Programación de shell-scripts: Práctica 3

NOTA IMPORTANTE PARA TODOS LOS EJERCICIOS: En todos los ejercicios deberá comprobarse que el número de parámetros es correcto, tal y como se indicó en la práctica anterior (practica 2) y con los mismos mensajes de error. Además, en los casos que proceda, deberán comprobarse otros posibles errores con sus respectivos mensajes de error (por ejemplo: comprobación de que el argumento es un fichero y/o un directorio).

1. Implementa un Shell-script llamado **ListadoFicherosNLineas** que muestre a la salida las rutas de los ficheros de texto que SUPEREN el número de líneas indicado por el usuario. Se iniciará la búsqueda en un directorio también indicado por el usuario y de forma recursiva (comprobando también los subdirectorios).

   La sintaxis de ejecución de este Shell-script es:

   ListadoFicherosNLineas <directorio_inicial_busqueda> <numero_lineas_umbral>

   La salida del Shell-script deberá ser exactamente como la siguiente:

   Los ficheros de texto que tienen más de xxxx líneas son:
   <ruta_fichero1>
   <ruta_fichero2>
   ...

   NOTA IMPORTANTE: El ejercicio deberá resolverse utilizando el recorrido de directorios mediante la sentencia ***for*** y no se podrá utilizar ningún comando u opción extra que resuelva el ejercicio de forma automática.

2. Hacer un Shell-script **CambiarShell** que modifique el intérprete de comandos del usuario indicado. El Shell-script recibirá la información del fichero del sistema *passwd* y dará como resultado la modificación de dicho fichero.

   La sintaxis de ejecución de este Shell-script es:

   CambiarShell <usuario> <nuevo_shell>

   NOTA IMPORTANTE: El ejercicio deberá resolverse utilizando la sentencia ***while*** combinado con la sentencia ***read***. Para tener un fichero ***passwd*** de ejemplo cópiese el fichero ***/etc/passwd*** al directorio donde se vaya a realizar el Shell-script.

3. Realizar un Shell-script llamado **ComandosFiltrar** que, a partir de la salida generada por el comando **ps uaxw**, muestre los comandos (último campo al completo) de los procesos que estén ocupando un espacio de memoria (quinto campo) que esté dentro del intervalo indicado por parámetro. El resto de información del comando **ps** no debe mostrarse.

   La sintaxis de ejecución de este Shell-script es:

   ComandosFiltrar <limite_tamaño_inferior> <limite_tamaño_superior>

   La salida del Shell-script deberá ser exactamente como la siguiente:

   Los comandos que están consumiendo entre ¿??? y ¿???? bytes de memoria son:
   Comando1
   Comando2
   ...

   NOTA IMPORTANTE: El ejercicio deberá resolverse utilizando la sentencia ***while*** combinado con la sentencia ***read***. **Se recomienda usar el comando *tr -s "<carácter>"* para transformar la salida del *ps* a formato CSV**.

4. Hacer un Shell-script **PrimerUIDLibre** que, usando el comando **getent passwd**, indique a la salida el UID libre más bajo disponible en el sistema (que no esté ya siendo usado por algún usuario), a partir del UID base indicado por parámetro (éste incluido). La salida deberá ser MUY escueta y mostrar solo el UID que se encuentre sin ningún texto adicional. La sintaxis de la orden sería:

   PrimerUIDLibre <UID_base>

   NOTA IMPORTANTE: El ejercicio deberá resolverse utilizando la sentencia ***while*** combinado con la sentencia ***read***. También será necesario incrementar una variable de forma numérica para lo que se recomienda utilizar el comando ***expr <expresión_entera>*** que, dada una expresión entera, vuelca a la salida su resultado. Por ejemplo:

   cont=0
   cont=$(expr $cont + 1)

5. En el directorio ***/home/asignaturas/so/shell-scripts*** encontrarás un fichero CSV (***ventas.csv***) con el histórico de ventas de una tienda de frutas. En la primera línea de este fichero podrás ver el significado de cada campo. Implementa un Shell-script llamado **VentasFruta** que, dada una fruta como argumento, muestre a la salida los días, EN ORDEN CRECIENTE, que se vendió dicha fruta junto con la cantidad vendida dicho día. La ruta al fichero CSV también se pasará como parámetro (no puede ponerse directamente **ventas.csv** en el código del Shell-script, tiene que funcionar con cualquier fichero CSV que se indique). La sintaxis de ejecución de este Shell-script es:

   VentasFruta <fichero_CSV> <fruta>

   La salida del Shell-script deberá ser exactamente como la siguiente:

   La fruta ¿??? se vendió los siguientes días:
   El día ¿???? se vendieron ¿???? Kg.
   El día ¿???? Se vendieron ¿???? Kg.
   …

   NOTA IMPORTANTE: El ejercicio deberá resolverse utilizando la sentencia ***while*** combinado con la sentencia ***read***.

6. Implementa un Shell-script llamado **CrearUsuarios** que, dado un fichero CSV como parámetro y un UID base a partir de cual asignar UIDs, cree usuarios nuevos en el sistema (añada líneas al fichero ***passwd*** del sistema con el formato exigido por dicho fichero), a partir de la información contenida en el fichero CSV. La sintaxis de ejecución de este Shell-script es:

   CrearUsuarios <fichero_CSV> <UID_base>

   El fichero CSV tendrá los siguientes campos separados por ; (punto y coma)
   Correo;Apellidos;Nombre;GID

   El login (nombre) de los usuarios será el usuario del correo (todo lo que hay antes de la @)
   El directorio personal de los nuevos usuarios estará bajo /home y tendrá el mismo nombre que el usuario.
   Al primer usuario creado se le asignará el UID indicado en el segundo parámetro (UID_base) y al resto UIDs consecutivos.
   Como campo comentario se indicará los <apellidos>, <nombre> del usuario.
   El intérprete asignado a los usuarios será /bin/bash.

   NOTA IMPORTANTE: El ejercicio deberá resolverse utilizando la sentencia ***while*** combinado con la sentencia ***read***.

   No será necesario validar que los usuarios ya existan o que el UID y el GID sean válidos.

   Para tener un fichero CSV de ejemplo cópiese el fichero ***/home/asignaturas/so/shell-scripts/usuarios.csv*** al directorio donde se vaya a realizar el Shell-script.

   Para tener un fichero ***passwd*** de ejemplo cópiese el fichero ***/etc/passwd*** al directorio donde se vaya a realizar el Shell-script