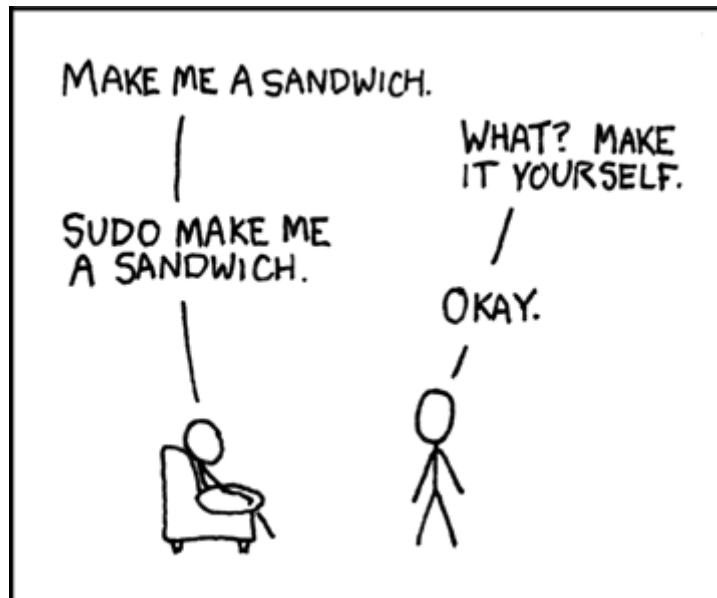


Práctica de Repaso Complementaria

Puedes buscar ayuda escribiendo `man nombre_del_comando`.

Recuerda que algunos comandos necesitan permisos de superusuario para ser ejecutados:



sudo te puede ayudar a hacerlo sin tener que cambiar de usuario

LISTADO DE ARCHIVOS

1. Lista recursivamente todos los archivos bajo el directorio /etc deteniendo el despliegue en cada pantalla.
2. ¿Qué diferencia a more y a less? Probar en la consola
3. Prueba otras opciones del comando ls usando el directorio /etc o /bin.
4. Lista archivos visibles e invisibles bajo el directorio actual, en formato largo.
5. Interpreta las siguientes líneas de salida del formato largo de salida del comando ls:

```
drwxr-xr-x 5 pedroso prod      512 Sep 11 15:09 articulos
-rw-r--r-- 1 pedroso prod     1430 Ago 10 07:56 docum1.prod
-rwxr-xr-x 3 admin1  adminfo 2330 Jan  7 09:34 rv
lrwxr--r-- 1 nicasio almacen 9500 Sep 11 10:56 listado.partes
```

REDIRECCIÓN ENTRADA/SALIDA

1. Sin editor de texto, mediante redirección, crea el archivo cap1 con el contenido "Este es el capítulo 1". Repite con cap2 y cap3. Verifica mostrando cada uno con el comando cat.
2. Añade al final de un archivo llamado libro el contenido de los archivos cap1, cap2 y cap3.
3. Crea un archivo llamado amigotes que contiene las líneas del archivo /etc/passwd ordenadas en orden descendente utilizando el comando sort y redirección de entrada y salida.

-
4. Muestra las líneas del fichero `amigotes` que tienen la palabra `home`, filtrando el resto mediante el comando `grep`.

EDITOR DE TEXTO

No hay un único editor de texto, en este curso hemos estudiado `nano` (que es más sencillo para empezar a aprender) pero si quieren investigar `vi`, `vim` o `emacs` es recomendable.

1. Abre la ayuda o el man de `nano`. ¿Se puede copiar y pegar?, ¿cómo se puede borrar líneas enteras?
2. Crea en tu directorio personal con el editor de textos un `print hola mundo` en código `python`. Guardalo con el nombre `hola.py`

SISTEMA DE ARCHIVOS

1. Cambia al directorio `bin`, dentro del directorio `usr`, dentro del directorio raíz. Muestra el directorio actual y los archivos contenidos en él.
2. Cambia al directorio propio del usuario desde cualquier otro directorio.
3. Crea un directorio que se llame `amigos` dentro de tu directorio personal, de forma que cuando listes los contenidos con el comando `ls` veas el archivo `hola.py` y `amigos`.
4. Copia el archivo `hola.py` en el directorio que acabas de crear, manteniendo permisos y propietario.
5. Crea un enlace llamado `'saludo'` al archivo `hola.py`.
6. Renombra `/home/tunombredeusuario/amigos`, por `/home/tunombredeusuario/mis_amigos`.
7. Encuentra todos los archivos que terminan en `.py` mediante el comando `find`. (en `/`)
8. Calcula cuánto ocupa el directorio `/home` mediante el comando `du`.

