

Realizado por: Edwin René Guamán Quinche

1-Cuál de las siguientes opciones usarías para decir la fecha? Por qué no usarías las otras opciones? Sugerencia: siempre es mejor almacenar en una variable para poderla usar posteriormente.

- `echo "Today is the `date +%j` of the year"`
Imprime: Today is the 336 of the year
Sólo imprime el número de día actual respecto a las 365 días del año
- `echo "Today is the `date +%j` of the year"`
imprime: Today is the `date +%j` of the year
Imprime la instrucción como cadena de datos
- `DT=`date +%A, %e %B, %Y`; echo '$DT'`
imprime: \$DT, se esta encerrando con comilla simple en la variable y se imprime como cadena
- `DT=`date +%A, %e %B, %Y`; echo "The date today is: $DT"`
imprime: The date today is: \$DT,
En \$DT se agrega \ y hace que \$ se considere un carácter para imprimir
- `DT=`date +%A, %e %B, %Y`; echo "The date today is: $DT"`
Imprime: The date today is: lunes, 2 diciembre, 2019
Forma correcta de imprimir la fecha, y además se almacena todo en una variable

2- Reciba dos nombres de archivos e imprima cuántas líneas tiene cada uno. Por ejemplo: `sh dosarchivos.sh /etc/passwd /etc/shadow`

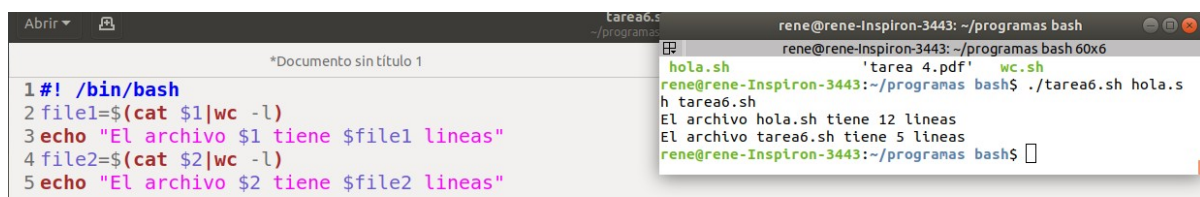
`/etc/passwd` tiene 25 líneas y `/etc/shadow` tiene 24 líneas

Sugerencia: dentro de los backticks se pueden usar variables y parámetros con normalidad).

Sugerencia2: Para contar líneas, podemos usar el comando "`wc -l`"

Código:

```
#!/bin/bash
file1=$(cat $1|wc -l)
echo "El archivo $1 tiene $file1 líneas"
file2=$(cat $2|wc -l)
echo "El archivo $2 tiene $file2 líneas"
```



```
rene@rene-Inspiron-3443: ~/programas bash
rene@rene-Inspiron-3443: ~/programas bash$ ./tarea6.sh hola.sh
El archivo hola.sh tiene 12 líneas
El archivo tarea6.sh tiene 5 líneas
rene@rene-Inspiron-3443: ~/programas bash$
```

3- Cree hacia `/root/` un respaldo de `/etc` con la fecha actual en su nombre. Por ejemplo el archivo de respaldo debe llamarse: `/root/etc-20160801.tar.gz`

Como pueden ver, el archivo no se llama etc.tar.gz simplemente sino que se llama etc-FECHA.tar.gz

Sugerencia: el comando date permite obtener la fecha en diversos formatos. En este caso lo pido en formato YYYYMMDD (año, mes, día)

Sugerencia2: Lógico que usaremos sustitución de comandos: backticks... para que el respaldo salga siempre con la fecha que invocamos al script de respaldo.

```
1 #! /bin/bash
2 fecha=`date +" %e-%m-%Y"`
3 archivo="$fecha"_etc.tar.gz"
4 tar -czvf /root/archivo/etc
5 echo "El archivo $archivo se ha crado correctamente"
```

4- El comando whoami permite conocer el usuario conque estamos. Guarde este comando hacia una variable y salude al usuario conque estamos. Ej, si estamos como root diría: Hola root!!!

```
1 #! /bin/bash
2 usuario=`whoami`
3 echo "Bienvenido $usuario"
```

5- Se puede usar \$() en vez e backticks? Demuéstrelo resolviendo uno de los anteriores ejercicios usando \$()

```
1 #! /bin/bash
2 mensaje=$(pwd)
3 echo "direccion: $mensaje"
```