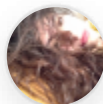


WUOLAH



emewinchester

www.wuolah.com/student/emewinchester



597

practica 5 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 5

PROGRAMACIÓN DE SCRIPTS (I)

Marina Delgado Pérez
GRADO INGENIERÍA INFORMÁTICA | HUELVA

diferénciate

Con la mejor formación práctica

www.mastersevilla.com

Titulación de prestigio
en el sector empresarial

MÁSTER EN DIRECCIÓN Y
GESTIÓN DE RECURSOS HUMANOS



BECAS

Contenido

1.- GUIONES	2
2.- PASO DE PARÁMETROS.....	2
3.- USO DE COMILLAS.....	4
Comilla simple	4
Comilla doble.....	4
Usos y comparaciones.....	4

PRELUDIO TEÓRICO

1.- GUIONES

Un guion es un fichero de texto ejecutable, con un comando de ejecución por línea.

Los guiones pueden crearse con cualquier editor de texto (por ejemplo, el editor de texto joe), con la característica principal de que cada línea de ese fichero debe ser un comando o una estructura de control de flujo (if, for, while...).

Tras escribir los comandos en el guion que queramos que se ejecuten, este se guarda (control + K + X) y otorgamos permisos de ejecución al fichero.

Los guiones, al ser ficheros ejecutables, para que el SO sea capaz de localizarlos deben:

- Estar en la ruta de la variable PATH (método correcto).
- O bien damos la ruta del fichero ejecutable al ejecutarlo (./nombre_guion).

El segundo método solo debe ser usado en caso de no ser capaz de modificar y añadir la ruta del fichero a la variable PATH.

2.- PASO DE PARÁMETROS

Los guiones nos permiten pasarles parámetros y poder trabajar con ellos dentro del fichero de texto. Los parámetros están separados por espacios en blanco. La sintaxis sería:

nombre_guion	parámetro_1	parámetro_2	...	parámetro_n
--------------	-------------	-------------	-----	-------------

Los parámetros son almacenados en variables, a las cuales podemos acceder mediante el símbolo dólar. Ejemplo:

`$numero_variable`

El nombre del guion también es guardado en una variable (variable 0). Otras variables importantes y útiles a la hora de programar scripts son:

- \$@: Guarda todos los parámetros, de manera literal (caracteres que se pasan por parámetro).
- \$#: Guarda el número de parámetros pasados al guion, sin contar el nombre del propio guion.

5

SACA UN 5 PELAO
EN NUESTRO TEST
DE INGLÉS



Y GANA
UN VIAJE A
NUEVA YORK

PARTICIPA Y GANA

EJEMPLO:

guion_ejemplo

```
echo $1

date +"Bienvenido, son las %H horas %m
minutos"

echo $2 $3 $4

echo $4 $3 $2

echo $#

echo $@

ls -l
```

La salida del script `guion_ejemplo` variará en función de los parámetros que le pasemos, por ejemplo:

- (1) `guion_ejemplo "como están ustedes"` (lo que está entre paréntesis se considera un solo parámetro).
- (2) `guion_ejemplo como están ustedes` (cada palabra es un parámetro)

Salidas de los dos casos:

Caso 1

```
como están ustedes

Bienvenido, son las 13
horas 03 minutos

(no mostraría nada)

(no mostraría nada)

1

como están ustedes

(contenido de la
carpeta donde estemos)
```

Caso 2

```
como

Bienvenido, son las 13
horas 03 minutos

están ustedes

ustedes estan

3

como están ustedes

(contenido de la
carpeta donde estemos)
```

PARTICIPA
AQUÍ



notelies.
myenglishcool.
com

NO TE LIES
VIAJA

MY
ENGLISH
COOL
EASIER.
CHEAPER...
BETTER

3.- USO DE COMILLAS

Comilla simple

La comilla simple anula por completo el efecto de un carácter especial. Esto se usa para llevar un carácter a un comando para que lo interprete el comando que lo tenga que interpretar.

Comilla doble

La comilla doble no anula el efecto de los caracteres especiales, sirven simplemente para delimitar un conjunto de caracteres.

