

# Programación y Administración de Sistemas

## Práctica 2. Programación de *shell scripts*

Convocatoria de junio (curso 2015/2016)

Pedro Antonio Gutiérrez Peña

7 de marzo de 2016

### Resumen

Se entregará el código de los *scripts*, junto con un fichero de texto que aclare las particularidades de los programas desarrollados e incluya ejemplos de ejecución. Es obligatorio comprobar errores en los argumentos que reciben los *scripts*. Se valorará la utilización de comentarios, la máxima modularidad en el código y la claridad en las salidas generadas. Todos los *scripts* deben prepararse para funcionar correctamente con el intérprete de órdenes `bash` de la máquina `ts.uco.es`. El día tope para la entrega de este guión de prácticas es el **domingo 17 de abril a las 23.55h**. La entrega se hará utilizando la tarea en Moodle habilitada al efecto. En caso de que dos alumnos entreguen códigos copiados, no se puntuarán ninguno de los dos. Comprueba que los comportamientos de los programas son similares a los esperados en los ejemplos de ejecución. Debes respetar los nombres de los *scripts* (`ejercicio1.sh...`) y comprimir todos ellos en un fichero `.tar.gz` llamado `apellido1.apellido2.nombre.PASJun1516P2.tar.gz`.

## 1. `ejercicio1.sh`

Crear un *script* que reciba como argumento un parámetro que será un directorio y un segundo argumento que será un número (número de *bytes*). El *script* debe buscar todos los ficheros que estén alojados en la carpeta que se pasa como primer argumento, cuyo tamaño sea mayor o igual que el número pasado como segundo argumento. Si no se especifica un segundo argumento, el *script* deberá mostrar todos los ficheros (es decir, como si el tamaño fuese 0). Para cada fichero, habrá que mostrar:

1. La carpeta que contiene al fichero.
2. El nombre del fichero sin carpeta.
3. El tamaño del fichero.
4. El número de referencias al i-nodo del fichero.
5. La cadena de permisos.
6. Un 1 si el fichero es ejecutable o un 0 si no lo es.

Para obtener el tamaño del fichero, el número de referencias al i-nodo y la cadena de permisos, utilizar el comando `stat`<sup>1</sup>. Todos estos campos se separarán por el carácter “;” y la salida deberá ordenarse numéricamente por tamaño de fichero (de mayor a menor). Un ejemplo de ejecución es el siguiente:

