

# Administración de Sistemas. Práctica 2

## Introducción

NOTA: Para realizar esta práctica no se requieren privilegios de administración. Se puede realizar en cualquier máquina con *bash*. Si requieres de una máquina virtual para realizarlos puedes descargar una desde:

[https://unizares-my.sharepoint.com/:f/g/personal/gvalles\\_unizar\\_es/EgGWxUUEN8ZlpUg6zgEQpDIBLfxCBi5jG\\_t\\_CJWCrdxA-g](https://unizares-my.sharepoint.com/:f/g/personal/gvalles_unizar_es/EgGWxUUEN8ZlpUg6zgEQpDIBLfxCBi5jG_t_CJWCrdxA-g)

Esta tiene los usuarios **root** y **as** con password **toor** y **as**, respectivamente.

La entrega de la práctica se realizará según las normas descritas al final de este documento. Se deben entregar únicamente tantos archivos como scripts tiene la práctica empaquetados en un único fichero tar.gz, solo archivos que sean scripts (ningún otro tipo de archivo), nombrados de la siguiente forma:

practica2\_1.sh.....practica2\_2.sh.....practica2\_6.sh

Cada script debe incluir un comentario en la cabecera del fichero con el nombre del autor o autores, así como su NIP. Además, en los mensajes escritos por pantalla, los scripts no emplearán acentos. En esta práctica no hay que entregar ningún fichero de texto adicional.

Esta práctica será corregida tanto manual como automáticamente. Para ayudarte en la realización de la práctica, dispones de unos test unitarios llamados test\_practica2\_[123456].py. El objetivo de su utilización es detectar posibles errores y no asumir la corrección de los scripts por pasar los tests.

Puedes descargar los tests de:

[https://github.com/universidad-zaragoza/tests\\_practicas\\_AS](https://github.com/universidad-zaragoza/tests_practicas_AS)

mediante el comando:

```
$> git config --global http.sslverify false
$> git clone https://github.com/universidad-zaragoza/tests_practicas_AS
```

Tal y como se indica en el fichero README.md, el repositorio está estructurado para **tener un directorio** por práctica y los tests en su directorio propio. Estos tests te servirán para asegurarte de que tus prácticas están listas para ser sometidas, aunque no tienen por qué cubrir todos los casos de error posibles. De hecho, si se te ocurre alguno te invitamos a añadirlo. Si es necesario instalar git en la máquina puedes hacerlo ejecutando:

```
$> sudo apt install git
```

En <https://git-scm.com/book/en/v2>, dispones de un manual muy completo de git en caso de que quieras consultarlo y en <https://docs.gitlab.com/> puedes documentarte sobre gitlab.

Los tests están escritos en Python por lo que para utilizarlos en la máquina virtual de la asignatura, tal y como se recomienda, será necesario que instales pip para instalar paquetes de Python y luego el módulo de Python pexpect ya que es utilizado por los tests. Puedes instalar ambos con los siguientes comandos:

```
$> sudo apt update && sudo apt install python-pip
$> pip install pexpect
```

Para cada uno de los scripts a realizar practica2\_[1-6].sh dispones de un test homónimo test\_practica2\_[1-6].py. En algunos casos, los tests comprueban que la salida del script sea igual a la mostrada en este documento, con lo que por favor deberás asegurar que vuestro formato de salida es idéntico al de los ejemplos.

En caso de encontrar algún error en los tests o tener alguna sugerencia de mejora, por favor notificar a alguno de los profesores de la signatura.

Todos los scripts entregados deben incluir al principio el nombre del autor o autores, su NIPs y grupo de acuerdo con el siguiente formato:

```
#!/bin/bash
#NIP1, Apellidos, Nombre, [MT], [1-5], [AB]
#NIP2, Apellidos, Nombre, [MT], [1-5], [AB]
```

[MT] se refiere al grupo de Mañana o de Tarde [1-5] se refiere al día de la semana siendo el lunes el día 1 [AB] se refiere a si es semana A o B en la EINA. En el repositorio de prácticas *as\_tests\_practicas* está disponible el script *check\_header.sh* para poder comprobar fácilmente si una cabecera es correcta.

Notas: este guión utiliza la sintaxis <...> para referirse a entradas del script que han sido escritas por el usuario.

## Programación en *Shell*

En la construcción de los siguientes programas o 'scripts' se valorará la presentación: importa el formato, los comentarios, la claridad de los comandos, la significación de los nombres elegidos; todo esto debe permitir considerar al programa como 'autodocumentado'.

Para todos ellos se ha de suponer que al ejecutarlos os encontráis en un directorio desconocido del sistema (es decir, no podéis suponer que estáis en un directorio en particular), y deben funcionar para cualquier usuario del sistema.

1. Escribir un script que pida un nombre de archivo al usuario mediante la cadena: "Introduzca el nombre del fichero: ", e indique si el archivo es legible, modificable y ejecutable por el usuario. La salida del script será la siguiente según los posibles casos:
  - Fichero no existe: "<nombre fichero> no existe"
  - Fichero existente: "Los permisos del archivo <nombre fichero> son:  
[ --- | r-- | -w- | --x | rw- | r-x | -wx | rwx ]"

Ejemplo:

```
as@as0:~$ ./practica2_1.sh
```

```
Introduzca el nombre del fichero: practica2_1.sh
```

```
Los permisos del archivo practica2_1.sh son: rwx
```

```
as@as0:~$ ./practica2_1.sh
```

```
Introduzca el nombre del fichero: practica2_7.sh
```

```
practica2_7.sh no existe
```

2. Escribir un script que reciba varios nombres de archivo como parámetros, para

cada uno valide si el nombre corresponde a un archivo común existente, y si es así lo muestre en pantalla paginando (con el comando more). [Aprovechar este apartado de la práctica para comprobar la diferencia entre el uso de \$\*, "\$\*", \$@ y "\$@" en la lectura de los parámetros de entrada]. En caso de que el parámetro de entrada no sea un fichero, el script deberá escribir por pantalla <parámetro> no es un fichero.

