

PRACTICAS DE SISTEMA OPERATIVO UNIX

PRÁCTICAS DE UNIX

PRÁCTICA 8. Programación shell general

Objetivos

Realizar supuestos de programación de guiones shell utilizando todas las herramientas tratadas hasta la fecha.

Herramientas

Las herramientas a utilizar serán las vistas y, además, df

DESARROLLO DE LA PRÁCTICA

1. Crear un guión shell llamado "*renom*" que permita renombrar ficheros. Se debe comprobar que el primer argumento sea un fichero y que el segundo no exista, visualizándose mensajes de error oportunos.

```
if [ -f $1 ]
then
  if [ ! -a $2 ]
  then
    mv $1 $2
  else
    echo "Error, el fichero $2 ya existía anteriormente."
  fi
else
  echo "Error, $1 no es un fichero."
fi
```

2. Hacer un guión llamado "*borrar*" que admita un número indeterminado de parámetros y realice lo siguiente:

- Si el parámetro especificado es un fichero ordinario y ocupa más de 50 bytes se borrará el fichero.
- Si es un directorio solicitará confirmación antes de proceder a borrarlo. Se debe tratar también el caso de que el directorio tenga ficheros lo cual se notificará antes de la solicitud de confirmación.
- En cualquier otro caso se visualizará: "No se procesa el argumento: <argumento>" Sólo se procesará el directorio actual.

```
for i in $*
do
  if [ -f $i ]
  then
    if [ -s $i > 50 ]
    then
      rm $i
    fi
  else
    if [ -d $i ]
    then
      if (find -name $i/*)
      then
        echo "$i es un directorio que contiene archivos. ¿Eliminar? (S/N)"
      else
        echo "$i es un directorio vacío. ¿Eliminar? (S/N)"
      fi
    fi
    read opcion
    if $opcion -eq "S"
    then
      rmdir $i
    fi
  fi
done
```

```
        fi
    else
        echo "$i no se procesa."
    fi
fi
fi
done
```

3. Generar un guión shell para manejar un fichero "agenda" que estará formado por los campos nombre, apellidos y teléfono, utilizando como separador de campos el carácter dos puntos (:). El shell debe presentar el siguiente menú y realizar los tratamientos oportunos con el fichero "agenda":

```
MENÚ DE OPCIONES
=====
[A]ltas
[B]ajas
[C]onsultas
[S]alir
Elija opción (A,B,C,S):
```

En caso de teclear otra opción se notificará del error y se volverá a presentar el menú.

```
function menu
{
    clear
    echo MENU DE OPCIONES
    echo =====
    echo      [A]ltas
    echo      [B]ajas
    echo      [C]onsultas
    echo      [S]alir
    echo Opción:
    read opcion
    case $opcion in
        A) alta;;
        B) baja;;
        C) consulta;;
    esac
}
function alta
{
    clear
    echo "Nuevo telefono"
    echo "=====\n\n"
    echo "Nombre: \c"
    read nombre
    echo "Apellido: \c"
    read apellido
    echo "Telefono: \c"
    read telefono
    echo "$nombre:$apellido:$telefono" >> agenda
}
function baja
{
    clear
    echo "Baja telefono"
    echo "=====\n\n"
    echo "Apellido: \c"
    read apellido
    for n in cat agenda
    do
        ape=echo $n | cut -f2 -d":"
        if ! $ape=$apellido
```

```
        then
            echo $n >> temp
        fi
        rm agenda
        mv temp agenda
    done
}
function consulta
{
    clear
    echo "Buscar telefono"
    echo "=====\n\n"
    echo "Apellido: \c"
    read apellido
    echo "`grep $apellido agenda | cut f1,2 -d":"` --> `grep $apellido
agenda | cut f3 -d":"`"
}
opcion=
until opcion=S
do
    menu
done
```