---

<sup>1</sup>Consulta `man stat`

```

1 i02gupep@NEWS:~/pas/1516$ ./ejercicio1.sh ejemploCarpeta/
2 Carpeta;Nombre;Tamaño;ReferenciasInodo;Permisos;Ejecutable
3 ejemploCarpeta/d1;svm_model_matlab.c;99;1;-rw-----;0
4 ejemploCarpeta/d1;smo_routine.c;104;1;-rw-----;0
5 ejemploCarpeta/images/Carpeta1;richard-stallman.jpg;42999;1;-rw-----;0
6 ejemploCarpeta/images/Teoria;raid0.png;50683;1;-rw-----;0
7 ejemploCarpeta/d2;prueba.c;233;1;-rw-----;0
8 ejemploCarpeta/images/Carpeta1;pirates.jpg;9836;1;-rw-----;0
9 ejemploCarpeta/images/logos;Logo_uco.gif;20005;1;-rw-----;0
10 ejemploCarpeta/images/logos;logo_eps_grande.jpg;43161;1;-rw-----;0
11 ejemploCarpeta/images/Carpeta1;Linus_torvalds.jpg;17670;1;-rw-----;0
12 ejemploCarpeta/d1/d4;libsysfs.so.2;38644;1;-rw-----;0
13 ejemploCarpeta/d1;libhandle.so.1.0.3;10760;1;-rw-r--r--;0
14 ejemploCarpeta/d2;libglib-2.0.so.0;822344;1;-rw-----;0
15 ejemploCarpeta/images/Teoria;hd.png;36710;1;-rw-----;0
16 ejemploCarpeta/images/Teoria;fedora.png;2386;1;-rw-----;0
17 ejemploCarpeta/images/Carpeta1;elinfiernoexiste.jpg;119847;1;-rw-----;0
18 ejemploCarpeta/d2;ejemploSuid.sh;483;1;-rw-----;0
19 ejemploCarpeta/d3;eje2;7275;1;-rwx-----;1
20 ejemploCarpeta/d1;ecoc.h;2377;1;-rw-----;0
21 ejemploCarpeta/images/Carpeta1;devocion-administrador-de-sistemas.png;21613;1;-rw-----;0
22 ejemploCarpeta/images/Teoria;debian.png;2626;1;-rw-----;0
23 ejemploCarpeta/d1;compat.h;370;1;-rw-----;0
24 ejemploCarpeta/images/Teoria;centos.png;8215;1;-rw-----;0
25 ejemploCarpeta/d1/d4;bcache.cpp;1233;1;-rw-----;0
26 ejemploCarpeta/d1;auto_fs.h;2288;1;-rw-r--r--;0
27 ejemploCarpeta/d1;audit.h;18220;1;-rw-r--r--;0
28 ejemploCarpeta/d2;a.out;7275;1;-rwx-----;1
29 ejemploCarpeta/d1/d4;act_func.c;92;1;-rw-----;0
30 ejemploCarpeta/images/Teoria;a2.out;7275;1;-rwx-----;1
31
32 i02gupep@NEWS:~/pas/1516$ ./ejercicio1.sh ejemploCarpeta/ 600
33 Carpeta;Nombre;Tamaño;ReferenciasInodo;Permisos;Ejecutable
34 ejemploCarpeta/images/Carpeta1;richard-stallman.jpg;42999;1;-rw-----;0
35 ejemploCarpeta/images/Teoria;raid0.png;50683;1;-rw-----;0
36 ejemploCarpeta/images/Carpeta1;pirates.jpg;9836;1;-rw-----;0
37 ejemploCarpeta/images/logos;Logo_uco.gif;20005;1;-rw-----;0
38 ejemploCarpeta/images/logos;logo_eps_grande.jpg;43161;1;-rw-----;0
39 ejemploCarpeta/images/Carpeta1;Linus_torvalds.jpg;17670;1;-rw-----;0
40 ejemploCarpeta/d1/d4;libsysfs.so.2;38644;1;-rw-----;0
41 ejemploCarpeta/d1;libhandle.so.1.0.3;10760;1;-rw-r--r--;0
42 ejemploCarpeta/d2;libglib-2.0.so.0;822344;1;-rw-----;0
43 ejemploCarpeta/images/Teoria;hd.png;36710;1;-rw-----;0
44 ejemploCarpeta/images/Teoria;fedora.png;2386;1;-rw-----;0
45 ejemploCarpeta/images/Carpeta1;elinfiernoexiste.jpg;119847;1;-rw-----;0
46 ejemploCarpeta/d3;eje2;7275;1;-rwx-----;1
47 ejemploCarpeta/d1;ecoc.h;2377;1;-rw-----;0
48 ejemploCarpeta/images/Carpeta1;devocion-administrador-de-sistemas.png;21613;1;-rw-----;0
49 ejemploCarpeta/images/Teoria;debian.png;2626;1;-rw-----;0
50 ejemploCarpeta/images/Teoria;centos.png;8215;1;-rw-----;0
51 ejemploCarpeta/d1/d4;bcache.cpp;1233;1;-rw-----;0
52 ejemploCarpeta/d1;auto_fs.h;2288;1;-rw-r--r--;0
53 ejemploCarpeta/d1;audit.h;18220;1;-rw-r--r--;0
54 ejemploCarpeta/d2;a.out;7275;1;-rwx-----;1
55 ejemploCarpeta/images/Teoria;a2.out;7275;1;-rwx-----;1

```

Si redirigimos la salida producida hacia un archivo .csv, ésta se debería poder abrir utilizando LibreOffice Calc (habría que elegir como separador únicamente el “;”):

```

1 i02gupep@NEWS:~/pas/1516$ ./ejercicio1.sh ejemploCarpeta/ 600 > salida.csv
2 i02gupep@NEWS:~/pas/1516$ libreoffice5.0 salida.csv

