

EJERCICIOS – shell script (Parte A-B)

1. Crear un shell script llamado **informacion.sh** que a partir de las variables saque por pantalla la siguiente información:
La fecha de hoy es : 11 de feb de 2013
Soy el usuario : vicent
Me encuentro en el directorio : /home/vicent
2. **quienesta.sh**, que indique el número de usuarios conectados al sistema y la fecha actual
Hay conectados 5 usuariosjue 26 de Feb, 12:50
3. **busca.sh**, pide un nombre por el teclado y comprueba si existe en el fichero de nombres del sistema (passwd)
4. **esta.sh**, lee un login por el teclado y comprueba si el usuario está conectado al sistema en esos momentos, escribiendo el mensaje correspondiente
5. **guarda.sh**, copia todos los ficheros del directorio actual (mejora: en el que puedas elegir el usuario) en un subdirectorío llamado seguridad. Si el directorio no existe el script lo debe de crear.
6. Modificar **esta.sh** anterior para que compruebe si están conectado los usuarios pasados como parámetros
7. Corregir los errores del siguiente script.

```
for i in *
do
    TEST=who | grep $i`
    if [ ! "TEST" ]
    then
        echo "$i Si esta conectado"
    else
        echo "$i NO esta conectado"
    fi
```
8. Realizar un shell-script llamado **grafica.sh** que reciba como argumentos numeros comprendidos entre 1 y 75. Generará una línea por cada argumento con tantos asteriscos como indique su argumento, en caso de que algún argumento no esté dentro del rango sacará un mensaje indicándolo y finalizará el script.
9. Programar un script llamado **estadistica_dir.sh** que realice un estudio de todo el arbol de directorios y ficheros partiendo del directorio pasado como parámetro. Debemos obtener la siguiente información:
 - Número de ficheros con permiso de lectura para nuestro usuario
 - Número de ficheros con permiso de escritura ...
 - Número de ficheros con permiso de ejecución ...
 - Número de ficheros sin permiso de lectura ...
 - Número de ficheros sin permiso de escritura ...
 - Número de ficheros sin permiso de ejecución ...
10. Hacer un shell-script llamado **lista_fich.sh** que acepte como argumentos nombres de ficheros y muestre el contenido de cada uno de ellos precedido de una cabecera con el nombre del fichero
11. Hacer un shell-script llamado **donde_esta.sh** que busque la presencia del comando pasado como argumento en alguno de los directorios referenciados en la variable \$PATH, señalando su localización y una breve descripción del comando caso de existir su página man.
12. shell-script **media.sh** que calcule la media aritmética de todos los argumentos pasados como parámetros.
13. Calcular el mayor de dos números **mayor.sh** introducidos por el teclado
14. **cuentavar.sh**, imprime y cuenta las variables pasadas como parámetro al script.
15. Comparar (mayor/menor/igual) **compara.sh** dos números pasados como parámetro
16. Calculadora básica **calculadora.sh**
17. Realizar un shell-scrip **triangulo.sh** que admita como argumento un número 'N' y que obtenga a la salida una serie de 'N' filas de forma triangular.
Ejemplo : para ./piramide.sh 5 la salida sería.

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```
18. Mejorar triangulo.sh para que sea una pirámide **piramide.sh**
19. Programar en shell script **alquilar.sh** que muestre los vehículos que se pueden alquilar. El usuario introducirá por el teclado el vehículo que quiera alquilar y el programa le mostrará el precio del alquiler para ese vehículo
Tabla de precios de alquiler:
coche : 20 € / km
furgoneta : 10 € / km
bicicleta : 5 € / km

20. **mi_logrotate.sh**, tenemos en un directorio una serie de ficheros de log que van creciendo de forma ilimitada con el uso regular de ciertos programas. Realizar un shell-script que actúe sobre los ficheros con nombre tipo '*.log' del directorio actual de forma tal, que si alguno de ellos supera en tamaño las 50 líneas, dejará solo las últimas 30 líneas del fichero y las restantes serán guardadas en un directorio `old_rot` en formato comprimido.

En dicho directorio habrá que conservar en formato comprimido no solo la última porción eliminada del original, sino las cuatro últimas porciones eliminadas. Para ello será necesario ir rotando los nombres de estos ficheros comprimidos incorporando en los mismos un dígito de secuencia.

(parte eliminada) --> *.log.rot1.gz --> *.log.rot2.gz --> *.log.rot3.gz --> *.log.rot4.gz --> eliminado

El programa durante su ejecución irá mostrando los ficheros encontrados y señalará aquellos que por su tamaño sea necesario rotar.

21. Crear usuarios a partir de un fichero de texto

<https://es.stackoverflow.com/questions/85771/c%C3%B3mo-crear-usuarios-a-partir-de-fichero-de-texto-en-un-script-de-bash>