

Programación y Administración de Sistemas

Práctica 1. Programación de *shell scripts*

Convocatoria de junio (curso 2024/2025)

Javier Sánchez Monedero
Pedro A. Gutiérrez

4 de marzo de 2025

Resumen

Esta serie de ejercicios se os entregan para que podáis practicar y profundicéis vuestros conocimientos de *bash* de cara al examen de prácticas. Estos ejercicios no se entregan, la evaluación de la práctica 1 se realizará mediante ejercicios similares a los expuestos en este guion. Para evitar problemas al ejecutar tus ejercicios de cara al examen, asegúrate de que todos los *scripts* que realices funcionen correctamente en los ordenadores de la UCO o conectándote por *ssh* al `ts.uco.es`. Para cualquier duda de los ejercicios, por favor, escribid preferentemente en el foro del moodle o enviad un correo al profesor de esta práctica `jsanchezm@uco.es`.

1. `ejercicio1.sh`

Se dispone de un directorio que contiene una serie de ficheros de texto que almacenan la asistencia de una serie de alumnos a las clases prácticas de la asignatura. El nombre de los ficheros se corresponde con el nombre de cada alumno y cada línea del mismo contiene la asistencia a una de las sesiones, siendo 0 si no ha asistido o 1 en caso de que sí haya asistido.

Desarrolla un *script* que permita calcular el número de alumnos que han asistido a cada una de las clases prácticas de la asignatura. Se debe asumir que todos los ficheros tendrán el mismo número de líneas. En caso contrario, finalizará la ejecución e indicará que hay un error en alguno de los ficheros. Para obtener el número de asistentes por sesión, el *script* deberá sumar por separado cada una de las líneas de todos los ficheros, de manera que se obtenga el total de alumnos que han asistido a cada sesión. Por ejemplo, el resultado de sumar la primera línea de cada fichero, representará la asistencia a la sesión 1. El *script* recibirá un único argumento que será el directorio donde se encuentran los ficheros.

A continuación, se muestra un ejemplo de ejecución del *script* utilizando la carpeta de ejemplo proporcionada.

```
1 i42vayuv@VTS3:~/PAS$ ./ejercicio1.sh
2 Uso: ./ejercicio1.sh <directorio>
3
4 i42vayuv@VTS3:~/PAS$ ./ejercicio1.sh FicherosEjercicio1
5 Asistieron 4 personas a la sesión 1
6 Asistieron 5 personas a la sesión 2
7 Asistieron 6 personas a la sesión 3
8 Asistieron 4 personas a la sesión 4
9 Asistieron 6 personas a la sesión 5
10 Asistieron 4 personas a la sesión 6
11 Asistieron 6 personas a la sesión 7
12 Asistieron 7 personas a la sesión 8
13 Asistieron 6 personas a la sesión 9
14 Asistieron 6 personas a la sesión 10
```

Nota: para probar el funcionamiento del *script*, puedes utilizar la carpeta `FicherosEjercicio1` que se proporciona en Moodle.

2. ejercicio2.sh

Desarrolla un *script* que permita realizar una copia de seguridad de un determinado directorio, archivándola en un fichero `tar`. El *script* permitirá la opción de archivar y comprimir o solo archivar el fichero. Es decir, si se indica que solo se archive, se creará un fichero `.tar`, mientras que si se indica que se comprima, se creará un fichero `.tar.gz`. Además, el *script* también permitirá indicar si se debe sobrescribir la copia en caso de que ya exista el fichero de destino. Por lo tanto, el *script* deberá recibir cuatro argumentos:

1. Directorio que se va a copiar.
2. Directorio donde se almacenará la copia comprimida. Si no existe, se deberá crear.
3. Valor 0 o 1 que indicará si la copia debe comprimirse o solo archiversse.
4. Valor 0 o 1 que indicará si la copia se deberá sobrescribir en caso de que ya exista.

El nombre del fichero de copia resultante deberá seguir el formato:

`nombredirectoriooriginal_usuario_fecha.tar[.gz]`, donde `usuario`, es el nombre del usuario que está realizando la copia y `fecha` es la fecha actual en formato AñoMesDía (por ejemplo: 20250115). La extensión deberá ser `.tar` cuando solo se archive o `.tar.gz` cuando también se comprima.

Por ejemplo, si el usuario `i42vayuv` hace una copia del directorio `FicherosPractical` el día 15 de enero de 2025, con la opción de solo archivar, el fichero se llamará `FicherosPractical_i42vayuv_20250115.tar`.

Para crear el fichero de copia, deberás utilizar la herramienta `tar`. Consulta los argumentos necesarios para archivar o para comprimir un fichero.

Recuerda realizar los controles de errores oportunos, incluyendo la comprobación de los argumentos de entrada del *script*.