```

## 2. ejercicio2.sh

Crear un *script* que clasifique los ficheros que hay dentro de una carpeta en función de su tamaño en pequeños, medianos y grandes. El primer argumento del *script* recibido por línea de comandos será la carpeta a procesar (siendo un argumento obligatorio). Habrá dos argumentos adicionales y opcionales que serán los umbrales para discernir las tres categorías (`umbral1` y `umbral2`), de forma que los ficheros pequeños serán todos aquellos cuyo tamaño sea menor que `umbral1`, los medianos serán aquellos ficheros cuyo tamaño esté entre `umbral1` y `umbral2` (incluidos), y los grandes serán los ficheros con tamaño mayor que `umbral2`. Se supone que los ta-

maños se especifican en *bytes*. El valor por defecto para `umbral1` y `umbral2` será 10000 y 100000, respectivamente. El *script* debe crear tres subcarpetas (pequenos, medianos y grandes) dentro de la carpeta en la que se realiza la llamada al *script* e incluir en cada una los ficheros correspondientes. Si las carpetas ya existen, borrarlas.

Un ejemplo de ejecución es el siguiente (observa como el fichero `libhandle.so.1.0.3` no es “mediano” en la segunda ejecución):

```

1 i02gupep@NEWTS:~/pas/1516$ ./ejercicio2.sh ejemploCarpeta/
2 Creando las carpetas pequenos, medianos y grandes...
3 Copiando los archivos...
4 i02gupep@NEWTS:~/pas/1516$ ls pequenos/ medianos/ grandes/
5 grandes/:
6 elinfiernoexiste.jpg libglib-2.0.so.0
7 medianos/:
8 audit.h                                hd.png                                libsysfs.so.2                        logo_eps_grande.jpg
9 raid0.png
10 devocion-administrador-de-sistemas.png libhandle.so.1.0.3 Linus_torvalds.jpg Logo_uco.gif
11 richard-stallman.jpg
12 pequenos/:
13 a2.out                                bcache.cpp                          compat.h                            ecoc.h                            ejemploSuid.sh                    pirates.jpg                        smo_routine.c
14 act_func.c                            auto_fs.h                          centos.png                         debian.png                         eje2                               fedora.png                       prueba.c                          svm_model_matlab.
15 c
16
17 i02gupep@NEWTS:~/pas/1516$ ls -la medianos/libhandle.so.1.0.3
18 -rw----- 1 i02gupep upi0 10760 feb 28 13:26 medianos/libhandle.so.1.0.3
19
20 i02gupep@NEWTS:~/pas/1516$ ./ejercicio2.sh ejemploCarpeta/ 10761 100000
21 Las carpetas de salida ya existen, se va a proceder a borrarlas...
22 Copiando los archivos...
23
24 i02gupep@NEWTS:~/pas/1516$ ls pequenos/ medianos/ grandes/
25 grandes/:
26 elinfiernoexiste.jpg libglib-2.0.so.0
27 medianos/:
28 audit.h devocion-administrador-de-sistemas.png hd.png libsysfs.so.2 Linus_torvalds.jpg
29 logo_eps_grande.jpg Logo_uco.gif raid0.png richard-stallman.jpg
30 pequenos/:
31 a2.out                                bcache.cpp                          compat.h                            ecoc.h                            ejemploSuid.sh                    libhandle.so.1.0.3                prueba.c
32 svm_model_matlab.c
33 act_func.c                            auto_fs.h                          centos.png                         debian.png                         eje2                               fedora.png                       pirates.jpg
34 smo_routine.c

```

### 3. ejercicio3.sh

Realizar un *script* que reciba el nombre de dos directorios por la línea de comandos. El *script* copiará todos los ficheros que haya en el primer directorio (o en cualquier subdirectorio del mismo) al segundo directorio, pero los nombrará en mayúsculas (es decir, todas las letras del nombre de fichero serán transformadas a mayúsculas<sup>2</sup>). Esto deberá hacerse tanto para carpetas como para directorios. Se respetará la estructura de directorios de la carpeta original. Se recomienda seguir los siguientes pasos<sup>3</sup>:

- Crear la carpeta destino.
- Generar la estructura de directorios destino. Usar `find` para listar los **directorios** de la carpeta origen y crear las carpetas destino correspondientes (en mayúscula) una a una.
- Copiar los ficheros. Usar `find` para listar los **ficheros** de la carpeta origen y copiarlos a la carpeta destino correspondiente (en mayúscula).

