

PRÁCTICA Nº9

OPOSICIONES CUERPOS DOCENTES 2010 SUPUESTO 3

PLANTEAMIENTO DEL EJERCICIO PRÁCTICO (Duración 2,5 h):

Este supuesto se realizará bajo el sistema operativo LINUX (o equivalente) y se podrá hacer uso del comando **man** para obtener ayuda.

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: 25

Nº Ficheros que puede modificar: 30

Nº Ficheros abiertos: 58

Fichero más antiguo del usuario: /home...

Fichero más recientemente modificado: /home/. ..

Tamaño fichero más pequeño: 8

Tamaño medio de fichero: 5487

Tamaño fichero más grande: 254568

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

BASH – Práctica 9. Supuesto 3 - Examen 2010 Solución propuesta

```
#!/bin/bash
#Compruebo si el usuario es root
if [ $UID -ne 0 ]
then
    echo "Este script solo lo puede ejecutar el root"
    exit 1
fi

#Función que recibe como argumento un nombre de usuario, y devuelve el número de
#ficheros de los que es propietario
function numFicheros_en_propiedad ()
{
    usuario=$1
    numero_ficheros=$(find /home -user $usuario -type f 2> /dev/null | wc -l)
    echo "N Ficheros de los que es propietario:" $numero_ficheros
}

#Función que pasándole como argumento el nombre de un usuario, nos devuelve el número de fichero que
#puede modificar, es decir, aquellos para los que tenga permiso de escritura (w)
function numFicheros_modificables ()
{
    usuario=$1
    $numero_ficheros_modificables=$(find /home -exec sudo -u $usuario test -w {} \; -print | wc -l)
    echo "N ficheros que puede modificar:" $numero_ficheros_modificables
}

# Función recibe por parámetro el nombre de un usuario, y devuelve el número
# de ficheros que tiene abiertos
function numFicheros_abiertos()
{
    usuario=$1
    numFicheros=$(lsof -u "$usuario" 2> /dev/null | uniq | wc -l)
    echo "N Ficheros abiertos:" $numFicheros
}

# Función recibe por parámetro el nombre de un usuario, y devuelve el nombre del fichero de su
# propiedad, que se creó en primer lugar en el sistema
function fichero_mas_antiguo()
{
    usuario=$1
    f_mas_antiguo=$(find /home -user $usuario -type f -exec ls --full-time {} 2>/dev/null \; | sort -k6 |
head -1 | cut -d" " -f9)
    echo "Fichero m antiguo del usuario:" $f_mas_antiguo
}

# Función recibe el nombre de un usuario del sistema, y me devuelve el nombre del fichero, que ha sido
# más recientemente modificado
function fichero_modificado_mas_reciente()
{
    usuario=$1
    f_modificado_mas_reciente=$(find /home -user $usuario -type f -exec ls -lc {} 2>/dev/null \; |
sort -k6 | tail -1 |cut -d" " -f8)
    echo "Fichero m recientemente modificado:" $f_modificado_mas_reciente
}

# Función recibe el nombre de un usuario y busca el fichero de menor tamaño, el tamaño medio y
#el fichero de mayor tamaño y los añade al informe
function ficheros_por_tamano()
{
    usuario=$1
    tamano_mas_pequeno=$(find /home -user $usuario -type f -exec stat -c %s {} 2> /dev/null \; |sort -n |
head -1)
    tamano_mas_grande=$(find /home -user $usuario -type f -exec stat -c %s {} 2> /dev/null \; | sort -n |
tail -1)
    numero_ficheros=$(find /home -user $usuario -type f 2> /dev/null | wc -l)
    tamano_total=0
    for tam in $(find /home -user $usuario -type f -exec stat -c %s {} 2> /dev/null \; | sort -n)
    do
        tamano_total=`expr $tamano_total + $tam`
    done
    tamano_medio=`expr $tamano_total / $numero_ficheros`
    echo "Tamaño del fichero más pequeño" $tamano_mas_pequeno
    echo "Tamaño medio de fichero:" $tamano_medio
    echo "Tamaño del fichero más grande:" $tamano_mas_grande
}
```

```

}
# Función recibe un nombre de usuario, y me devuelve el tipo de archivo más usual que tiene, para ello,
# buscamos los tipos de los ficheros del usuario, buscamos el número de ocurrencias de cada tipo, y nos
# quedamos con el de mayor número de ocurrencias.
function tipo_fichero_mas_usual()
{
    usuario=$1
    tipo_f_mas_usual=$(find /home -user $usuario -type f -exec file -b {} 2> /dev/null \; |cut -d"," -f1 |
        sort | uniq -c | sort -n | tail -1 | tr -d 0-9" ")
    echo "Tipo más usual del usuario:" $tipo_f_mas_usual
}

# Función recibe un nombre de usuario, y añade al fichero informe.dat el informe de sus ficheros
function crea_informe()
{
    usuario=$1
    echo "#####"
    echo "Usuario: $usuario"
    numFicheros_en_propiedad $usuario
    numFicheros_modificables $usuario
    numFicheros_abiertos $usuario
    fichero_mas_antiguo $usuario
    fichero_modificado_mas_reciente $usuario
    ficheros_por_tamano $usuario
    tipo_fichero_mas_usual $usuario
}

if (( $# == 2 )) && [ "$1" = "-u" ] && [ $(grep ^$2: /etc/passwd | cut -d: -f3) -ge 1000 ]
then
    usuario=$2
    crea_informe $usuario
else
    if [ $# -eq 1 -a "$1" = "-c" ]
    then
        usuarios=$(who | cut -d" " -f1 | uniq)
        for usuario in $usuarios
        do
            crea_informe $usuario
        done
    else
        if [ $# -eq 0 ]
        then
            for usuario in $(cat /etc/passwd | cut -d: -f1)
            do
                if [ $(id -u $usuario) -ge 1000 ]
                then
                    crea_informe $usuario
                fi
            done
        else
            echo "Error en los parámetros"
            exit 1
        fi
    fi
fi

```