4. Guión shell que elimine los procesos de su propiedad que lleven ejecutándose más de cinco minutos, a excepción del proceso de conexión, el proceso del shell inicial y el correspondiente a este guión shell.

```
for i in ps
do
    proceso=echo $i | tr -s" " | cut -f2 -d" "
    if [ `time $proceso` -gt 300 ]
    then
        if [ ! `echo $i | tr -s" " | cut -f1 -d" "` = 0 ]
        then
            kill -9 $i
        fi
    fi
done
```

5. Guión shell que genere un listado con información de los grupos dados de alta en el sistema. El listado tendrá el siguiente formato.

```
NOMBRE DEL GRUPO:      xxxxxxxxx
    nombre de usuario
    nombre de usuario
. . .
NOMBRE DEL GRUPO:      xxxxxxxxx
. . .
```

```
for i in cat /etc/group
do
    grupo=`echo $i | cut -f1 -d","`
    usuarios=`echo $i | cut -f4 -d","`
    echo "NOMBRE DEL GRUPO:      $grupo\n"
    for j in $usuarios
    do
        echo "          $j\n"
    done
done
```

6. Se tiene en un fichero la información sobre los jugadores de fútbol que han conseguido goles en cada jornada durante la liga. El formato del fichero es el siguiente:
jornada:nombre del jugador:equipo del jugador:n_goles:equipo goleado

Crear un guión shell que genere un listado en el que aparezca el número de goles que ha conseguido cada equipo en cada una de las jornadas.

```
for equipo in `cat goleadores | cut -f3 -d":"`
do
  goles=0
  for i in `grep $equipo goleadores`
  do
    if [ echo $i | cut -f3 -d":" -eq $equipo ]
    then
      (( $goles=$goles + `echo $equipo | cut -f4 -d":"` ))
    fi
  done
  echo "$equipo ---> $goles"
done
```

7. Teniendo en cuenta un fichero llamado "alumnos" con una estructura como la indicada al final del presente ejercicio, escriba un guión shell que genere un fichero "medias" con las notas medias de cada curso. (El fichero puede estar desordenado y no se conocen a priori cuantos ni cuales son los cursos de alumnos.)

N_EXPEDIENTE:CURSO:ASIGNATURA:NOTA

```
for curso in `cat alumnos | cut -f2 -d":"`
do
  nota_sum=0
  n=0
  for i in `grep $curso alumnos`
  do
    if [ echo $i | cut -f2 -d":" -eq $curso ]
    then
      (( $nota_sum=$nota_sum + `echo $nota_sum | cut -f4 -d":"` ))
      (( $n = $n + 1 ))
    fi
  done
  (( $media= $nota_sum / $n ))
  echo "$curso ---> $media "
done
```

8. Supuesto que el nombre de los usuarios del sistema comienza por "p", "r" o "f", obtenga el número de usuarios dados de alta. Visualice también un listado con el nombre de usuario y su directorio de conexión.

```
num=0
for i in `grep /etc/passwd "[p,r,f]$" `
do
  echo $i | cut -f1,5 -d":"
  (( $num=$num+1 ))
done
echo "\nTOTAL: $num"
```

9. Crear un guión que liste el nombre de los usuarios que no tienen fichero ".profile" y el número de ellos, supuesto que el nombre de los usuarios comienza por "p", "r" o "f", y se tiene permiso de lectura en sus directorios de conexión.

```
for usuario in `grep /etc/passwd "[p,r,f]$" | cut -f5 -d":"`
do
  if [ ! -f $usuario/.profile ]
  then
    echo "`grep $usuario /etc/passwd | cut -f1 -d":"` no tiene
                                                .profile."
  fi
done
```

10. Sin utilizar el comando "sleep", crear un guión shell que admita un número del 1 a 60 como parámetro posicional. Este número será el número de segundos que tardará en aparecer en video inverso el mensaje "TIEMPO CONSUMIDO".
Indicar la llamada al guión shell para no necesitar esperar a que termine su ejecución para tener el control del terminal.

```
{ guion_ej10 nn }&
```

