```
melanie@melanie:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
melanie@melanie:~$ echo $HOME
/home/melanie
```