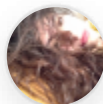


# WUOLAH



emewinchester

[www.wuolah.com/student/emewinchester](http://www.wuolah.com/student/emewinchester)



594

## practica 6 APSO resuelta.pdf

*PRÁCTICAS RESUELTAS (incluye teoría)*



**2º Administración y Programación de Sistemas Operativos**



**Grado en Ingeniería Informática**



**Escuela Técnica Superior de Ingeniería  
UHU - Universidad de Huelva**

# 5

**CONSIGUE UN 5  
PELAO EN INGLÉS  
Y GANA UN VIAJE  
A NUEVA YORK**

NO TE LIES  
**VIAJA**

**MY  
ENGLISH  
COOL**  
EASIER,  
CHEAPER...  
BETTER

**PARTICIPA Y GANA**

# APSO: PRÁCTICA 6

---

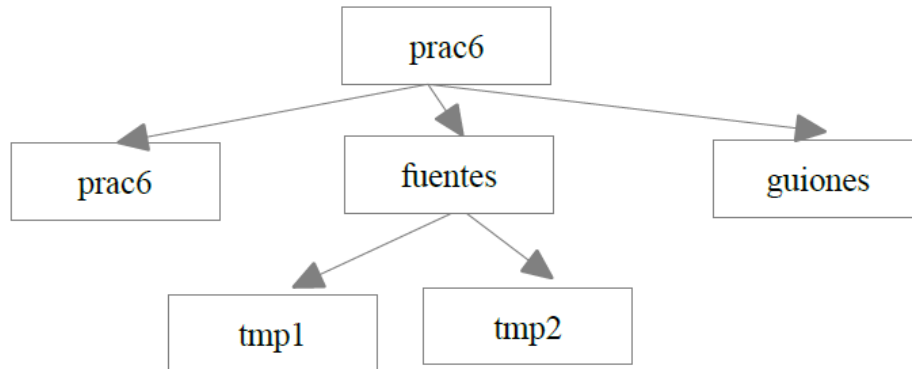
PROGRAMACIÓN DE SCRIPTS (II)

Marina Delgado Pérez  
GRADO INGENIERÍA INFORMÁTICA | HUELVA

# ENUNCIADO Y RESOLUCIÓN PRÁCTICA

---

1. Cree con un solo comando el siguiente árbol de directorios a partir de su directorio personal:



```
mkdir -p prac6/prac6 prac6/fuentes/{tmp1,tmp2} prac6/guiones
```

2. Incluya dentro del fichero `.profile` la variable `GUION` con la ruta al directorio `guiones` creado anteriormente. Haga que tome efecto esta variable. Copie todos los ficheros que a partir de su directorio personal terminen en `.txt` en el directorio `tmp1`.

```
joe .profile
(dentro del .profile)
GUION="$HOME/prac6/guiones"
(control+k+x)
. .profile
cp $(find . -iname "*.txt") prac6/fuentes/tmp1
```

3. Muévase al directorio `guiones`. Cree un guión en el directorio `guiones` llamado `tratafichero`. Este guión debe recoger un único parámetro. Si el parámetro es un fichero ordinario debe visualizar su contenido con el comando `more`. Si se trata de un directorio, se debe ver el contenido del mismo con el comando `ls -la`. Si no es ni un fichero ni un directorio debe hacerse un `echo` del parámetro. (Se necesita usar el comando `test` y la estructura `if`).

```

cd $GUION
joe tratafichero
(dentro del fichero...)
if test -f $1
then
    more $1
else if test -d $1
then
    ls -l
    else
    echo $1
    fi
fi
(control+k+x)
chmod u+x tratafichero

```

4. Muévase al directorio `prac6` que está dentro de `prac6`. Ejecute desde aquí el guión anterior dos veces. La primera vez pásese como parámetro `$HOME/prac2/f1.txt`. La segunda pásese como parámetro la variable `GUION` creada en el apartado 2.

```

cd ..
guiones/tratafichero $HOME/prac2/f1.txt
guiones/tratafichero $GUION

```

5. Muévase al directorio `guiones`. Cree un guión llamado `fpfinal` que haga lo siguiente:

Si no se pasan 3 parámetros visualiza un mensaje que dice: Se necesitan al menos 3 parámetros.

Si se pasan 3 parámetros se comprueba que el primero y el tercero son directorios. Si no lo son, visualiza el mensaje: El primer y tercer parámetro han de ser directorios.

Copia todos los nombres (solo el nombre, no la ruta) de enlaces simbólicos (ficheros soft link) encontrados a partir del primer directorio especificado, cuyo nombre coincida con el segundo parámetro, en el directorio especificado en tercer lugar con el nombre `enlaces.txt`. Los errores no deben salir por pantalla.

