

CAPÍTULO 3: PROGRAMACIÓN BÁSICA

- 3.1. Implemente la utilidad *basename*, la cual muestra por la salida estándar (*stdout*) el nombre del fichero que se le pase como argumento, eliminando el prefijo del directorio que pueda contener. Por ejemplo, dado el path *a/b/c/d*, *basename* devuelve *d*. El siguiente ejemplo muestra y registra el proceso de compilación e instalación de una aplicación GNU.

```
$ basename a/b/d/c
d
```

Para resolver este problema se van a utilizar expresiones regulares para extraer una subcadena con la parte final de la cadena dada hasta alcanzar el carácter '/', éste inclusive, eliminando la parte inicial de la cadena original. Una vez hecho esto, se eliminará dicho carácter quedando únicamente la parte buscada.

```
#!/bin/bash
# mibasename Implementación propia de la utilidad basename
# uso mibasename cadena

# Se asigna el parámetro a una variable
cadena=$1
# Se extrae la subcadena del final hasta /
subcadena=`expr match "$cadena" '.*\(/.*\)'`
# Se elimina el patrón /
resultado=${subcadena##*\}/
# Se muestra el resultado
echo $resultado
```

- 3.2. La utilidad *basename* tiene un segundo parámetro opcional (*basename path [sufijo]*). Si indica dicho parámetro, y éste coincide con el sufijo del fichero, también se eliminará. Por ejemplo:

```
$ basename include/stdio.h .h
stdio
```

Modifique el script creado en el ejercicio 3.1 para añadir dicha funcionalidad.

Simplemente se debe añadir una orden para la eliminación de patrones al final de una cadena.

```
#!/bin/bash
# mibasename Implementación propia de la utilidad basename
# uso mibasename cadena

# Se asigna el parámetro a una variable
cadena=$1
# Se extrae la subcadena del final hasta /
subcadena=`expr match "$cadena" '.*\(/.*\)'`
# Se elimina el patrón /
resultado=${subcadena##*/}
# Se elimina el sufijo
sufijo=$2
resultado=${resultado%%$sufijo}
# Se muestra el resultado
echo $resultado
```

3.3. Determine lo que realiza cada una de las siguientes órdenes:

- `TEMPORAL=`grep "^$1:" /etc/passwd 2>/dev/null``

Localiza en el fichero */etc/passwd* todas aquellas líneas que comiencen por lo indicado en el primer parámetro de la llamada seguido de dos puntos (:) ignorando los posibles errores que se puedan producir. Ese campo del fichero */etc/passwd* se corresponde con el nombre de usuario.

Por lo tanto, intenta localizar al usuario que se indica como primer parámetro de la llamada al script.

- `grep ":$GID:" /etc/group | cut -f1 -d:`

Presenta el nombre del grupo (se busca en el fichero */etc/group*) cuyo identificador se encuentra en la variable GID.

3.4. Los directorios de búsqueda que almacena la variable de entorno PATH a veces resultan difíciles de discriminar debido a que cuesta encontrar el delimitador dos puntos (:). Escriba un script que muestre los directorios de PATH, cada uno en una línea.

Para realizar el ejercicio se va a hacer uso de la opción *-e* de la orden *echo*, que habilita la interpretación de los caracteres especiales por parte de este comando. Gracias a esta opción y simplemente haciendo un reemplazo de caracteres en la cadena PATH (: por el carácter especial de salto de línea *\n*) se consigue lo solicitado.

```
$ echo -e ${PATH//\:/\\n}
```