Un ejemplo de ejecución sería el siguiente:

```

1 i02gupep@NEWTS:~/pas/1516$ ./ejercicio3.sh ejemploCarpeta/ ejemploCarpetaMayusculas
2 Procesando archivos...

```

<sup>2</sup>El comando `tr 'a-z' 'A-Z'` devuelve la cadena que se le pasa por el `stdin` en mayúsculas

<sup>3</sup>También se podría realizar mediante una implementación recursiva, con una función que procesa una ruta, renombrando todos los ficheros que cuelgan de ella y llamando recursivamente a la misma función si se trata de un directorio.

```

3 i02gupep@NEWS:~/pas/1516$ find ejemploCarpetaMayusculas/
4 ejemploCarpetaMayusculas/
5 ejemploCarpetaMayusculas/D2
6 ejemploCarpetaMayusculas/D2/LIBGLIB-2.0.SO.0
7 ejemploCarpetaMayusculas/D2/A.OUT
8 ejemploCarpetaMayusculas/D2/PRUEBA.C
9 ejemploCarpetaMayusculas/D2/EJEMPLOSUID.SH
10 ejemploCarpetaMayusculas/IMAGES
11 ejemploCarpetaMayusculas/IMAGES/CARPETA1
12 ejemploCarpetaMayusculas/IMAGES/CARPETA1/ELINFIERNOEXISTE.JPG
13 ejemploCarpetaMayusculas/IMAGES/CARPETA1/LINUS_TORVALDS.JPG
14 ejemploCarpetaMayusculas/IMAGES/CARPETA1/RICHARD-STALLMAN.JPG
15 ejemploCarpetaMayusculas/IMAGES/CARPETA1/DEVOCION-ADMINISTRADOR-DE-SISTEMAS.PNG
16 ejemploCarpetaMayusculas/IMAGES/CARPETA1/PIRATES.JPG
17 ejemploCarpetaMayusculas/IMAGES/LOGOS
18 ejemploCarpetaMayusculas/IMAGES/LOGOS/LOGO_UCO.GIF
19 ejemploCarpetaMayusculas/IMAGES/LOGOS/LOGO_EPS_GRANDE.JPG
20 ejemploCarpetaMayusculas/IMAGES/TEORIA
21 ejemploCarpetaMayusculas/IMAGES/TEORIA/A2.OUT
22 ejemploCarpetaMayusculas/IMAGES/TEORIA/HD.PNG
23 ejemploCarpetaMayusculas/IMAGES/TEORIA/RAID0.PNG
24 ejemploCarpetaMayusculas/IMAGES/TEORIA/CENTOS.PNG
25 ejemploCarpetaMayusculas/IMAGES/TEORIA/DEBIAN.PNG
26 ejemploCarpetaMayusculas/IMAGES/TEORIA/FEDORA.PNG
27 ejemploCarpetaMayusculas/D1
28 ejemploCarpetaMayusculas/D1/D4
29 ejemploCarpetaMayusculas/D1/D4/LIBSYSFS.SO.2
30 ejemploCarpetaMayusculas/D1/D4/ACT_FUNC.C
31 ejemploCarpetaMayusculas/D1/D4/BCACHE.CPP
32 ejemploCarpetaMayusculas/D1/SMO_ROUTINE.C
33 ejemploCarpetaMayusculas/D1/COMPAT.H
34 ejemploCarpetaMayusculas/D1/ECOC.H
35 ejemploCarpetaMayusculas/D1/SVM_MODEL_MATLAB.C
36 ejemploCarpetaMayusculas/D1/LIBHANDLE.SO.1.0.3
37 ejemploCarpetaMayusculas/D1/AUDIT.H
38 ejemploCarpetaMayusculas/D1/AUTO_FS.H
39 ejemploCarpetaMayusculas/D3
40 ejemploCarpetaMayusculas/D3/EJE2
41 ejemploCarpetaMayusculas/D3/DISCO2.PNG

```

## 4. ejercicio4.sh

Realizar un *script* que permita cifrar ficheros utilizando `gpg`. El *script* recibirá un argumento y, según su contenido, funcionará de dos formas:

- Si el argumento es un fichero, cifraremos solo ese fichero.
- Si el argumento es una carpeta, comprimiaremos primero la carpeta en un fichero `.tar.gz` y luego cifraremos el fichero comprimido resultante<sup>4</sup>.