Ejemplo:

```
as@as0:~$./practica2_2.sh no_existe
no_existe no es un fichero
```

3. Escribir un script que reciba un único nombre de archivo en la línea de comandos, verifique que existe y que es un archivo común (o regular), lo convierta en ejecutable para el dueño y el grupo y muestre el modo final mediante el comando stat (deberás buscar el formato correcto). Si el fichero no existe, debes mostrar el mensaje de error: "<nombre\_archivo> no existe". En caso de recibir un número distinto de argumentos por la línea de comandos, el script imprimirá el siguiente mensaje de error: "Sintaxis: practica2\_3.sh <nombre\_archivo>".

Ejemplos:

```
as@as0:~$./practica2_3.sh practica2_3.sh
-rwxr-xr-x
```

```
as@as0:~$./practica2_3.sh primer_arg "segundo argumento"
Sintaxis: practica2_3.sh <nombre_archivo>
```

4. Escribir un script que nos pida pulsar una tecla escribiendo por pantalla el mensaje: "Introduzca una tecla: ", y nos diga si la tecla pulsada es una letra, un número o un carácter especial. Si el usuario pulsa más de una tecla, se ignorarán todas menos la primera letra pulsada.

Ejemplos:

```
as@as0:~$./practica2_4.sh
Introduzca una tecla: a
a es una letra
```

```
as@as0:~$./practica2_4.sh
Introduzca una tecla: 4
4 es un numero
```

```
as@as0:~$./practica2_4.sh
Introduzca una tecla: <espacio>
es un caracter especial
```

5. Escribir un script que pida introducir la ruta de un directorio por teclado y muestre cuantos archivos y cuantos directorios hay dentro de ese directorio (sin aplicar recursividad en ambos casos). En caso de que la ruta leída no se corresponda con un directorio, el script escribirá por pantalla el siguiente mensaje:

"<ruta\_leida> no es un directorio". Una vez determinados el número de ficheros y directorios, el script mostrará el siguiente mensaje: "El numero de ficheros y directorios en <dir> es de <num\_files> y <num\_dirs>, respectivamente".

Ejemplos:

```
as@as0:~$./practica2_5.sh
```

Introduzca el nombre de un directorio: <dir\_no\_existe>

<dir\_no\_existe> no es un directorio

```
as@as0:~$./practica2_5.sh
```

Introduzca el nombre de un directorio: <dir\_valido>

El numero de ficheros y directorios en <dir\_valido> es de 0 y 13, respectivamente

6. Escribir un script que copie todos los programas del directorio actual (archivos ejecutables) hacia un nuevo subdirectorio temporal cuyo nombre siga el siguiente patrón, binXXX (siendo X un carácter alfanumérico), y se encuentre en el directorio raíz del usuario. Durante la ejecución, el script mostrará los nombres de los ficheros copiados e indicará su número o que no ha movido ninguno. Si en el directorio raíz del usuario ya existieran uno o varios directorios que siguieran el patrón, hay que utilizar el que haya sido menos recientemente modificado (esta información se puede obtener mediante `stat -c %n,%Y <nombre_directorio>` que imprimirá el nombre del directorio y la fecha de última modificación). Considera la utilización el comando *mktemp* para la creación de los ficheros binXXX.

```
as@as0:~$./practica2_6.sh
```

Se ha creado el directorio /home/as/binM9Y

Directorio destino de copia: /home/as/binM9Y

./practica2\_1.sh ha sido copiado a /home/as/binM9Y

./practica2\_2.sh ha sido copiado a /home/as/binM9Y

./practica2\_3.sh ha sido copiado a /home/as/binM9Y

./practica2\_4.sh ha sido copiado a /home/as/binM9Y

./practica2\_5.sh ha sido copiado a /home/as/binM9Y

./practica2\_6.sh ha sido copiado a /home/as/binM9Y

./test\_practica2\_1.py ha sido copiado a /home/as/binM9Y

./test\_practica2\_2.py ha sido copiado a /home/as/binM9Y

./test\_practica2\_3.py ha sido copiado a /home/as/binM9Y

./test\_practica2\_4.py ha sido copiado a /home/as/binM9Y

./test\_practica2\_5.py ha sido copiado a /home/as/binM9Y

./test\_practica2\_6.py ha sido copiado a /home/as/binM9Y

Se han copiado 12 archivos

```
as@as0:~$./practica2_6.sh
```

Se ha creado el directorio /home/as/binM9Y

Directorio destino de copia: /home/as/binM9Y

No se ha copiado ningun archivo

## Consideraciones finales

Una vez completados los 6 scripts asegura por favor que su cabecera cumple el formato establecido para después empaquetarlos en un fichero comprimido tar.gz y subirlos a la entrega correspondiente de tu grupo en Moodle. El nombre del fichero deberá ser

<NIP1>\_<NIP2>.tar.gz o <NIP1>.tar.gz y para su creación puedes emplear el siguiente comando (asumiendo que los NIPS de los alumnos son 161803 y 132471):

```
as@as0:~$ tar cvzf 132471_161803.tar.gz ./practica2_{1,2,3,4,5,6}.sh
```

Recuerda que el comando `check_header.sh` del repositorio de tests permite comprobar la validez de la cabecera de un script.

Fecha de última modificación: 28 de febrero de 2022