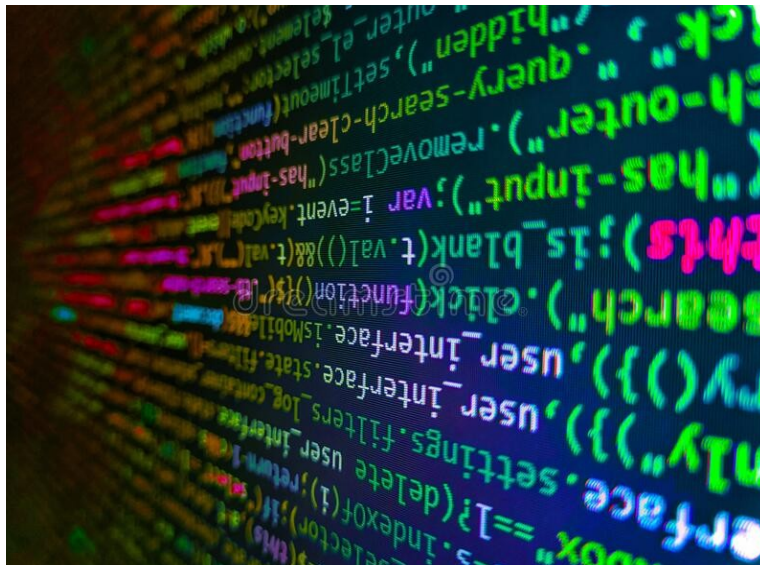


Sistemas Operativos

Práctica 2 – Comandos Para Manipular Ficheros

Semestre de Primavera 2023



Jose Maria Fernández Paredes
Víctor Sort Rubio

ÍNDICE

1. Introducción	3
2. Pruebas realizadas	3
2.1 Ejercicio 1	3
2.2 Ejercicio 2	3
2.3 Ejercicio 3	4
2.4 Ejercicio 4	4
2.5 Ejercicio 5	4
2.6 Ejercicio 6	5
3. Conclusiones y valoraciones personales	5

1. INTRODUCCIÓN

En esta segunda práctica se nos planteaban 6 problemas o ejercicios que se resolvían mediante la realización de 1 script cada uno de ellos, los cuales posteriormente se tenían que ejecutar en la terminal y requerían de argumentos por líneas de comando.

Al ya tener unos conocimientos previos de bash scripting obtenidos en la anterior práctica, la dificultad de esta incrementaba un poco y se introducían unos nuevos comandos para aprender y aplicar en los scripts (que se encuentran en el mismo zip dónde se encuentra este informe), referentes principalmente al manejo de ficheros y a la gestión de los procesos.

2. PRUEBAS REALIZADAS

Antes de ver las pruebas de cada ejercicio en detalle, comentar que en todos los ejercicios hemos hecho las comprobaciones iniciales referentes al número correcto de parámetros (viendo si nos daba error introducir más o menos parámetros que los necesarios) y las comprobaciones de si los directorios pasados por parámetro existen.

2.1 Ejercicio 1

Para probar este ejercicio, que se ha resuelto en una sola línea usando diversos comandos con tuberías, hemos comprobado que la salida sea la misma que la dada en el ejemplo usando el directorio gutenbergl y la extensión html.

También lo hemos comprobado en otro directorio dónde había menos de 5 ficheros txt, para comprobar que los listaba todos, aunque no llegaran a 5.

```
oslab:/media/sf_Carpeta_compartida> cd EjercicisP2
oslab:/...> ./ejercicio1.sh gutenbergl html
gutenberg/etext02/nb17v11h/nb17v11h.html
gutenberg/etext02/nc13v11h/nc13v11h.html
gutenberg/etext03/jj13b10h/jj13b10h.html
gutenberg/etext04/rob3w10h/rob3w10h.html
gutenberg/etext05/cfvrw10h/cook.html
oslab:/...> ./ejercicio1.sh prova_txt txt
prova_txt/text1
prova_txt/text2
prova_txt/text3
```

2.2 Ejercicio 2

Este ejercicio lo hemos hecho iterando sobre el resultado de hacer un find en el directorio dado. Para que la salida fuera cómo la dada en la muestra, añadimos un sort a la salida del find. Hemos probado el código con gutenbergl, y hemos visto que la salida es exactamente igual a la deseada.

```
oslab:/...> ./ejercicio2.sh gutenbergl
668655 gutenbergl/
33317565 gutenbergl/etext00
24478008 gutenbergl/etext01
43850780 gutenbergl/etext02
6564239 gutenbergl/etext02/andv111h    (y muchas líneas más...)
```

2.3 Ejercicio 3

Para ir probando este código creamos un fichero prova.txt con pocas palabras simples y probando diferentes casos, por ejemplo, poniendo la palabra “lala” y queriendo remplazar la cadena “la” por “el”. El código devuelve correctamente que la cadena se encuentra dos veces y al abrir el archivo vemos que se muestra “elel”, pues el código funciona correctamente aunque la cadena se encuentra repetida en la misma palabra.

