



# Preparador Informática

[www.preparadorinformatica.com](http://www.preparadorinformatica.com)

## PRÁCTICA 8 bis SHELL SCRIPTS

OPOSICIONES INFORMÁTICA 2018  
ARAGÓN

#### 4.6.- Administración. Script Bash

Dado a modo de ejemplo el siguiente archivo de configuración del sistema (p.e. /etc/mifichero.conf) conformado por secciones (config, volume, etc.), cada una con directivas en forma de clave/valor:

```
[config]
bm_open_method=0

[volume]
#num_mounts=4
mount_on_startup=0
mount_removable=0
#autorun=0,1
autorun=1

[autorun]
x-content/image-dcf=pcmanfm

[ui]
always_show_tabs=0
max_tab_chars=32
win_width=1402
win_height=737
splitter_pos=186
media_in_new_tab=1
desktop_folder_new_win=0
...
```

A) Realiza un script de administración en **bash** que cumpla los siguientes requisitos

A.1) El script deberá recibir tres parámetros: (1) la ruta del archivo de configuración, (2) nombre de la directiva clave a modificar y (3) el valor a asignar. En el caso de que el número de parámetros sea diferente o el fichero no exista se deberá mostrar un mensaje de error.

A.2) Se comprobará que la directiva clave indicada esta activa en el archivo (existe y no está comentada). En caso contrario se informará con un mensaje de error.

A.3) En el caso de que la directiva clave exista se actualizará su valor con el valor indicado como tercer parámetro.

Criterios de corrección y calificación

- Script funcional a lo que se solicita: *0,25 puntos*
- Se puntuará con 0 puntos en el caso de que la solución carezca de sentido.
- Los posibles comandos se podrán dar por válidos si semánticamente tienen sentido, aunque sintácticamente no sean correctos de manera literal.

## SOLUCIÓN PROPUESTA

```
#!/bin/bash
####CÓDIGOS DE ERROR####
ERROR_NUM_PARAM=1
ERROR_FICHERO=2
ERROR_DIRECTIVA=3
#####

#Comprueba que el número de parámetros es correcto
#Si el nº de parámetros es diferente de 3 muestra un mensaje de error
#y finaliza la ejecución del script
if [ $# -ne 3 ]
then
    echo "ERROR: Ejecución con número de parámetros incorrecto"
    echo "USO: $0 <ruta archivo conf> <directiva a modificar> <valor a asignar>"
    exit $ERROR_NUM_PARAM
fi

archivo=$1
directiva=$2
valor=$3
#Comprueba que el archivo de configuración existe y es un archivo regular
#Si no existe ningún fichero con el nombre especificado muestra un mensaje de error
#y finaliza la ejecución del script
if [ ! -f $archivo ]
then
    echo "ERROR: NO existe ningún fichero de configuración llamado $archivo"
    exit $ERROR_FICHERO
fi

#Comprueba que la directiva clave indicada está activa en el archivo
#Si la directiva está comentada o no existe muestra un mensaje de error
#y finaliza la ejecución del script
if [ ! $(grep ^$directiva= $archivo) ]
then
    echo "ERROR: La directiva no existe o está comentada"
    exit $ERROR_DIRECTIVA
fi

#Si el script se ejecuta correctamente, es decir pasa satisfactoriamente
#todas las comprobaciones, procede a actualizar la directiva con el valor indicado

#En la variable directivaActual guardamos la línea que contiene la directiva
#y su valor actual
directivaActual=`grep ^$directiva= $archivo`

#En la variable directivaActualizada guardamos el nombre de la directiva a actualizar
#y le concatenamos el signo igual y el nuevo valor de la directiva
directivaActualizada=$directiva="$valor"

#Como hay una directiva (x-content/image-dcf) que utiliza caracteres especiales (//)
#para que el comando sed no dé problemas y se confundan dichos caracteres con los
#caracteres separadores que normalmente se utilizan en el comando sed (/), se puede
#utilizar otro carácter separador como la ",".
#Para ver un ejemplo: https://rm-rf.es/como-escapar-caracteres-dentro-de-variable-sed/

#Con el comando sed actualizamos la directiva
#-i: por defecto, sed escribe la salida en la salida estándar. Esta opción le dice a sed
#que modifique los archivos en curso.
sed -i "s,$directivaActual,$directivaActualizada," $archivo
```