Para cifrar el fichero usaremos el cifrado simétrico ofrecido por el comando `gpg` (utilizar la opción `--symmetric`)<sup>5</sup>. El archivo cifrado tendrá como nombre `fichero.gpg`, donde `fichero` es el nombre original del fichero. El comando `gpg` también permite descifrar los ficheros.

Un ejemplo de ejecución es el siguiente:

```

1 i02gupep@NEWS:~/pas/1516$ echo "probando..." > f1.txt
2 i02gupep@NEWS:~/pas/1516$ ./ejercicio4.sh f1.txt
3 Procesando un fichero
4 Fichero cifrado resultante: f1.txt.gpg
5 i02gupep@NEWS:~/pas/1516$ ls f1.txt.gpg
6 f1.txt.gpg
7 i02gupep@NEWS:~/pas/1516$ cat f1.txt.gpg
8 Caracteres irreconocibles
9 i02gupep@NEWS:~/pas/1516$ gpg -d f1.txt.gpg
10 gpg: datos cifrados CAST5
11 gpg: cifrado con 1 frase contraseña

```

<sup>4</sup>`tar zcvf fichero.tar.gz directorio`

<sup>5</sup>Consulta el enlace <https://www.gnupg.org/gph/es/manual/book1.html> y, en concreto, <https://www.gnupg.org/gph/es/manual/x129.html> para obtener más información sobre el cifrado de ficheros usando la suite *Gnu Privacy Guard* (`gpg`). Un tutorial bastante completo se encuentra en <http://goo.gl/KTYych>.

```

12 | probando...
13 | gpg: ATENCIÓN: la integridad del mensaje no está protegida
14 | i02gupep@NEWS:~/pas/1516$ ./ejercicio4.sh ejemploCarpeta/images
15 | Procesando un directorio
16 | Fichero cifrado resultante: images.tar.gz.gpg
17 | i02gupep@NEWS:~/pas/1516$ mkdir tmp
18 | i02gupep@NEWS:~/pas/1516$ cd tmp
19 | i02gupep@NEWS:~/pas/1516/tmp$ gpg -d ../images.tar.gz.gpg | tar xzv
20 | gpg: datos cifrados CAST5
21 | gpg: cifrado con 1 frase contraseña
22 | gpg: ATENCIÓN: la integridad del mensaje no está protegida
23 | ejemploCarpeta/images/
24 | ejemploCarpeta/images/Carpeta1/
25 | ejemploCarpeta/images/Carpeta1/elinfiernoexiste.jpg
26 | ejemploCarpeta/images/Carpeta1/Linus_torvalds.jpg
27 | ejemploCarpeta/images/Carpeta1/richard-stallman.jpg
28 | ejemploCarpeta/images/Carpeta1/pirates.jpg
29 | ejemploCarpeta/images/Carpeta1/devocion-administrador-de-sistemas.png
30 | ejemploCarpeta/images/logos/
31 | ejemploCarpeta/images/logos/logo_eps_grande.jpg
32 | ejemploCarpeta/images/logos/logo_uco.gif
33 | ejemploCarpeta/images/Teoria/
34 | ejemploCarpeta/images/Teoria/a2.out
35 | ejemploCarpeta/images/Teoria/hd.png
36 | ejemploCarpeta/images/Teoria/centos.png
37 | ejemploCarpeta/images/Teoria/debian.png
38 | ejemploCarpeta/images/Teoria/fedora.png
39 | ejemploCarpeta/images/Teoria/raid0.png