Ejecute este guion con los parámetros siguientes:

**Primer parámetro:** `/home`

**Segundo parámetro:** ficheros que empiecen por una letra mayúscula

**Tercer parámetro:** directorio `fuentes` creado en el apartado 1



*¿Estás de exámenes?  
Estos podríais ser  
tú y tu colega...*

# MENTES BRILLANTES

MENTES  
BRILLANTES

```
cd $GUION
joe fpfinal
(dentro del fichero...)

if [ $# -ne 3 ]
then
    echo "Se necesitan al menos 3 parametros."
else if [ $# -eq 3 ]
then
    if [ -d $1 ] && [ -d $3 ]
    then
        find $1 -type l -name "$2" -printf "\n%f"
        >$3/enlaces.txt 2>/d
    else
        echo "El primer y tercer parametro han de ser
directorios."
    fi
fi
fi
(control+k+x)
Fpfinal /home "[A-Z]*" $HOME/prac6/fuentes
```

6. Cree un guión llamado `mtam` que muestre, para cada fichero encontrado a partir del directorio pasado como primer parámetro cuyo nombre coincide con el segundo parámetro, su tamaño. Compruebe que el número de parámetros es el correcto y que el primero es un directorio. Si hay algún error debe comunicarse por pantalla. (Usa el `for` en el guión).

Ejécútelos con los parámetros siguientes:

**Primer parámetro:** directorio `fuentes` creado anteriormente.

**Segundo parámetro:** ficheros que empiecen por `s` y tengan un punto en su nombre.

```
joe mtam
(dentro del fichero de texto...)
```

```
if [ $# -ne 2 ]
then
    echo "Debe introducir dos parámetros."
else
    if [ ! -d $1 ]
    then
        echo "El primer parametro debe ser un directorio."
    else
```

Marina Delgado Pérez

3

WUOLAH

una película de  
**THOMAS  
LILTI**

VINCENT  
LACOSTE

WILLIAM  
LEBGHIL

★★★★★

"IRRESISTIBLE,  
HILARANTE Y  
CONMOVEDORA"

ROLLING STONE

5 DE ABRIL  
EN CINES

```

        for I in $(find $1 -type f -name $2)
        do
            wc -c $I
        done
    fi
fi
(control+k+x)
Chmod u+x mtam
mtam $HOME/prac6/fuentes "s*.*"

```

7. Cree un guión llamado `anida`, al que se le pasará un parámetro y deberá hacer lo siguiente:

Pedirá que se introduzca un número por el teclado.  
 Creará tantos directorios anidados, con el nombre que se ha pasado como primer parámetro, como indique el número leído, a partir del directorio en el que nos encontramos

**Ejemplo:** Se ejecuta `anida p1` y el número que se lee es 3. Esto creará un directorio `p1`, que tiene dentro un directorio `p1`, que a su vez tiene dentro un directorio `p1`, a partir del directorio en el que estamos.

**NOTA:** Se monta un bucle `while`, que itera tantas veces como indique el contador, y dentro se va creando la ruta de los subdirectorios sobre una variable, que es la usada para crear los subdirectorios.

Muévase al directorio `tmp2`. Ejecute desde aquí el guión anterior de la siguiente forma: `anida p6` y se teclea el número 2 por teclado cuando lo solicite.

```

joe anida
(dentro del fichero de texto...)

echo "Introduzca por teclado un número: "
read NUMERO

DIRECTORIO="$PWD"

while [ $NUMERO -gt 0 ]
do
    mkdir $DIRECTORIO/$1
    DIRECTORIO="$DIRECTORIO/$1"
    let NUMERO=$NUMERO-1
done

(control+k+x)
Chmod u+x anida
cd $HOME/prac6/fuentes/tmp2
$GUION/anida p6

```

\$GUION/anida p1

8. Muévase al directorio `guiones`. Cree un guión llamado `opcion` que lea una palabra del teclado. Si la palabra empieza por un número almacena la palabra al final de un fichero llamado `numero`, en un fichero llamado `vocal` si empieza por vocal y en un fichero llamado `otro` para cualquier otro caso. Use el `case` para decidir a que fichero va la palabra. Debe estar haciendo siempre lo mismo hasta que se pulse un punto, pero no debe parar si se introduce un punto seguido de cualquier otro conjunto de caracteres.

```
cd $GUION
joe opcion
(dentro del guion...)
#inicializo la variable para que no me compare con basura
PALABRA="inicio"
until [ $PALABRA = "." ]
do
    echo "Introduzca una palabra: "
    read PALABRA
    if [ $PALABRA != "." ]
    then
        case $PALABRA in
            [0-9]*)
                echo $PALABRA >>numero ;;

            [aeiouAEIOU]*)
                echo $PALABRA >>vocal ;;

            *)
                echo $PALABRA >>otro ;;
        esac
    fi
done

# NO USAR COMILLAS  despues del IN para describir la opcion, es
# SIN COMILLAS
# NO USAR EL CAT PORQUE CAT TRABAJA CON FICHEROS, NO CON CARACTERES
# COMO TAL
(control+k+x)
chmod u+x opcion
```

9. Modifique los permisos de los directorios `tmp1` y `tmp2` para que el propietario tenga todos los permisos menos el de ejecución, el grupo solo el de lectura y los otros solo el de ejecución.

```
chmod 641 ../fuentes/temp[12]
```

10. Consulte el número de procesos total que tiene activos en este preciso momento (contando los de todas las ventanas abiertas). (Se usa `ps` y `wc`).

```
ps -u | wc -l
```