Usos y comparaciones

Si lo que estamos entrecomillando es simplemente texto, el uso de la comilla simple o doble es indiferente, producirá el mismo resultado.

```
mkdir "esto es una prueba"  
mkdir 'esto es una prueba'
```

Ambas instrucciones crean un directorio llamado "esto es una prueba". Otra manera de hacer la misma operación sería:

```
mkdir esto\ es\ una\ prueba
```

La diferencia entre comillas dobles o simples toma importancia cuando tratamos con variables en la línea de comando, de forma que el la Shell expande una variable en su contenido cuando se ejecuta, es decir:

```
prueba="Esto es una prueba"
```

```
echo $prueba
```

```
echo "$prueba"
```

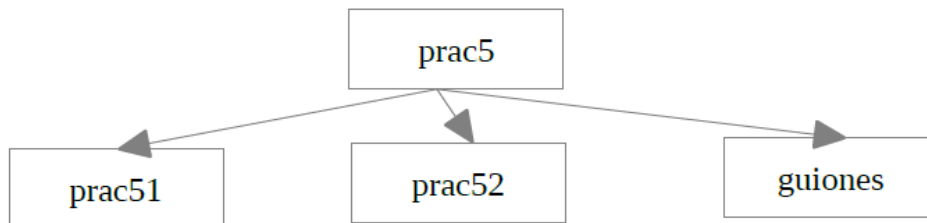
Tras ejecutar `echo $prueba`, el terminal devolverá: **Esto es una prueba**

Sin embargo, si ejecutamos `echo '$prueba'`, el terminal devolverá: **\$prueba**

En pocas palabras, las comillas dobles permiten interpretar el significado de los caracteres especiales (en este caso, el símbolo del dólar), mientras que las comillas simples anulan el significado de los caracteres especiales y hace que el IC los tome por caracteres simples.

ENUNCIADO Y RESOLUCIÓN PRÁCTICA

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



```
mkdir prac5 prac5/prac51 prac5/prac52 prac5/guiones
```

2. Añada en el fichero `.profile` a la variable `PATH` una nueva ruta, la ruta hasta el directorio `guiones` que hemos creado en el apartado 1. Esta nueva línea no tendrá efecto hasta que no ejecutemos el fichero `.profile`. Ejecute el fichero `.profile` de la forma adecuada para que el nuevo `PATH` tome efecto. Visualice de nuevo la variable `PATH` para asegurarse que tiene lo que tenía en la práctica 4 más la nueva ruta que hemos añadido en este apartado.

```
joe .profile
(dentro del fichero -> PATH=$PATH:$HOME/prac5/guiones )
(control + k + x)
. .profile
echo "$PATH"
```

!!!**IMPORTANTE!!!** No continúe adelante si no ha conseguido añadir esa nueva ruta a la variable `PATH`. Si continúa con el resto de los apartados sin hacer los anteriores no podrá realizarlos correctamente.

3. Muévase al directorio `guiones`. Cree un guion llamado `busca` en este directorio que reciba como parámetros un directorio y un número. El guion debe buscar los ficheros (no directorios) a partir del directorio especificado que hayan sido modificados hace menos de `n` días, siendo `n` el número de días especificado. Muévase de nuevo a su directorio personal. Ejecute desde aquí el guion pasando como parámetros su directorio personal y el número 10.

```
cd prac5/guiones
joe busca
(dentro del guion)
find $1 -type f -mtime -$2
(guardo guion: control + k + x)
chmod u+x busca
cd
busca . 10
```


4. Cree tres nuevos alias en el fichero .profile.

Uno se llamará listaf y visualizará la lista de todos los ficheros y directorios a partir de su directorio personal (incluso los que empiezan por punto) así como sus permisos (opción -l).