```

## 5. ejercicio5.sh

Realizar un *script* que averigüe la contraseña de un resumen *hash* generado con el algoritmo SHA1. El *script* recibirá dos argumentos, el resumen de la contraseña y el número de caracteres de la misma (que será siempre 1, 2 o 3 caracteres). Supondremos que todos los caracteres son alfabéticos en minúscula de la “a” a la “z”. Tendrás que hacer uso del comando `shasum`. Investiga acerca de la característica *brace expansion* de `bash`, que te puede facilitar la labor<sup>6</sup>. Un ejemplo de ejecución es el siguiente:

```

1 | i02gupep@NEWS:~/pas/1516$ echo gi | shasum
2 | 998e7c1af061a94aa08e1b252c70dff1ce887c36 -
3 | i02gupep@NEWS:~/pas/1516$ ./ejercicio5.sh 998e7c1af061a94aa08e1b252c70dff1ce887c36 2
4 | Buscando contraseñas de 2 caracteres...
5 | Encontrada la clave: gi
6 | i02gupep@NEWS:~/pas/1516$ echo gii | shasum
7 | 6be14592a05d8ac214b5ae3773aff9cba2bbbb97 -
8 | i02gupep@NEWS:~/pas/1516$ ./ejercicio5.sh 6be14592a05d8ac214b5ae3773aff9cba2bbbb97 2
9 | Buscando contraseñas de 2 caracteres...
10 | i02gupep@NEWS:~/pas/1516$ ./ejercicio5.sh 6be14592a05d8ac214b5ae3773aff9cba2bbbb97 3
11 | Buscando contraseñas de 3 caracteres...
12 | Encontrada la clave: gii

```

En el fichero de peculiaridades, incluir cuál es la contraseña que genera el siguiente resumen (obtenida a partir del *script* desarrollado):

```
d3902e622ed64cea963844de17fceb3926e132ce
```

## 6. ejercicio6.sh

Crear un *script* que “recolecte” ficheros conocidos en un conjunto de subcarpetas. El *script* recibirá un número de argumentos indeterminado, de manera que los ficheros se buscarán en cada una de las carpetas que se pasen como argumento en la línea de comandos (sino se especifica ningún argumento, se tomará la carpeta actual). Para cada carpeta, se buscarán ficheros de los siguientes tipos:

- Ficheros ejecutables.

<sup>6</sup>Información adicional: <http://blog.desdelinux.net/viernes-terminal-bash-expansion-llaves/>

- Librerías: aquellos cuyo nombre empiece por “lib”.
- Imágenes: aquellos que tengan como extensión “.png”, “.gif” o “.jpg”.
- Ficheros de cabecera: aquellos que tengan como extensión “.h”.

El nombre de las carpetas a las que se copiarán los archivos encontrados se debe solicitar al usuario. Si el usuario no los escribe tras 5 segundos o si los deja vacíos, se utilizarán los siguientes nombres:

- Ejecutables: “\$HOME/bin”.
- Librerías: “\$HOME/lib”.
- Imágenes: “\$HOME/img”.
- Ficheros de cabecera: “\$HOME/include”.

Si las carpetas existen, advertir al usuario y preguntar si sobrescribir ficheros, cuando sea necesario<sup>7</sup>.

Todos los ficheros de imágenes deberán convertirse a pdf. Para ello, haz uso del comando `convert ficheroOriginal.png ficheroOriginal.pdf`. El fichero pdf resultante deberá ser copiado a la carpeta de imágenes.

Hacer un fichero de *log* con todas las operaciones de copiado, creación de carpetas o conversión de ficheros que se vayan produciendo (`./ejercicio6.log`). El formato de este fichero de *log* es libre, pero debe contener fecha y hora para cada operación, origen y destino para las operaciones de copiado, nombre de las carpetas creadas y nombre de los ficheros convertidos. Al finalizar la ejecución del *script* se debe mostrar:

- Número de directorios procesados.
- Número de ficheros copiados para cada una de las distintas categorías (si el fichero no se sobrescribió no debe contarse).
- Tiempo necesario para realizar todas las operaciones (en segundos).

