



Preparador Informática

www.preparadorinformatica.com

PRÁCTICA 15 SHELL SCRIPTS

OPOSICIONES SAI 2010
(Supuesto 3)
ANDALUCÍA

EJERCICIO

ANDALUCÍA SAI 2010 (SUPUESTO 3)

El ejercicio práctico se realizará bajo sistema operativo LINUX. Se podrá hacer uso del comando `man`.

Realizar un script llamado **informe.sh** que genere un informe de uso del sistema de ficheros por parte de los usuarios.

La sintaxis del script será:

informe.sh [-u usuario | -c]

-u usuario	recopilará información del sistema para el usuario dado
-c	recopilará información del sistema para todos los usuarios conectados
sin parámetro	recopilará información del sistema para todos los usuarios del sistema

El contenido del informe será el siguiente:

#####

Usuario: xxxxxxx

Nº Ficheros de los que es propietario: 28

Nº Ficheros que puede modificar: 34

Nº Ficheros abiertos: 12

Fichero más antiguo del usuario: /home...

Fichero más recientemente modificado: /home/. ..

Tamaño fichero más pequeño: 8

Tamaño medio de fichero: 4300

Tamaño fichero más grande: 150648

Tipo de fichero más usual del usuario: ASCII C program text

#####

AYUDA: En el ejercicio hay que utilizar una función que no hemos visto en los manuales para averiguar el número de ficheros abiertos. En el manual 2 vimos como buscar con la ayuda de *man* comandos que no sabemos su nombre exacto, pero sabemos alguna palabra clave (en inglés) sobre lo que hace. Ejemplo: `man -k "texto que busco"`



SOLUCIÓN PROPUESTA

```
#!/bin/bash

#####INICIO DE DECLARACIÓN DE FUNCIONES#####

#Función que recibe como argumento un nombre de usuario y devuelve
#el número de ficheros de los que es propietario
function num_ficheros_propietario()
{
    usuario=$1
    num_ficheros=$(find /home/$usuario -user $usuario -type f 2> /dev/null | wc -l)
    echo "Nº Ficheros de los que es propietario: " $num_ficheros
}

function num_ficheros_modificables()
{
    usuario=$1
    num_ficheros_mod=$(find /home/$usuario -type f 2> /dev/null -exec sudo -u
$usuario test -w {} \; -print | wc -l)
    echo "Nº ficheros que puede modificar: " $num_ficheros_mod
}

function num_ficheros_abiertos()
{
    usuario=$1
    num_abiertos=$(lsof -u $usuario 2> /dev/null | sort | cut -d' ' -f1 | uniq | wc -l)
    echo "Nº Ficheros abiertos: " $num_abiertos
}

function fichero_mas_antiguo()
{
    usuario=$1
    fich_mas_antiguo=$(find /home/$usuario -user $usuario -type f -printf "%T+ %f\n"
2> /dev/null | sort -n | head -1 | cut -d' ' -f2)
    echo "Fichero más antiguo del usuario: " $fich_mas_antiguo
}

function fichero_mas_reciente_modificado()
{
    usuario=$1
    fichero=$(find /home/$usuario -user $usuario -type f -exec stat -c "%y %n" {} 2>
/dev/null \; | sort -n | tail -1 | cut -d' ' -f4)
    echo "Fichero más recientemente modificado: " $fichero
}

function tam_fichero_mas_pequeno()
{
    usuario=$1
    tam_mas_peq=$(find /home/$usuario -user $usuario -type f -exec stat -t {} \; 2>
/dev/null | cut -d' ' -f1,2 | sort -n -t' ' -k2 | head -1 | cut -d' ' -f2)
    tam_mas_peq2=$(find /home/$usuario -user $usuario -type f -exec stat -c %s {} \; 2>
/dev/null | sort -n | head -1)
    echo "Tamaño fichero más pequeño: " $tam_mas_peq
}

```



```

function tam_fichero_medio()
{
    usuario=$1
    total=0
    num_ficheros=$(find /home/$usuario -user $usuario -type f -printf "%s \n" 2>
/dev/null | wc -l)
    for tam in $(find /home/$usuario -user $usuario -type f -printf "%s \n" 2> /dev/null)
    do
        total=`expr $total + $tam`
    done
    echo "Tamaño medio de fichero: "`expr $total / $num_ficheros`
}

function tam_fichero_mas_grande()
{
    usuario=$1
    tam_mas_grande=$(find /home/$usuario -user $usuario -type f -exec stat -t {} \; 2>
/dev/null | cut -d' ' -f1,2 | sort -n -t' ' -k2 | tail -1 | cut -d' ' -f2)
    tam_mas_grande2=$(find /home/$usuario -user $usuario -type f -exec stat -c %s {} \;
2> /dev/null | sort -n | tail -1)
    echo "Tamaño fichero más grande: "$tam_mas_grande
}

function fichero_mas_comun()
{
    usuario=$1
    fich_mas_comun=$(find /home/$usuario -user $usuario -type f -exec file -b {} 2>
/dev/null \; | sort | uniq -c | head -1 | cut -c9-)
    echo "Tipo de fichero más usual del usuario: "$fich_mas_comun
}

function informe()
{
    USUARIO=$1
    echo "#####"
    echo "Usuario: "$USUARIO
    num_ficheros_propietario $USUARIO
    num_ficheros_modificables $USUARIO
    num_ficheros_abiertos $USUARIO
    fichero_mas_antiguo $USUARIO
    fichero_mas_reciente_modificado $USUARIO
    tam_fichero_mas_pequeno $USUARIO
    tam_fichero_medio $USUARIO
    tam_fichero_mas_grande $USUARIO
    fichero_mas_comun $USUARIO
    echo "#####"
}

#####FIN DE DECLARACIÓN DE FUNCIONES#####

```



```
#####INICIO PROGRAMA PRINCIPAL#####
```

```
if [ $UID -ne 0 ]
then
    echo "ERROR: Este script debe ejecutarlo el root"
    exit 1
fi

#Recopila información del sistema para el usuario dado
if [ $# -eq 2 ] && [ "$1" = "-u" ] && [ $(id -u $2) 2> /dev/null -ge 1000 ]
then
    USUARIO=$2
    informe $USUARIO

#Recopila información del sistema para todos los usuarios conectados
elif [ $# -eq 1 ] && [ "$1" = "-c" ]
then
    for USUARIO in $(who | cut -d' ' -f1 | uniq)
    do
        informe $USUARIO
    done

#Recopila información del sistema para todos los usuarios del sistema
elif [ $# -eq 0 ]
then
    for USUARIO in $(cat /etc/passwd | cut -d":" -f1 )
    do
        if [ $(id -u $USUARIO) -ge 1000 ]
        then
            informe $USUARIO
        fi
    done
else
    echo "ERROR en parámetros"
    echo "USO: $0 [-u usuario | -c]"
    exit 2
fi
```