USUARIOS

1. Muestra el nombre del usuario en la sesión actual, y después el nombre de la máquina. (`whoami` y `hostname`)
2. Muestra qué usuarios hay en el sistema en el momento actual. (`who`)
3. Crea un usuario llamado `pepito` con contraseña `grillo`, y añádelo al grupo `admin`. Después bórralo.

SEGURIDAD

1. El usuario `ana` desea cambiar su contraseña. ¿Cómo lo hace? ¿Debe pedir ayuda al administrador?
2. El usuario `ana` olvidó su contraseña. ¿Qué puede hacer por sí mismo? ¿Necesita ayuda del administrador? ¿Cómo puede el administrador averiguar la contraseña que el usuario olvidó? ¿Cómo hace el administrador para cambiar la contraseña del usuario `ana`?

PROCESOS

1. Muestra los procesos de tu sesión actual que se están ejecutando.
2. Muestra todos los procesos en tu sistema que se están ejecutando.
3. Crea un nuevo proceso al ejecutar el comando `more /etc/passwd` y páralo con `CTRL+Z`. Comprueba que aunque estés de nuevo en la línea de comandos dicho proceso sigue activo. Elimina definitivamente dicho proceso con el comando `kill`.

4. ¿Cómo se lanza un proceso o comando y, sin que éste haya terminado, accedemos de nuevo a la línea de comandos.
5. Comprueba el consumo de recursos del sistema por los procesos mediante el comando top (pulsar la h estando ahí te dará la ayuda de top)
6. Muestra al menos dos formas diferentes de ver los procesos en ejecución en el momento actual.

TAREAS

1. ¿Con qué combinación de teclas podemos pausar una tarea dentro de bash ?
2. ¿Cómo se lleva una tarea al fondo al momento de hacer el comando?
3. ¿Cómo hago para ver las tareas que están en el fondo?
4. ¿Cómo se trae la última tarea abierta al frente?
5. Si tengo varias tareas en el fondo ¿Cómo hago para seleccionar una específica para traerla al frente?
6. ¿Qué diferencia hay entre tareas de bash y procesos del sistema? ¿Hay alguna relación entre ellos?

COMODINES

Recordemos que la expansión de comodines, es otra de las tareas que lleva a cabo el intérprete de línea de comandos (a diferencia de la expresión regular que sirve como formato de argumento en algunos programas o comandos). Más detalle de esto se puede encontrar en el man de bash (bajo el título Patterns o Patrones)

- ***** Representa cualquier cadena de caracteres, incluyendo una vacía.
- **?** Representa cualquier carácter individual.
- **[]** Representa un conjunto de caracteres posibles. ([abcd] o [a-d] son lo mismo)
- **{ }** Representa un producto cartesiano de caracteres. No se tienen en cuenta los archivos de la carpeta actual, **se crean todas las posibles combinaciones entre los elementos que se listan dentro**. con touch {a..d} creo los archivos a b c y d, no es lo mismo hacer touch {abcd}

La expansión de ***** **?** y **[]** va a depender de lo que tenga sentido en el contexto, por ejemplo si uno desea listar los archivos que terminen en .py, usaremos *.py para hacer referencia a ellos, solo se expandirá a los nombres de carpeta y archivos que ya existen en esa carpeta en la que estoy.

Para ver más claramente esto ejecute en las siguientes instrucciones en una carpeta vacía:

```
alumno@pantera:~$ touch [1-9]
alumno@pantera:~$ ls
'[1-9]'
alumno@pantera:~$ touch [1..9]
alumno@pantera:~$ ls
'[1-9]' '[1..9]'
alumno@pantera:~$ touch {1-9}
alumno@pantera:~$ ls
'[1-9]' '[1..9]' {1-9}
alumno@pantera:~$ touch {1..9}
alumno@pantera:~$ ls
1 '[1-9]' '[1..9]' {1-9} 2 3 4 5 6 7 8 9
alumno@pantera:~$ █
```