Otro se llamará gente y contará los usuarios que actualmente se encuentran en el sistema (Se supone que cada fila del comando who es un usuario. Otro se llamará saludo y visualizará el siguiente mensaje:

Hola. Son las <hora> horas y <minuto> minutos del <día de la semana en letras> <día del mes> de <mes> de <año>. Chao

HAGA QUE TOMEN EFECTO

```
joe .profile
(editamos fichero)
alias listaf='ls -l -a -R "$HOME"'
alias gente='who | wc -l'
alias saludo='date +"Hola. Son las %H horas y %m minutos
del %A %d de %B de %Y. Chao."'
(guardamos fichero-> control + k + x)
. .profile
```

5. Cree desde aquí, en el directorio guiones, un guion llamado fnuevos que pasándole como parámetro el nombre de un fichero y un directorio indique cuántos ficheros hay a partir del directorio indicado que son más nuevos que el fichero especificado (debe salir un número). Cree otro guion llamado fviejos que pasando los mismos parámetros indique cuántos son más viejos que el fichero especificado.

```
joe prac5/guiones/fnuevos
(editamos el fichero)
find $2 -newer $1 | wc -l
(guardamos fichero: control+k+x)
chmod i+x prac5/guiones/fnuevos
```

```
joe prac5/guiones/fviejos
(editamos el fichero)
find $2 ! -newer $1 | wc -l
(guardamos fichero: control+k+x)
chmod i+x prac5/guiones/fviejos
```



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

MENTES BRILLANTES

MENTES
BRILLANTES

6. Vaya al directorio `prac51`. Cree un guión llamado `trataf` en este directorio que para cada fichero que encuentre a partir del directorio pasado como parámetro que termine en `.txt` lo visualice por pantalla página a página. Los errores no deben salir por pantalla. Muévase a su directorio personal. Compruebe desde aquí su ejecución pasando como parámetro el directorio `/home/so/velez/MI`.

(NOTA: este ejecutable no se encuentra en ninguna de las rutas del PATH, así que hay que especificar la ruta para poder ejecutarlo)

```
cd prac5/prac51
joe trataf
(dentro del fichero trataf...)
more $(find 2>/dev/null "$1" -type f -iname "*.txt")
(guardamos fichero)
chmod u+x trataf (asignamos permisos de ejecución al propietario)
./trataf "$ORIGEN" (comprobamos si funciona)
```

7. Introduzca en el fichero `.profile` la modificación realizada a la variable `PS1` (en la practica 4), para que el prompt sea nuestro nombre de usuario, seguido de la ruta y del símbolo `>`. Haga que tome efecto.

```
joe .profile
(editamos el .profile)
PS1='$LOGNAME: $PWD>'
(guardamos el .profile -> control + k + x)
. .profile
(debería aparecer el nuevo prompt justo después del comando anterior)
```

8. Usando el comando `cat` cree un guion en el directorio guiones llamado `copiar` al que le pasen como parámetro dos nombres de ficheros y la ruta absoluta a un directorio y copie el fichero especificado en primer lugar en el directorio especificado, con el nombre del fichero especificado en segundo lugar.

```
profile
cat >prac5/guiones/copiar
cp $1 $3/$2
(control+d)
```

9. Muévase al directorio `guiones`. Cree un guion llamado `vertodo` en este directorio que visualice página a página el contenido de todos los ficheros (no directorios) que a partir del directorio en el que se encuentra en ese momento empiecen por `s` y terminen en `.c` o `.p` y sean más nuevos que un fichero pasado como parámetro. Los posibles errores no deben salir por pantalla y los ficheros deben ser visualizados por orden alfabético.

(NOTA: el nombre de un fichero viene determinado por su RUTA ABSOLUTA)

una película de
**THOMAS
LILT**

VINCENT
LACOSTE

WILLIAM
LEBGHIL

★★★★★

"IRRESISTIBLE,
HILARANTE Y
CONMOVEDORA"

ROLLING STONE

5 DE ABRIL
EN CINES

```
cd prac5/guiones
joe vertodo
(dentro del fichero)
more $(find . ! -type d -iname "s*.[pc]" -newer $1
2>/dev/null | sort)
(control+k+x)
chmod u+x vertodo
```

10. Cree en el directorio *guiones* un guion llamado *infor* que muestre la siguiente información:

- Su nombre de usuario
- Su grupo de usuario
- Terminal en el que se encuentra
- Fecha y hora actual del sistema
- Calendario de Marzo de este año

```
joe infor
(dentro del fichero)
echo "$LOGNAME"
id -n -g
tty
cal marzo 2019
```