Un ejemplo de ejecución es el siguiente:

```

1 i02gupep@NEWS:~/pas/1516$ ./ejercicio6.sh ejemploCarpeta/
2 Introduzca el directorio donde copiar los ejecutables:
3
4 Introduzca el directorio donde copiar las librerías:
5
6 Introduzca el directorio donde copiar las imágenes:
7
8 Introduzca el directorio donde copiar las cabeceras:
9 Utilizando los ficheros:
10 /home/i02gupep/bin para almacenar los ficheros ejecutables
11 /home/i02gupep/lib para almacenar las librerías
12 /home/i02gupep/img para almacenar las imágenes
13 /home/i02gupep/include para almacenar los ficheros de cabeceras
14 Procesando directorio ejemploCarpeta/...
15 Número de directorios procesados: 1
16 Número de ficheros ejecutables: 3
17 Número de librerías: 3
18 Número de imágenes: 26
19 Número de ficheros de cabecera: 4
20 Tiempo necesario: 4
21
22 i02gupep@NEWS:~/pas/1516$ ls ~/bin/ ~/img/ ~/include ~/lib
23 /home/i02gupep/bin/:
24 a2.out a.out eje2
25 /home/i02gupep/img/:
26 centos.pdf devocion-administrador-de-sistemas.pdf elinfiernoexiste.jpg hd.pdf
   logo_eps_grande.jpg pirates.jpg richard-stallman.jpg