A continuación se muestra un ejemplo de ejecución del *script*:

```
1 i42vayuv@VTS3:~/PAS$ ./ejercicio2.sh
2 Argumentos incorrectos. Uso: ./ejercicio2.sh <directorio_origen> <directorio_destino> <compresion> <
  sobrescribir>
3
4 i42vayuv@VTS3:~/PAS$ ./ejercicio2.sh FicherosEjercicio1/ backup 0 0
5 Copia realizada en backup/FicherosEjercicio1_i42vayuv_20250216.tar.
6
7 i42vayuv@VTS3:~/PAS$ ./ejercicio2.sh FicherosEjercicio1/ backup 0 0
8 Ya se ha realizado esta copia hoy (backup/FicherosEjercicio1_i42vayuv_20250216.tar).
9 No se sobrescribirá la copia.
10
11 i42vayuv@VTS3:~/PAS$ ./ejercicio2.sh FicherosEjercicio1/ backup 1 0
12 Copia realizada en backup/FicherosEjercicio1_i42vayuv_20250216.tar.gz.
13
14 i42vayuv@VTS3:~/PAS$ ./ejercicio2.sh FicherosEjercicio1/ backup 1 0
15 Ya se ha realizado esta copia hoy (backup/FicherosEjercicio1_i42vayuv_20250216.tar.gz).
16 No se sobrescribirá la copia.
17
18 i42vayuv@VTS3:~/PAS$ ./ejercicio2.sh FicherosEjercicio1/ backup 1 1
19 Ya se ha realizado esta copia hoy (backup/FicherosEjercicio1_i42vayuv_20250216.tar.gz).
20 Copia realizada en backup/FicherosEjercicio1_i42vayuv_20250216.tar.gz.
```

3. ejercicio3.sh

Desarrolla un *script* que permita comprobar que todos los usuarios de un sistema que dispongan de una clave privada de SSH la tengan correctamente protegida. Es decir, el *script* deberá comprobar que los permisos de dicha clave estén configurados de tal forma que solo el propietario de la misma pueda acceder a su contenido. Dicha clave, suele ubicarse dentro del `home` del usuario, en una carpeta oculta llamada `.ssh`. El nombre del fichero por defecto es `id_rsa`. Para este ejercicio, puedes suponer que el fichero de clave, en caso de existir, estará en la ruta

/home/<usuario>/`.ssh/id_rsa`. Para ver si realmente el fichero está protegido, deberás tener en cuenta no solo los permisos del propio fichero, sino también los de la carpeta `.ssh` y del propio /home/<usuario>.

En el caso de encontrar algún usuario que tenga un fichero de clave desprotegido, el *script* deberá mostrar un mensaje y, además, deberá crear un fichero en el escritorio de dicho usuario (/home/<usuario>/Desktop) para advertirle de ello.

Para probar el *script* desarrollado, se proporciona en Moodle otro *script* que creará una estructura de directorios con diferentes permisos. Puedes utilizarlo, o bien crear la carpeta de ejemplo manualmente.

A continuación se muestra un ejemplo de ejecución del *script* desarrollado con la carpeta de ejemplo generada por el *script* de Moodle:

```
1 i42vayuv@VTS3:~/PAS$ ./generar_ficheros_ejercicio3.sh
2
3 i42vayuv@VTS3:~/PAS$ ./ejercicio3.sh FicherosEjercicio3/home
4 El usuario eva tiene una clave privada de ssh en FicherosEjercicio3/home/eva/.ssh/id_rsa que no está
   protegida. La clave debe ser accesible únicamente por el propietario.
5 El usuario javi tiene una clave privada de ssh en FicherosEjercicio3/home/javi/.ssh/id_rsa que no est
   á protegida. La clave debe ser accesible únicamente por el propietario.
```

Nota: para consultar los permisos de un fichero o directorio puedes usar el comando `stat`. Investiga qué argumentos debes utilizar. Adicionalmente, puedes usar el comando `cut` si necesitas extraer una parte concreta del resultado del comando `stat`.

4. ejercicio4.sh

Desarrolla un *script* que permita generar una cadena de caracteres aleatoria de una longitud N concreta. El *script* dará la posibilidad de generar tres tipos diferentes de cadenas en función de los caracteres que podrán contener:

1. Solo letras mayúsculas o minúsculas.
2. Solo números.
3. Letras mayúsculas, minúsculas y números.

El *script* recibirá dos argumentos:

1. Longitud de la cadena (N).
2. Tipo de cadena (`alfa`, `num` o `alfanum`).

El segundo argumento será opcional, por lo que se podrá ejecutar solo introduciendo el primero o indicando ambos. En el caso de que no se introduzca el tipo de cadena que se debe generar, el *script* deberá pedirle este dato al usuario por teclado. Si el usuario introduce un valor inválido, se le deberá volver a pedir. Recuerda realizar cualquier otro control de errores que estimes oportuno.

