

Ejercicios de Bash

Funciones

1) Realizar una función que no de la información del archivo */proc/cpuinfo* obtener los datos de *vendor_id*, *model name*, *cpu MHz*, */proc/meminfo* que me muestre *MemTotal*, *MemFree*, del archivo y mediante comandos obtener la arquitectura y versión del kernel. Si tenemos mas de un procesador entonces que nos muestro solo una linea de *vendor_id*, *model name* y *cpu Mhz*.

Este ejemplo seria la salida :

```
ID del Vendedor   : AuthenticAMD
Nombre del Modelo : AMD Athlon(tm) II X2 250 Processor
CPU Mhz          : 3013.461
Memoria Total    : 4062236 kB
Memoria Libre    : 194928 kB
Arquitectura     : x86_64
Version del Kernel : 2.6.32-5-amd64
```

2) Realizar un script para hacer backup del directorio */etc*, llamando el archivo de salida *backup_etc.tar.bz2* y guardarlo en */tmp* y realizar el hash de la copia final.

Funciones y parámetros

3) Crear un programa que pasándole como parámetro el nombre del usuarios, obtener como se muestra la salida : *Nombre del usuario*, *ID*, *GID* y *shell*.

Este ejemplo seria la salida :

```
Nombre : root ( root )
ID      : 0
GID     : 0
Shell   : /bin/bash
```

4) Crear un script que por parámetros nos cree un usuario para eso le vamos a pasar como parámetro el *nombre del usuario*, *descripción* y *shell*. Luego imprimir el nombre del programa y cantidad de parámetros que se pasaron y todos los parámetros pasados.

Control de flujo, return y exit

5) Crear una función donde le pasamos un valor numérico y nos retorne si dicho valor es un número primo o no.

6) Realizar una función que verifique el parámetro que le pasamos si es un archivo, directorio, y mostrar la siguiente salida :

Este ejemplo seria la salida :

```
Dueño   : rwx
Grupo   : r--
Otros   : ---
Tipo    : Directorio
Nombre  : pepe
```

Control de flujo, return, exit, select y funciones

7) Generar un menú con las siguientes opciones, llamando el archivo **menu.sh**.

- Usuarios
- Filesystem
- Sistema
- Salir

Y muestre como prompt : **Elija algunas de las siguientes opciones :**

Usuarios

Agregar

Para la opción de **Usuario** crear un archivo llamado **usuarios.sh** que contenga un **submenu** :

- Agregar
- Borrar
- Informacion
- Salir

Por cada ítem (**Agregar**, **Borrar** y **Información**) crear sus respectivas funciones y pregunte si quiere continuar agregando, borrando o obteniendo información de un usuario, también **crear la función llamada verificar_usuario** que se le pasa como parámetro el nombre del usuario y esta función retorna **0** (verdadero si existe el usuario que no pida el shell) y **1** (falso si no existe el usuario), en shell que aparezca un **menú** con todos los **shell** que hay creando para este una función llamada **mostrar_shell** y retorno la opción elegida.

Dentro de la función agregar pedirá la siguiente información :

- Nombre del usuario.
- Shell

Una vez ingresado estos valores y al ejecutar el comando **useradd** con los parámetros ingresados verificar si fue correctamente el comando ejecutado. Una vez esto preguntar si desea agregar otro usuario, para esto crear una función llamada **continuar** que se le pase como parámetro el título (ej. Desea agregar otro usuario (S/N) ? :) y retorno **0** (si dice que desea continuar) y **1** (lo contrario).

Borrar

Crear una función llamada **borrar** que ingrese el usuario y utilizando la anterior función si existe o no el usuario, si existe que lo borre y sino que diga que no existe el usuario, también utilizar la función anteriormente creada llamada **continuar** pasándole como parámetro (Desea borrar otro usuario (S/N) ? :). Cuando se borra el usuario verificar si se ejecuto la sentencia bien o mal.

Información

Crear una función llamada **información** que pidiendo un usuario verifique si existes, si existe que me muestre la siguiente información:

- Nombre del usuario : xxxx
- ID usuario : xx
- ID grupo : xx
- Shell : /bin/xxx
- Directorio : /home/xxxx

Y si desea continuar mostrando otro usuario.

Filesystem

dentro del menú principal mostrar el del los filesystem montados el espacio libre, utilizado,etc (df -h).

Sistema

Crear el archivo llamado **sistema.sh** que contenga un menú de la siguiente forma :

- Memoria.
- CPU.
- Salir

Memoria mostrar información de la memoria (/proc/meminfo).

Cpu mostrar información del cpu (/proc/cpuinfo).

Resultados de los ejercicios de Bash

1)

#vi obtener_info.sh

```
obtener_info()
{
  echo -n "ID del Vendedor  : "
  cat /proc/cpuinfo | grep 'vendor_id' | cut -d ':' -f2 | uniq

  echo -n "Nombre del Modelo : "
  cat /proc/cpuinfo | grep "model name" | cut -d ':' -f2 | uniq

  echo -n "CPU Mhz          : "
  a
  cat /proc/cpuinfo | grep 'cpu MHz' | cut -d ':' -f2 | uniq

  echo -n "Memoria Total    : "
  cat /proc/meminfo | grep 'MemTotal' | cut -d ':' -f2

  echo -n "Memoria Libre     : "
  cat /proc/meminfo | grep 'MemFree' | cut -d ':' -f2

  echo -n "Arquitectura      : "
  arch

  echo -n "Version del Kernel : "
  uname -r
}

clear
obtener_info
```

2)

#vi backup.sh

```
backup()
{
  echo "Realizando backup /etc"
  tar cvfz /tmp/backup_etc.tar.bz2 /etc
  md5sum /tmp/backup_etc.tar.bz2
}

clear
backup.sh
```

3)

#vi parm01.sh

```
DESC=`grep $1 /etc/passwd | cut -d':' -f5`
NOMBRE=`grep $1 /etc/passwd | cut -d':' -f1`
echo "Nombre  : $NOMBRE ( $DESC )"

echo -n "ID      : "
grep $1 /etc/passwd | cut -d':' -f3

echo -n "GID    : "
grep $1 /etc/passwd | cut -d':' -f4
```

```
echo -n "Shell : "  
grep $1 /etc/passwd | cut -d':' -f7
```

```
# chmod +x parm01.sh  
#!/parm01.sh root
```

4)

```
#vi parm02.sh
```

```
useradd -m -s $3 -c "$2" $1
```

```
echo "Nombre del Programa : $0"  
echo "Cantidad de Parametros : $#"  
echo "Parametros : $*"
```

```
# chmod +x parm02.sh  
#!/parm02.sh juan "Ejemplo de juan" /bin/bash
```