

PRÁCTICA 11 **SOLUCIÓN PROPUESTA**

OPOSICIONES CUERPOS DOCENTES 2006

EJERCICIO PRÁCTICO - PLANTEAMIENTO

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

Elabora un script para automatizar las tareas de mantenimiento de usuarios en un servidor Linux, en el que realizan prácticas de scripts diversos grupos de alumnos.

Recibirá como parámetro el curso actual y admitirá además las siguientes opciones:

- f Creación del fichero de grupos del curso actual. Tendrá el siguiente formato:
claveGrupo:numeroAlumnos
- c Creación de cuentas de usuarios y grupos. El login reflejará grupo al que pertenece, curso académico y número de alumno. Además los directorios home estarán organizados por grupos, garantizando al máximo la privacidad.
- b Eliminación de cuentas, grupos y directorios

```
#!/bin/bash

if [ $UID -ne 0 ]
then
    echo "ERROR, este script solo lo puede ejecutar el root"
    exit 1
fi

#Función que recibe como parámetro el nombre del curso que se quiere crear, y crea el
#fichero con las claves de cada grupo de clase, y su número de alumnos
function crear_curso()
{
    local curso=$1
    echo "Introduce el número de grupos"
    read numeroGrupos
    for ((i=0;i<$numeroGrupos;i++))
    do
        echo "Clave del grupo"
        read claveGrupo
        echo "Número de alumnos"
        read numeroAlumnos
        echo "$claveGrupo:$numeroAlumnos" >> $curso
    done
}

#Función que, previa consulta al fichero del curso pasado por parámetro, realiza la
#creación de las cuentas de usuario y de grupos. Creará tantas cuentas de usuario, como
#número de alumnos tenga cada grupo. Los directorios de usuario están organizados
function crear_cuentas ()
{
    local curso=$1
    if [ ! -f $curso ]
    then
        echo "No se ha creado el fichero de curso para $curso"
        exit 1
    fi

    for grupo_alumnos in $(cat $curso)
    do
        grupo=$(echo $grupo_alumnos | cut -d":" -f1)
        numAlumnos=$(echo $grupo_alumnos | cut -d":" -f2)
        # Creamos el grupo para el grupo de clase
        groupadd $grupo 2>/dev/null
        for ((i=1;i<=$numAlumnos;i++))
        do
            #creamos los directorios de trabajo organizados por grupos
            mkdir -p /home/$curso/$grupo/$grupo$curso$i      #/home/2017-18/SMR1/SMR12017-181
            #creamos al usuario, asignándolo al grupo curso, y dando el como
            #nombre, el grupo y un número
            useradd -b /home/$curso/$grupo/$grupo$curso$i -g $grupo $grupo$curso$i
                                                                2>/dev/null

            #Hacemos propietario del directorio al usuario y al grupo
            chown $grupo$curso$i /home/$curso/$grupo/$grupo$curso$i
            chgrp $grupo /home/$curso/$grupo/$grupo$curso$i
            #Cambiamos el modo para garantizar la privacidad del directorio.
            #Solo el alumno propietario del directorio puede leer, escribir y
            #ejecutar en él.
            chmod 700 /home/$curso/$grupo/$grupo$curso$i
        done
    done
}
```

#Esta función recibe como argumento el identificador de un curso académico, y realiza
 #la eliminación de todas las cuentas de usuario, grupos y directorios, creados para ese
 #curso.

```
function eliminar_cuentas ()
{
    local curso=$1
    if [ ! -f $curso ]
    then
        echo "No se ha creado el fichero de curso para $curso"
        exit 1
    fi

    for grupo_alumnos in $(cat $curso)
    do
        grupo=$(echo $grupo_alumnos | cut -d":" -f1)
        numAlumnos=$(echo $grupo_alumnos | cut -d":" -f2)
        for ((i=1;i<=$numAlumnos;i++))
        do
            #eliminamos al usuario
            userdel $grupo$curso$i 2>/dev/null
        done
        #borramos el grupo
        groupdel $grupo 2>/dev/null
    done
    #borramos los directorios del curso
    rm -r /home/$curso 2>/dev/null
}
```

#Programa principal, donde se comprueban los parámetros del Shell y se invocan a las
 #diferentes funciones, dependiendo de la opción elegida.

```
if [ $# -eq 2 ]
then
    case $1 in
        "-f")
            crear_curso $2
            ;;
        "-c")
            crear_cuentas $2
            ;;
        "-b")
            eliminar_cuentas $2
            ;;
        *)
            echo "La opción $1 es incorrecta"
            exit 1
            ;;
    esac
else
    echo "Número de parámetros incorrecto"
    exit 1
fi
```