Finalmente probamos el código usando la extensión txt en gutenber con las cadenas math y english, tal y como la muestra, y vimos que la salida es la misma a la esperada, así como que realmente si hacia los cambios.

```
oslab:/media/sf_Carpeta_compartida/ExercicisP2> cd gutenber
oslab:/...> ./ejercicio3.sh txt math english
./copyright.txt 0
./donate-howto.txt 0
./etext00/00ws110.txt 6
./etext00/1cahe10.txt 3
./etext00/1vkip11.txt 2      (y muchas líneas más...)
```

2.4 Ejercicio 4

Para probar este ejercicio hemos seguido el propio ejemplo, pero haciendo un paso previo. Hemos creado una copia de gutenber y hemos visto que si ejecutamos el script, nos dice que la copia tiene todos los ficheros del primer directorio. Luego, borramos un par de ficheros en la copia y vemos que si ejecutamos ahora el script, nos dice que son esos los que no están.

```
oslab:/...> cp -r gutenber gutenber_copia
oslab:/...> ./ejercicio4.sh gutenber gutenber_copia
El segundo directorio tiene todos los archivos del primero.
oslab:/...> rm gutenber_copia/etext93/tarz210.txt
oslab:/...> rm gutenber_copia/etext99/uspis10.txt
oslab:/...> ./ejercicio4.sh gutenber gutenber_copia
gutenber/etext93/tarz210.txt
gutenber/etext99/uspis10.txt
```

2.5 Ejercicio 5

En este ejercicio se han tenido que hacer comprobaciones adicionales al resto de los ejercicios. Se ha comprobado (y hemos probado que hiciera bien) que el archivo coches.csv se encontrara en el directorio del script, que el segundo parámetro fuera sólo “ITV” o “Seguro” o “IVTM” y el tercero “SI” o “NO”, y que la matrícula del coche se encontrara en el archivo.

Una vez hechas las comprobaciones de todos los posibles errores, hemos ido modificando el fichero, notando que si se le pedía modificar un dato “SI” por otro “SI”, no suponía ningún error.

```
oslab:/...> ./ejercicio5.sh 3524BZR ITV SI
Estatus de la ITV actualizado para el vehiculo 3524BZR.
oslab:/...> ./ejercicio5.sh 3524BZR ITV SI
Estatus de la ITV actualizado para el vehiculo 3524BZR.
```

2.6 Ejercicio 6

Este ejercicio también requería de la comprobación adicional de comprobar que el segundo parámetro fuera “RSS” o “VSZ”, así cómo comprobar también que el usuario entrado por parámetro tenía procesos a su nombre.

Una vez hechas esta comprobaciones, ejecutamos en el terminal el comando ps aux par poder listar todos los procesos actuales y poder comprobar correctamente y manualmente, que el resultado una vez ejecutáramos el script con diferentes usuarios y listando por VSZ o RSS sea el deseado, y así ha sido.

```
oslab:/...> ./ejercicio6.sh ruut VSZ
ERROR: ruut no tiene ningún proceso asociado
oslab:/...> ./ejercicio6.sh ruut VSZ
ERROR: El segundo argumento tiene que ser VSZ o RSS
oslab:/...> ./ejercicio6.sh root VSZ
1031824 103316 /usr/bin/X
980972 97216 /usr/bin/dockerd
831720 53124 containerd
522484 17948 /usr/sbin/NetworkManager
493536 12680 /usr/lib/udisks2/udisksd
```

3. CONCLUSIONES Y VALORACIONES PERSONALES

Cómo se ha visto en el punto anterior, hemos sido capaces de programar todos los scripts y funcionan correctamente en todos los posibles casos, considerando también mensajes de error a posibles errores en los argumentos. Por esta parte pues, estamos muy satisfechos con nuestros resultados.

Además, referente al objetivo general de la práctica, mejorar nuestra capacidad de programar en bash scripting y aprender nuevos comandos de manipulación de ficheros y procesos, estamos contentos por haber alcanzado los conocimientos que se esperaban. Aún así, debido a su dificultad al tratar-se de un lenguaje con bastante complejidad, nos ha costado bastante tiempo realizar algunos de los ejercicios, no tanto por no tener la idea de que teníamos que hacer, sino por no saber cómo escribir-la. Tener limitación de comandos ha añadido bastante dificultad.

Finalmente, creemos que hemos trabajado muy bien en equipo, que nos hemos ayudado mutuamente y esforzado para aprender el temario que tocaba. En resumen, y en nuestra opinión, hemos hecho una muy buena práctica.