

```
#!/bin/bash
clear

function cpuram() {
clear
echo Processor $(cat /proc/cpuinfo | grep "model name" | head -1)
echo Memoria: $(cat /proc/meminfo | grep -i memtotal | awk '{print $2}') KB
echo Pulsa Enter para continuar
read nada
}

function espaciaraiz() {
clear
for directorio in $(ls /)
do
test -d /$directorio && du -sh /$directorio 2> /dev/null
done
echo Pulsa Enter para continuar
read nada
}

function prioridad() {
clear
renice -n -5 -u alumno
echo Pulsa Enter para continuar
read nada
}

function nginxinstalado() {
clear
dpkg -l |grep nginx &> /dev/null
case $? in
0) echo El servidor nginx esta instalado ;;
*) echo Nginx no esta instalado, lo instalamos; apt-get -y install nginx ;;
esac
echo Pulsa Enter para continuar
read nada
}

function nginxactivo() {
clear
netstat -tlnp | grep nginx &> /dev/null
case $? in
0) echo El servidor nginx esta activo ;;
*) echo Nginx no está activo, lo arrancamos; service nginx start ;;
esac
echo Pulsa Enter para continuar
read nada
}
```

```

function workers() {
echo "sed -i 's/worker_processes\ 4/worker_processes\ 10/g' "/etc/nginx/nginx.conf" > /tmp/atjobs.txt
echo "service nginx restart" >> /tmp/atjobs.txt
at -t 12050230 -f /tmp/atjobs.txt
atq
echo Pulsa Enter para continuar
read nada
}

function procesos() {
clear
while true
do
clear
echo Procesos nginx
echo =====
ps -U "www-data" -o uid,pid,cmd,rss --sort=-rss
echo Procesos de alumno que mas memoria consumen
echo =====
ps -U alumno -o uid,pid,cmd,rss --sort=-rss | head -6
sleep 3
done
echo Pulsa Enter para continuar
read nada
}
while true
do
clear
echo "=====
echo "1) Muestra marca y modelo del microprocesador y tamaño de memoria RAM"
echo "2) Muestra ocupación de todos los directorios de /"
echo "3) Subir ligeramente la prioridad a todos los procesos del usuario alumno"
echo "4) Comprobar si está instalado el servidor web nginx y si no instalarlo"
echo "5) Comprobar si el servidor web nginx está arrancando y respondiendo"
echo "6) Programar la reconfiguración del servidor nginx en la madrugada del domingo al lunes"
echo "7) Mostrar cada 3 segundos todos los procesos del usuario que corre nginx y los 5 que más memoria real consumen del
usuario alumno, ordenados por memoria real (rss) descendente (mostrar uid,pid,cmd y rss)"
echo "8) Salir"
echo "=====
echo "Introduce una opción:"
read opcion
case $opcion in
    1) cpuram;;
    2) espacioraiz;;
    3) prioridad;;
    4) nginxinstalado;;
    5) nginxactivo;;
    6) workers;;
    7) procesos;;
    8) echo "Finalizando programa"; break;;
    *) echo "Opción no válida, elige una opción entre 1 y 8";
    echo "Pulse una Enter para continuar"; read nada;;
esac
done

```

Ejercicio 3:

Realizar un script en bash llamado **ResetPassUsers.sh** que reseteará la password de todos los usuarios con login del sistema (con bash) excepto root utilizando el comando **chpasswd**. Para poder probarlo, previamente se crearán “a mano” los usuarios jaimito, jorgito, gilito y juanito con /bin/bash y directorio home. Se pasará la nueva contraseña como parámetro. El script debe comprobar si el usuario que lo ejecuta es root y si no mostrar mensaje e impedir su ejecución.

Ejemplo de uso de chpasswd: `echo 'usuario:contraseña' | chpasswd`

```
#!/bin/bash
#
# Script que resetea las passwords de todos los usuarios con la password pasada
# como parametro
# Uso: ResetPassUsers.sh <NuevaPassword>
#

function resetea() {
for usuario in $(cat /etc/passwd|grep bash|grep -v root|cut -f1 -d:)
do
echo Cambiando password a $usuario
echo =====
echo "$usuario:$contrasen"|chpasswd
done
}
contrasen=$1
clear
if [ $# -eq 1 ]
then
case $(id -u) in
0) resetea;;
*) clear; echo Debes ser root para ejecutar este script;;
esac
else
echo Numero de parametros incorrecto
fi
```