

**PRÁCTICA Nº8****OPOSICIONES CUERPOS DOCENTES 2010  
SUPUESTO 1****PLANTEAMIENTO DEL EJERCICIO PRÁCTICO (Duración 2,5 h):  
SOLUCIÓN PROPUESTA**

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 **copiaSeg** para gestionar el sistema centralizado de copias de seguridad de un servidor.

El script sin parámetros hará una copia de seguridad de todos los usuarios del sistema, y con el parámetro **-u usuario** hará una copia de seguridad de la información del usuario pasado como parámetro. Este script sólo podrá ser ejecutado por root. La sintaxis del script será:

**copiaSeg [-u usuario ]**

Los usuarios podrán utilizar el sistema de copias automatizado creando en su directorio de trabajo un fichero oculto, llamado **copiaSeg.dat** con el siguiente contenido:

**#Copias que se desean mantener numeroCopias=2**

**#Contenido de la copia, único obligatorio**

**contenidoCopia=dir1:dir2:fichero1:fichero2,,.**

En caso de que no exista alguno de los campos configurables anteriores, se usarán los valores por defecto, definidos en el propio script. El único campo obligatorio es el campo **contenidoCopia**, que contendrá todos los ficheros y directorios que se desean incluir en la copia. Si dicho campo no estuviese definido, no se copiaría nada para ese usuario.

El sistema **creará** un directorio **/copiaSeg** y dentro un directorio por usuario, si no existiese, que sólo podrá ser accesible por su dueño. En el interior de este directorio se almacenarán tantas copias de seguridad **copiaSeg\_hhmmDDMMYYYY.tgz** como defina **numeroCopias**. Para que un fichero o directorio sea añadido a la copia debe ser de su propiedad.

El script **borrará** las copias más antiguas manteniendo en el sistema sólo las que se indiquen en el fichero de configuración **copiaSeg.dat** y, si no está definido **numeroCopias**, lo que marque el valor por defecto.

```
#!/bin/bash
#Compruebo si el usuario es root.
if [ $UID -ne 0 ]
then
    echo "Este script solo lo puede ejecutar el root"
    exit 1 # Si el root no ejecuta este Shell-script, devolvemos error,
fi

export numeroCopias=2 #variable global al Shell-script, que indica el número de
                        #copias por defecto.

# Función que pasándole como parámetro el usuario y el contenido (lista de
# ficheros # y directorios separados por ":" que queremos salvar) realiza la copia
# de seguridad la copia de seguridad
function realizarCopia ()
{
    usuario=$1
    directorios=$2
    carpeta="/copiaSeg/$usuario" #variable con el directorio de copias de seguridad
    if [ ! -d $carpeta ]
    then
        mkdir $carpeta
        chown $usuario.$usuario $carpeta
        chmod 700 $carpeta
    fi

    contenidoCopia=""
    antIFS="$IFS"
    IFS=":" #Establecemos el separador para dividir los directorios en el for por :
    for valor in $directorios
    do
        # comprobamos que el valor (fichero o directorio de la lista a salvar) existen y
        # es de usuario
        if [ "$(stat -c %U $valor 2> /dev/null)" = "$usuario" ]
        then
            contenidoCopia="$contenidoCopia:$valor"
        fi
    done

    if [ -n "$contenidoCopia" ]
    then
        tgz $carpeta/copiaSeg_$(date +%H%M%d%m%Y) $contenidoCopia 2>/dev/null
    fi
    IFS="$antIFS"
}
```

```

# Función que pasándole la ruta de un fichero de configuración devuelve los
# directorios a almacenar
function leeDirectorios ()
{
    directorioTrabajo=$1
    if [ -r $directorioTrabajo/.copiaSeg.dat ]
    then
        grep "^contenidoCopia" $directorioTrabajo/.copiaSeg.dat | head -1 | cut -d= -f2
    else
        echo ""
    fi
}

# Función que pasándole la ruta del fichero de configuración devuelve el número de
# copias a almacenar
function leeNumeroCopias ()
{
    directorioTrabajo=$1
    local copias
    if [ -r $directorioTrabajo/.copiaSeg.dat ]
    then
        copias=$(grep "^numeroCopias" $directorioTrabajo/.copiaSeg.dat | head -1 |
                                                         cut -d= -f2)
    fi
    if expr $copias + 0 >/dev/null 2>&1
    then
        echo $copias
    else
        echo $numeroCopias #valor por defecto
    fi
}

#Función que se le pasa un directorio y un número y deja sólo el número de copias
function eliminarCopiasSeguridad ()
{
    copiaSeg_Usuario=$1
    copias_a_almacenar=$2
    copiasEliminar=$(ls -t $copiaSeg_Usuario/copiaSeg_?????????????.tgz 2>/dev/null |
                                                             tail -n +$(( $copias_a_almacenar+1 )))

    for copia_a_eliminar in $copiasEliminar
    do
        rm -r $copia_a_eliminar
    done
}

function directorioTrabajo()
{
    usuario=$1
    cat /etc/passwd | grep "^$usuario:" | cut -d: -f6
}

```

```

if (($#==2)) && [ "$1" = "-u" ] && id $2 >/dev/null 2>&1
then
    usuario=$2
    directorios=$(leeDirectorios $(directorioTrabajo $usuario))
    if [ -n "$directorios" ]
    then
        realizarCopia $usuario $directorios
        eliminarCopiasSeguridad /copiaSeg/$usuario $(leeNumeroCopias
                                                                $(directorioTrabajo $usuario))
    fi
else
    if [ $# -eq 0 ]
    then

        for usuario in $(cat /etc/passwd | cut -d: -f1)
        do
            directorios=$(leeDirectorios $(directorioTrabajo $usuario))
            if [ -n "$directorios" ]
            then
                realizarCopia $usuario $directorios
                eliminarCopiasSeguridad /copiaSeg/$usuario $(leeNumeroCopias
                                                                $(directorioTrabajo $usuario))
            fi
        done
    else
        echo "Error en los parámetros"
        exit 1
    fi
fi

```