



UNIVERSITAT_{DE}
BARCELONA

SISTEMAS OPERATIVOS II

PRÁCTICA 1

3 de octubre 2020

Autores y NIUB:

María Isabel González Sánchez 20221913

Oriol Saguillo González 20150502

Grupo: B00

Índice

1. Experimentos con el <i>Chroot jail</i>	2
1.1. Problemas y posibles fallos solucionados	2
1.2. Preguntas del primer ejercicio	2
2. Ejercicio de la jaula	3
2.1. Realización y ampliación de la Jaula	3
2.2. Testeo de las comandas bash, ls, cp y rm	4
2.3. Testeos de la jaula	5

1. Experimentos con el *Chroot jail*

1.1. Problemas y posibles fallos solucionados

En esta 1ª práctica, hemos seguido el PDF sin problemas hasta el punto de introducir la siguiente comanda en la terminal:

```
sudo chroot gabia /bin/statistics
```

Al principio, no funcionaba y nos salía el siguiente error:

```
oslab: /Escritorio/gabia$ sudo chroot gabia /bin/statistics
```

```
chroot: no se puede cambiar el directorio raíz a “gabia”: No existe el archivo o el directorio
```

Después de preguntar al profesor de prácticas, nos dimos cuenta que nos faltaban 2 directorios dentro del directorio “gabia”: *lib* y *lib64*. Una vez creados con sus respectivas librerías en el interior, repetimos la comanda desde el directorio escritorio y este es el resultado:

```
Could not open file 'file.txt'
```

Esto se debe a que nuestro archivo *file.txt* y el ejecutable *statistics* no están en el mismo directorio. Por ello, debemos especificarle el path. A partir de aquí, ampliaremos la información en el siguiente subapartado.

1.2. Preguntas del primer ejercicio

Ahora reponderemos brevemente las preguntas del PDF:

- **¿Puede la aplicación leer el fichero que hay en el interior del directorio “data”? En caso que no pueda, ¿por qué no puede?**

No, ya que el ejecutable *statistics* y el fichero *file.txt* están respectivamente en los directorio “bin” y “data”. Por lo tanto, no sabe encontrar el archivo con la información que le facilitamos al comando.

- **¿Qué valor debe tener la variable FILE en el código C para que se lea correctamente el fichero? ¿Se consigue hacer poniendo una ruta completa al fichero?**

Debería ser así: define FILE "../data/file.txt".

No haría falta una ruta completa, solo volver al directorio "gabia": mediante [...] volvemos al directorio "gabia" desde "bin".

Una vez ahí, podemos acceder a "data" y por último al archivo. Habría que recompilar el archivo C para sobrescribir el ejecutable y tener la variable FILE actualizada.

2. Ejercicio de la jaula

Para esta 2ª parte del informe, iremos paso a paso comprobando cada comando. Primero, explicaremos como hemos conseguido los caminos hacia las comandas, después como hemos comprobado su funcionamiento y, por último, testaremos los límites de la jaula.

2.1. Realización y ampliación de la Jaula

Para poder introducir las comandas que se nos piden dentro del directorio "gabia", hemos seguido los mismos pasos que en el ejercicio 1. La diferencia más notable se aprecia en como encontrar el *path* de las librerías, que se consigue mediante la comanda "which":

```
oslab: /Escritorio$ which bash /usr/bin/bash
```

Una vez obtenido el *path* de dónde se ubicaban los comandos, importamos las librerías dinámicas que necesitamos para hacer *bash*, *ls*, *cp* y *rm*. Para conocerlo, tenemos el ejemplo de *bash*:

```
oslab: /Escritorio$ ldd $(which bash)
```

```
linux-vdso.so.1 (0x00007ffc607b9000)
```

```
libtinfo.so.6 =_./lib/x86_64-linux-gnu/libtinfo.so.6 (0x00007fa20be75000)
```

```
libdl.so.2 =_./lib/x86_64-linux-gnu/libdl.so.2 (0x00007fa20be6f000)
```

```
libc.so.6 =_./lib/x86_64-linux-gnu/libc.so.6 (0x00007fa20bc7d000)
/lib64/ld-linux-x86-64.so.2 (0x00007fa20bfe2000)
```

Para el resto de *paths* se haría de la misma forma, pero cambiando *bash* por *ls*, *cp* o *rm*.

2.2. Testeo de las comandas *bash*, *ls*, *cp* y *rm*

Primero, realizamos el *chroot jail* con la comanda *bash*, en vez de “statistics”, mediante:

```
oslab: /Escritorio$ sudo chroot gabia /bin/bash
```

Una vez dentro de *bash*, miraremos qué directorios tenemos dentro del directorio principal “gabia” mediante *ls*:

```
bash-5.0# ls
bin data lib lib64
```

Nos encontraremos con 4 directorios, de los cuales, el que nos interesa es *bin*. Mediante *cd*, entraremos en este directorio y comprobaremos que el ejecutable “statistics” funciona correctamente:

```
bash-5.0# cd bin
bash-5.0# ./statistics
Summary:
Vowels: 1345529
Consonants: 2047454
Digits: 1214
Space chars: 1060666
Puntuacion chars: 196974
```

Luego nos dirigiremos al directorio “data” para comprobar el funcionamiento de la comanda *cp*:

```
bash-5.0# cd ..
bash-5.0# ls
bin data lib lib64
bash-5.0# cd data
```

```
bash-5.0# ls
```

```
file.txt
```

Mediante *cp*, crearemos un fichero copia del archivo original “file.txt”:

```
bash-5.0# cp file.txt copia.txt
```

```
bash-5.0# ls
```

```
copia.txt file.txt
```

Por último, eliminaremos la copia del directorio “data” mediante la comanda *rm* y salimos del *bash*:

```
bash-5.0# rm copia.txt
```

```
bash-5.0# ls
```

```
file.txt
```

```
bash-5.0# exit
```

```
exit
```

2.3. Testeos de la jaula

A modo de comprobación, quisimos ver si la jaula era hermética y no permitía fugas mediante comandas no introducidas. Para ello, intentamos hacer un *grep* y obviamente no lo encuentra, porque no está su librería:

```
bash-5.0# f grep -o
```

```
bash: grep: command not found
```

```
exit
```