

Programación del Shell

Ejercicios

Estructura de archivos.

Estructura del archivo /etc/passwd:

```
#nombre:contraseña:UID:GID:comentarios:dir_propio:shell
jperez:Xdio0IKLLP:1130:103:Juan Pérez:/home/jperez:/bin/bash
```

Estructura del archivo /etc/group:

```
#grupo:GID:lista_usuarios
docentes:103:victor,german,fernando
```

Estructura del archivo /etc/hosts:

```
#número_IP nombre_máquina alias
10.137.137.250    liebre.cdi.edu.uy liebre localhost mailhost
```

Nota.

En la construcción de los siguientes programas o 'scripts' se valorará la presentación interna y externa. En la presentación interna, importa la indentación, los comentarios, la claridad de los comandos, la significación de los nombres elegidos; todo esto debe permitir considerar al programa como 'autodocumentado'. En la presentación externa, importan las leyendas aclaratorias, ayudas, y avisos de ejecución que favorezcan al usuario ocasional. Se aconseja colocar los archivos de los programas en un subdirectorio bin del directorio propio del usuario, donde no deberá haber otra cosa que estos programas.

Para la resolución de estos scripts se aportarán unas pistas con puntos de teoría a consultar.

De los documentos “[bash-comandos.pdf](#)”, “[bash-scripts.pdf](#)”, “[comandos.html](#)” y “[Curso básico programación del Shell.pdf](#)” abreviados como “[bco](#)”, “[bsc](#)”, “[com](#)” y “[cur](#)” respectivamente.

1. Recibir un nombre de archivo como parámetro e indicar, imprimiendo todas las leyendas que correspondan, si el archivo es legible, modificable y ejecutable por el usuario.
([cur-5](#), [bsc-15](#), [bco-22](#))
2. Recibir varios nombres de archivo como parámetros, y para cada uno validar si el nombre corresponde a un archivo común existente, y si es así mostrarlo en pantalla paginando.
([cur-5-6](#), [bsc-15](#), [bco-22](#))
3. Recibir un nombre de directorio, validar existencia y condición de directorio y mostrar nombres de todos los directorios y subdirectorios bajo él, en formato de página largo 23.
([cur-5](#), [bsc-15](#), [bco-22](#), [bco-10](#), [com-”ls”-”grep”-”pr”-”more”](#))
4. Escribir un programa `seaejec` que reciba un nombre de archivo, verifique que existe y que es un archivo común, lo convierta en ejecutable para el dueño y el grupo y muestre el modo final.
([cur-5](#), [bsc-15](#), [bco-22](#), [com-”chmod”](#))
5. Escribir un programa `copiabin.sh` que mueva todos los programas del directorio actual (archivos ejecutables) hacia el subdirectorio `bin` del directorio propio del usuario, muestre los nombres de los que mueve e indique cuántos ha movido o que no ha movido ninguno. Si el

directorio bin no existe, deberá ser creado.

(cur-5-6-9, bsc-15, bco-22, com-"mkdir"- "cp")

6. Usando el archivo /etc/passwd escribir el programa `usuarios` que lista los nombres de login, el directorio propio del usuario y el intérprete invocado por defecto de todos los usuarios, ordenados alfabéticamente por nombre de login. Con leyendas explicativas adecuadas para cada columna.

(bco-10, com-"cut"- "sort"- "more")

7. Usando solamente el archivo /etc/group, escribir los siguientes programas:

a) 'grupo1': listar los nombres y números de grupo y la lista de usuarios de cada uno, ordenados por nombre. Con leyendas explicativas adecuadas para cada columna. Estableciendo una pausa al terminar. Limpiando la pantalla al iniciar.

b) 'grupo2': igual, ordenados por número de grupo. Con leyendas explicativas adecuadas para cada columna. Estableciendo una pausa al terminar.

c) 'grupo3': reúne las dos salidas anteriores, así como nombre de la máquina y fecha del día.

(bco-10, bsc-10, com-"cat"- "cut"- "sort"- "more"- "hostname"- "date")

8. Escribir un programa `usugrup` que dado un nombre de login de usuario determine si existe en el sistema, y si es así, presente su nombre de usuario, , número de usuario (UID), grupo primario y grupos secundarios si los tiene, con leyendas adecuadas.

(cur-5, bco-10-24, com-"id"- "cut"- "tr"- "wc")

9. Escribir un programa `grupusu` que dado un nombre de grupo determine si existe en el sistema, y si es así, presente su nombre, número de grupo (GID), y lista de usuarios que pertenezcan a él, ya sea como grupo primario (en /etc/passwd) o como grupo secundario (lista en /etc/group).

(cur-5, bco-10-24, com-"cat"- "cut"- "tr"- "grep")

10. Escribir los siguientes programas:

a) `ligass`: muestra los nombres de archivo que son enlaces simbólicos.

b) `ligash`: muestra los archivos que tiene enlaces hard.

Ambos programas reciben un nombre como parámetro, y validarán que corresponda a un directorio del sistema.

(cur-5-6, bco-10-24, com-"cat"- "cut"- "sed"- "grep")

11. Escribir un programa `saludo` que, según la hora, escriba el saludo correspondiente al nombre de pila del usuario. En el archivo /etc/passwd los usuarios deben estar ingresados con nombre y apellido separados por blanco. Los saludos corresponden a las siguientes horas: Buenos días, de 05:00 hasta 12:59; Buenas tardes, de 13:00 hasta 19:59; Buenas noches 20:00 hasta 04:59. Ejemplo de mensaje: Buenos días, Juan.

A efectos de pruebas, se recibirán la hora y el nombre de login como parámetros, dejando comentados los comandos donde se extrae la hora real y se toma el usuario real.

(cur-5-9-10, bco-10, com-"cat"- "cut"- "date"- "grep")

12. Un script de respaldo produce, entre otros mensajes, líneas del tipo

"Total bytes written 18804023"

Guarda su salida en el archivo `respaldo.error`. Escribir un script `total.cinta` que sume los bytes grabados e indique el total en bytes, Mb y Gb.

Crear un archivo `respaldo.error` de prueba, con un contenido tal como

Total bytes written 1800

Total bytes written 1000

(cur-6-9)