11. (1) Crear un guión shell que nos permita contar el total de bloques ocupados por los siguientes tipos de ficheros: a) ficheros de bloques, b) ficheros especiales de caracteres, c) directorios, d) tuberías, e) ficheros ordinarios, f) enlaces.
(2) Generar un guión shell que nos permita contar los ficheros de unos directorios pasados como argumentos, mostrando por la salida estándar el nombre del directorio y el total de ficheros del directorio. La totalización de ficheros deberá ser recursiva en cada caso.
(3) Programar un guión shell que simule el comportamiento del comando df, produciendo una salida en megabytes del tipo siguiente:

```
Nombre          Espacio          Espacio          Porcentaje
sist.ficheros   ocupado          total            ocupacion
-----
...             ...             ...             ...
...             ...             ...             ...

Total espacio disco  -----          -----          -----

tot_tam=0
tot_ocu=0
echo " Nombre          Espacio          Espacio          Porcentaje"
echo " fichero         ocupado          total            ocupacion"
echo
"-----"
for f in ls -R -l /*
do
    tamaño=echo $f | tr -s" " | cut -f -d" "
    ocupa= echo $f | tr -s" " | cut -f -d" "
    ((tot_tam=tot_tam + tamaño))
    ((tot_ocu= tot_ocu + tamaño))
    ((porcen=tot_ocu))
done
```

- (4) Crear un guión shell que permita realizar las siguientes operaciones sobre un fichero de texto tipo base de datos, que contengan los campos abajo indicados:
(a) Actualizaciones (altas, bajas, modificaciones) y consultas, dando a elegir al usuario la clave de búsqueda.
(b) Clasificación por cualquier campo.

Estructura de la Base de Datos:

NIF(9) :NOMBRE(10) :APELLIDOS(20) :DIRECCIÓN(20) : COD_POST(5) :TELEF(9)

```
function menu
{
    clear
    echo "1- Altas\n2-Bajas\n3-Modificaciones\n4-Clasificación\nX-
                                           Salir\n"

    read opcion
    clear
    case opcion in
        1) altas;;
        2) bajas;;
        3) modificaciones;;
        4) clasificacion;;
    esac
}
function altas
{
```

```
    echo "NIF: \c"
    read nif
    echo "Nombre: \c"
    read nombre
    echo "Apellidos: \c"
    read apellidos
    echo "Dirección: \c"
    read direccion
    echo "C.P.: \c"
    read cp
    echo "Tlf: \c"
    read telefono
    echo "$nif:$nombre:$apellidos:$direccion:$cp:$telefono" >> agenda
}
function bajas
{
    echo "Introduzca Clave para Baja (1-NIF, 2-Nombre, 3-Apellidos, 4-
        Direccion, 5-CP, 6-Tlf): \c"

    read clave
    echo "Valor de la Clave de Baja: \c"
    read valor
    for unidad in cat agenda | cut -f$clave -d:
    do
        if [ ! $unidad -eq $valor ]
        then
            echo grep agenda $unidad >> temp
        fi
    done
    rm agenda
    mv temp agenda
}
function modificaciones
{
    echo "Introduzca Clave Modificación (1-NIF, 2-Nombre, 3-Apellidos, 4-
        Direccion, 5-CP, 6-Tlf): \c"

    read clave
    echo "Valor de la Clave a Modificar: \c"
    read valor
    mv agenda temp
    for unidad in cat temp | cut -f$clave -d:
    do
        if [ $unidad -eq $valor ]
        then
            echo grep temp $unidad
            altas;;
        else
            echo grep temp $unidad >> agenda
        fi
    done
    rm temp
}
function clasificacion
{
    echo "Introduzca Clave Clasificación (1-NIF, 2-Nombre, 3-Apellidos,
        4-Direccion, 5-CP, 6-Tlf): \c"

    read clave
    sort -t: +$clave agenda agenda
}
until [ opcion -eq X ]
do
    menu
done
```