En la siguiente esta sesión de terminal, se creó un directorio corchetes, se accedió a él y se crearon archivos file1.txt, file2.txt y file3.txt usando touch file{1..3}.txt. El intento inicial de listar los archivos con ls file[123].txt falló porque los archivos aún no existían. Tras crearlos, se mostró cómo los diferentes patrones de corchetes funcionan para listar archivos, y que ls file[1..3].txt no es una notación válida, a diferencia de ls file[1-3].txt que lista correctamente los archivos.

```
alumno@pantera:~$ mkdir corchetes
alumno@pantera:~$ cd corchetes/
alumno@pantera:~/corchetes$ ls
alumno@pantera:~/corchetes$ ls file[123].txt
ls: no se puede acceder a 'file[123].txt': No existe el archivo o el directorio
alumno@pantera:~/corchetes$ touch file{1..3}.txt
alumno@pantera:~/corchetes$ ls
file1.txt file2.txt file3.txt
alumno@pantera:~/corchetes$ ls file[123].txt
file1.txt file2.txt file3.txt
alumno@pantera:~/corchetes$ ls file[12].txt
file1.txt file2.txt
alumno@pantera:~/corchetes$ ls file[1..3].txt
file1.txt file3.txt
alumno@pantera:~/corchetes$ ls file[1-3].txt
file1.txt file2.txt file3.txt
alumno@pantera:~/corchetes$ █
```

EXPRESIONES REGULARES

Recordemos los diferentes operadores de las expresiones regulares:

- . cualquier carácter
- ? 0 ó 1 veces el carácter o la expresión anterior
- * 0 ó más veces el carácter o la expresión anterior
- + 1 ó más veces el carácter o la expresión anterior
- {n,m} entre n y m veces el carácter o la expresión anterior
- | escoger entre una u otra expresión
- ^ inicio de línea
- \$ final de línea

-
- - rango de letras o números
 - [] () agrupar
 - ^ negación (este no lo vimos, pero puede llegar a ser útil)
 - conjunto de caracteres:
[:alnum:] [:alpha:] [:digit:] [:upper:] [:lower:] [:blank:]

En el siguiente link encontrarás el log de un entrenamiento de aprendizaje automático:
https://raw.githubusercontent.com/aleoncavallo/entorno/main/training_log.txt

Para descargarlo en la consola se puede usar el comando wget
(recuerda también que para pegar en la consola se usa ctrl + mayús + v)

1. Filtra las líneas del log que contienen la palabra "accuracy"
2. De las líneas del log que contienen la palabra "accuracy", toma el valor de las mismas (pista: usar el comando cut luego del grep)
3. Filtra las líneas que contienen errores (ERROR)
4. Filtra las líneas que contienen advertencias (WARNING)
5. Encontrar líneas que indican el inicio o la finalización del entrenamiento
6. Buscar líneas que contienen una precisión (accuracy) mayor a 0.8 (pista: usar -E)
7. Buscar líneas que contengan valores de pérdida (loss) o precisión (accuracy)

Ahora con este archivo:

<https://raw.githubusercontent.com/aleoncavallo/entorno/main/messiesunperro.txt>

Encuentra las expresiones regulares que ayuden a encontrar

1. Palabras que tienen dos o más 'o' seguidas
2. Palabras cuya tercera letra es d
3. Palabras que contienen pan o plan
4. Palabras que acaban en cial
5. ¿Cuántas líneas vacías?
6. Líneas terminadas en guión medio '-' (atención que el - puede ser confundido con el indicador de que se está por indicar otra opción a bash)
7. Líneas que comienzan con dos dígitos
8. Fechas con el formato dd/mm/aaaa
9. Emails con el formato [nombre@dominio.xxx](#)

Si ves el archivo podrás notar que cada tanto hay dos dígitos que no tienen nada que ver con la historia, esos son los números de página del pdf del cual fue copiado.

Filtra las líneas que no comienzan con dos dígitos y redirige esa salida a otro archivo, debe quedar un archivo limpio de esos números de página.

¿Por qué nos sirve el comando (1) y el (2) no?

1. `grep -E ^[[digit:]]{2} messiesunperro.txt`
2. `grep ^[[digit:]]{2} messiesunperro.txt`

¿A cuál de los dos es equivalente este? `grep ^[[digit:]][[digit:]] messiesunperro.txt`

fuelle de la mayoría de las preguntas y repasos (ahí también está la respuesta de varios):
<https://acastan.gitbook.io/servicios/repaso/ejercicio-0c-repaso-linux-comandos-unix-ejercicio>

Puedes encontrar un buen curso sobre el funcionamiento de Linux y sus comandos en
<http://linuxjourney.com/>