A continuación se muestra un ejemplo de ejecución del *script*:

```
1 i42vayuv@VTS3:~/PAS$ ./ejercicio4.sh
2 Uso: ./ejercicio4.sh <longitud de la cadena> [tipo de cadena]
3
4 i42vayuv@VTS3:~/PAS$ ./ejercicio4.sh 10
5 Introduce el tipo de cadena (alfanum, alfa o num): alfanum
6 wqfyua8wKl
7
8 i42vayuv@VTS3:~/PAS$ ./ejercicio4.sh 10 alfa
9 omNwJWzWmU
10
11 i42vayuv@VTS3:~/PAS$ ./ejercicio4.sh 30
12 Introduce el tipo de cadena (alfanum, alfa o num): num
13 205361320090006836059156471288
```

5. ejercicio5.sh

Crea un *script* que reciba como único argumento la ruta de un directorio. El *script* deberá mostrar de forma recursiva todos los archivos que se encuentren dentro del directorio especificado. Es decir, deberán mostrarse los ficheros que estén inmediatamente dentro de dicho directorio, pero también aquellos que se encuentren dentro de algún subdirectorio. Sin embargo, los subdirectorios no deberán mostrarse. Por cada fichero, se deberá mostrar:

1. El nombre del fichero sin la ruta
2. La ruta absoluta del fichero.
3. La fecha de modificación expresada en segundos desde la Época.
4. Tamaño en bytes
5. La cadena de permisos del fichero

La información de cada fichero se mostrará en una fila y cada campo estará separado por un tabulador. La lista de ficheros deberá estar ordenada ascendentemente por la fecha de modificación. El *script* deberá recibir un único argumento que será el directorio a listar. Si el directorio no existe, se deberá mostrar un mensaje de error y finalizar el *script*.

A continuación se muestra un ejemplo de ejecución del *script*:

```
1 i42vayuv@VTS3:~/PAS$ ./ejercicio5.sh
2 Uso: ./ejercicio5.sh <directorio>
3
4 i42vayuv@VTS3:~/PAS$ ./ejercicio5.sh FicherosEjercicio3/
5 lsU1RWJ9Mw      /home/i42vayuv/PAS/FicherosEjercicio3/home/victor/Desktop/lsU1RWJ9Mw      1708079367
6 0 bytes      -rw-----
7 55v4SE0n2R      /home/i42vayuv/PAS/FicherosEjercicio3/home/isa/55v4SE0n2R      1708079367
8 0 bytes      -rw-----
9 5slWb12pwG      /home/i42vayuv/PAS/FicherosEjercicio3/home/victor/5slWb12pwG      1708079367
10 0 bytes      -rw-----
11 61Hi0woJtP      /home/i42vayuv/PAS/FicherosEjercicio3/home/eva/Desktop/61Hi0woJtP      1708079367
12 0 bytes      -rw-----
13 63vG5ilRgr      /home/i42vayuv/PAS/FicherosEjercicio3/home/isa/Desktop/63vG5ilRgr      1708079367
14 0 bytes      -rw-----
15 ...
16 id_rsa          /home/i42vayuv/PAS/FicherosEjercicio3/home/isa/.ssh/id_rsa      1708079367
17 128 bytes      -rw-----
18 id_rsa          /home/i42vayuv/PAS/FicherosEjercicio3/home/jose/.ssh/id_rsa      1708079367
19 128 bytes      -rw-----
20 id_rsa          /home/i42vayuv/PAS/FicherosEjercicio3/home/eva/.ssh/id_rsa      1708079367
21 128 bytes      -rwxrwxr--
22 id_rsa          /home/i42vayuv/PAS/FicherosEjercicio3/home/javi/.ssh/id_rsa      1708079367
23 128 bytes      -rwxrwxr--
24 id_rsa          /home/i42vayuv/PAS/FicherosEjercicio3/home/maria/.ssh/id_rsa      1708079367
25 128 bytes      -rwxrwxr--
26 id_rsa          /home/i42vayuv/PAS/FicherosEjercicio3/home/victor/.ssh/id_rsa      1708079367
27 128 bytes      -rwxrwxr--
```

6. ejercicio6.sh

Escribe una función en bash que permita generar un fichero de índice para un directorio que reciba como parámetro. Dicho índice, deberá contener en, formato HTML, un listado con los ficheros y directorios que contenga. En HTML, una lista se puede especificar mediante las etiquetas `` (apertura) y `` (cierre). Los elementos de la lista se indican entre la etiqueta de apertura y la de cierre utilizando una etiqueta `` por cada elemento. Además, los elementos asociados con directorios deberán tener un enlace que permita navegar hacia el índice de dicho directorio. Para ello, se puede utilizar la etiqueta `<a>`, como se muestra en el siguiente ejemplo:

```

1 <ul>
2   <li>fichero1</li>
3   <li>fichero2</li>
4   <li><a href="directorio1/index.html">directorio1</a></li>
5 </ul>

```

Una vez implementada dicha función, crea un *script* que reciba como único argumento la ruta de un directorio existente y genere, utilizando la función implementada, los índices de dicho directorio y de todos los subdirectorios que contenga, de forma recursiva. Recuerda realizar los controles de errores que estimes oportunos.