```

<sup>7</sup>Es obligatorio hacer una función que dado un fichero origen y un fichero destino, compruebe si el destino existe, pregunte si el fichero se debe sobrescribir y opere en consecuencia.

```

27 | centos.png devocion-administrador-de-sistemas.png elinfiernoexiste.pdf hd.png
    | logo_eps_grande.pdf pirates.pdf richard-stallman.pdf
28 | debian.pdf disco2.pdf                      fedora.pdf          Linus_torvalds.jpg
    | Logo_uco.gif                          raid0.pdf
29 | debian.png disco2.png                      fedora.png          Linus_torvalds.pdf
    | Logo_uco.pdf                          raid0.png
30 | /home/i02gupep/include:
31 | audit.h auto_fs.h compat.h ecoc.h
32 | /home/i02gupep/lib:
33 | libglib-2.0.so.0 libhandle.so.1.0.3 libsysfs.so.2
34 |
35 | i02gupep@NEWTS:~/pas/1516$ ./ejercicio6.sh ejemploCarpeta/dl/ ejemploCarpeta/images/
36 | Introduzca el directorio donde copiar los ejecutables: eee
37 |
38 | Introduzca el directorio donde copiar las librerías: lll
39 |
40 | Introduzca el directorio donde copiar las imágenes:
41 | Introduzca el directorio donde copiar las cabeceras:
42 | Utilizando los ficheros:
43 | eee para almacenar los ficheros ejecutables
44 | lll para almacenar las librerías
45 | /home/i02gupep/img para almacenar las imágenes
46 | /home/i02gupep/include para almacenar los ficheros de cabeceras
47 | El directorio /home/i02gupep/img ya existía
48 | El directorio /home/i02gupep/include ya existía
49 | Procesando directorio ejemploCarpeta/dl/...
50 | El fichero /home/i02gupep/include/compat.h existe. ¿Desea sobrescribirlo? (S/N): S
51 | El fichero /home/i02gupep/include/ecoc.h existe. ¿Desea sobrescribirlo? (S/N): S
52 | El fichero /home/i02gupep/include/audit.h existe. ¿Desea sobrescribirlo? (S/N): S
53 | El fichero /home/i02gupep/include/auto_fs.h existe. ¿Desea sobrescribirlo? (S/N): S
54 | Procesando directorio ejemploCarpeta/images/...
55 | El fichero /home/i02gupep/img/devocion-administrador-de-sistemas.png existe. ¿Desea sobrescribirlo? (
    | S/N): S
56 | El fichero /home/i02gupep/img/devocion-administrador-de-sistemas.pdf existe. ¿Desea sobrescribirlo? (
    | S/N): N
57 | El fichero /home/i02gupep/img/hd.png existe. ¿Desea sobrescribirlo? (S/N): S
58 | El fichero /home/i02gupep/img/hd.pdf existe. ¿Desea sobrescribirlo? (S/N): N
59 | El fichero /home/i02gupep/img/centos.png existe. ¿Desea sobrescribirlo? (S/N): S
60 | El fichero /home/i02gupep/img/centos.pdf existe. ¿Desea sobrescribirlo? (S/N): N
61 | El fichero /home/i02gupep/img/debian.png existe. ¿Desea sobrescribirlo? (S/N): S
62 | El fichero /home/i02gupep/img/debian.pdf existe. ¿Desea sobrescribirlo? (S/N): N
63 | El fichero /home/i02gupep/img/fedora.png existe. ¿Desea sobrescribirlo? (S/N): S
64 | El fichero /home/i02gupep/img/fedora.pdf existe. ¿Desea sobrescribirlo? (S/N): N
65 | El fichero /home/i02gupep/img/raid0.png existe. ¿Desea sobrescribirlo? (S/N): S
66 | El fichero /home/i02gupep/img/raid0.pdf existe. ¿Desea sobrescribirlo? (S/N): N
67 | El fichero /home/i02gupep/img/elinfiernoexiste.jpg existe. ¿Desea sobrescribirlo? (S/N): S
68 | El fichero /home/i02gupep/img/elinfiernoexiste.pdf existe. ¿Desea sobrescribirlo? (S/N): N
69 | El fichero /home/i02gupep/img/Linus_torvalds.jpg existe. ¿Desea sobrescribirlo? (S/N): S
70 | El fichero /home/i02gupep/img/Linus_torvalds.pdf existe. ¿Desea sobrescribirlo? (S/N): N
71 | El fichero /home/i02gupep/img/richard-stallman.jpg existe. ¿Desea sobrescribirlo? (S/N): S
72 | El fichero /home/i02gupep/img/richard-stallman.pdf existe. ¿Desea sobrescribirlo? (S/N): N
73 | El fichero /home/i02gupep/img/pirates.jpg existe. ¿Desea sobrescribirlo? (S/N): S
74 | El fichero /home/i02gupep/img/pirates.pdf existe. ¿Desea sobrescribirlo? (S/N): N
75 | El fichero /home/i02gupep/img/logo_eps_grande.jpg existe. ¿Desea sobrescribirlo? (S/N): S
76 | El fichero /home/i02gupep/img/logo_eps_grande.pdf existe. ¿Desea sobrescribirlo? (S/N): N
77 | El fichero /home/i02gupep/img/Logo_uco.gif existe. ¿Desea sobrescribirlo? (S/N): S
78 | El fichero /home/i02gupep/img/Logo_uco.pdf existe. ¿Desea sobrescribirlo? (S/N): N
79 | Número de directorios procesados: 2
80 | Número de ficheros ejecutables: 1
81 | Número de librerías: 2
82 | Número de imágenes: 12
83 | Número de ficheros de cabecera: 4
84 | Tiempo necesario: 45
85 |
86 | i02gupep@NEWTS:~/pas/1516$ tail ejercicio6.log
87 | 2016-02-28:18-04-12: Fichero ejemploCarpeta/images/Teoria/debian.png convertido a debian.pdf
88 | 2016-02-28:18-04-13: Fichero ejemploCarpeta/images/Teoria/fedora.png copiado a /home/i02gupep/img
89 | 2016-02-28:18-04-14: Fichero ejemploCarpeta/images/Teoria/raid0.png copiado a /home/i02gupep/img
90 | 2016-02-28:18-04-15: Fichero ejemploCarpeta/images/Carpeta1/elinfiernoexiste.jpg copiado a /home/
    | i02gupep/img
91 | 2016-02-28:18-04-17: Fichero ejemploCarpeta/images/Carpeta1/Linus_torvalds.jpg copiado a /home/
    | i02gupep/img
92 | 2016-02-28:18-04-18: Fichero ejemploCarpeta/images/Carpeta1/richard-stallman.jpg copiado a /home/
    | i02gupep/img
93 | 2016-02-28:18-04-20: Fichero ejemploCarpeta/images/Carpeta1/pirates.jpg copiado a /home/i02gupep/img
94 | 2016-02-28:18-04-21: Fichero ejemploCarpeta/images/logos/logo_eps_grande.jpg copiado a /home/i02gupep
    | /img
95 | 2016-02-28:18-04-23: Fichero ejemploCarpeta/images/logos/Logo_uco.gif copiado a /home/i02gupep/img
96 | 2016-02-28:18-04-24: Fichero ejemploCarpeta/images/Teoria/a2.out copiado a eee

```