A continuación se muestra un ejemplo de ejecución del *script*:

```

1 i42vayuv@VTS2:~/PAS$ ./ejercicio6.sh
2 Uso: ./ejercicio6.sh <directorio>
3 i42vayuv@VTS2:~/PAS$ ./ejercicio6.sh FicherosEjercicio3/
4 Se ha creado el fichero FicherosEjercicio3//index.html con el contenido del directorio
5 FicherosEjercicio3/.
6 Se ha creado el fichero FicherosEjercicio3/home/index.html con el contenido del directorio
7 FicherosEjercicio3/home.
8 Se ha creado el fichero FicherosEjercicio3/home/victor/index.html con el contenido del directorio
9 FicherosEjercicio3/home/victor.
10 Se ha creado el fichero FicherosEjercicio3/home/victor/Desktop/index.html con el contenido del
11 directorio FicherosEjercicio3/home/victor/Desktop.
12 Se ha creado el fichero FicherosEjercicio3/home/victor/.ssh/index.html con el contenido del
13 directorio FicherosEjercicio3/home/victor/.ssh.
14 ...
15 Se ha creado el fichero FicherosEjercicio3/home/isa/index.html con el contenido del directorio
16 FicherosEjercicio3/home/isa.
17 Se ha creado el fichero FicherosEjercicio3/home/isa/Desktop/index.html con el contenido del
18 directorio FicherosEjercicio3/home/isa/Desktop.
19 Se ha creado el fichero FicherosEjercicio3/home/isa/.ssh/index.html con el contenido del directorio
20 FicherosEjercicio3/home/isa/.ssh.
21 Se ha creado el fichero FicherosEjercicio3/home/eva/index.html con el contenido del directorio
22 FicherosEjercicio3/home/eva.
23 Se ha creado el fichero FicherosEjercicio3/home/eva/Desktop/index.html con el contenido del
24 directorio FicherosEjercicio3/home/eva/Desktop.
25 Se ha creado el fichero FicherosEjercicio3/home/eva/.ssh/index.html con el contenido del directorio
26 FicherosEjercicio3/home/eva/.ssh.

```

La Figura 1 muestra un ejemplo de cómo se visualiza uno de los ficheros de índice en el navegador.

Contenido de FicherosEjercicio3/home

- [eva](#)
- [index.html](#)
- [isa](#)
- [javi](#)
- [jose](#)
- [maria](#)
- [victor](#)

Figura 1: Ejemplo de fichero de índice generado por el *script* del ejercicio 6.

A. Conexión en remoto a la UCO

Para poder trabajar en remoto en el servidor de la UCO, necesitaremos, por un lado, conectarnos a una sesión de *ssh* para poder tener una terminal remota y, por otro lado, conectarnos

por `sftp` para poder transferir archivos.

A.1. Conexión SSH

A.1.1. Linux

Para conectarnos por `ssh` desde Linux, basta con abrir una terminal y escribir:

```
1 ssh usuarioUCO@ts.uco.es
```

En caso de que no tengamos el cliente de `ssh` instalado, deberemos instalarlo con

```
1 apt install openssh-client
```

A.1.2. Windows 10+

En primer lugar es necesario habilitar el cliente `ssh`, que viene deshabilitado por defecto. Para ello, puedes seguir estas instrucciones ¹. Una vez habilitado, basta con abrir una ventana de `cmd` y escribir el comando:

```
1 ssh usuarioUCO@ts.uco.es
```

A.2. Conexión SFTP

A.2.1. Linux

Para conectarnos por `sftp` desde Linux, basta con abrir una ventana del explorador de archivos y en la barra de direcciones escribir:

```
1 sftp://usuarioUCO@ts.uco.es/home/usuarioUCO
```

A.2.2. Windows 10+

En el caso de Windows deberemos instalar algún cliente de `sftp` como por ejemplo WinSCP. El usuario de conexión será nuestro usuario de la UCO y la dirección del servidor `ts.uco.es`.

¹<https://www.howtogeek.com/336775/how-to-enable-and-use-windows-10s-built-in-ssh-commands/>