



Universitat
de les Illes Balears

21719 - Avaluació del Comportament de
Sistemes Informàtics

Fecha de entrega: 16 de marzo

Shell scripting

La entrega de los ejercicios se deberá de hacer en un archivo comprimido en tar.gz y debe de contener la siguiente estructura de ficheros:

```
/<nombre1><APELLIDO1><nombre2><APELLIDO2>/memoria.pdf  
/<nombre1><APELLIDO1><nombre2><APELLIDO2>/scripsts/ej1/ej1.sh  
/<nombre1><APELLIDO1><nombre2><APELLIDO2>/scripsts/ej2/ej2.sh  
/<nombre1><APELLIDO1><nombre2><APELLIDO2>/scripsts/ej3/ej3.sh  
/<nombre1><APELLIDO1><nombre2><APELLIDO2>/scripsts/ej4/ej4.sh  
/<nombre1><APELLIDO1><nombre2><APELLIDO2>/scripsts/ej5/ej5.sh  
/<nombre1><APELLIDO1><nombre2><APELLIDO2>/scripsts/ej6/ej6.sh  
/<nombre1><APELLIDO1><nombre2><APELLIDO2>/scripsts/ej7/ej7.sh  
/<nombre1><APELLIDO1><nombre2><APELLIDO2>/scripsts/ej8/ej8.sh  
/<nombre1><APELLIDO1><nombre2><APELLIDO2>/menu.sh
```

El documento memoria.pdf debe de tener una pequeña documentación con el código de la resolución de cada ejercicio y del script menu.sh. En el directorio scripts se debe de incluir, separado por carpetas, la solución a cada ejercicio, con un archivo .sh con permisos de ejecución. Si para probar cada script se necesita algún archivo auxiliar, éste también se ha de incluir en el directorio correspondiente. Finalmente el archivo menu.sh debe ser un script que funcione como un menú en el que al pulsar los números de los ejercicios (1..8) ejecute cada uno de los scripts.

1. Cree un guión shell, llamado mediana, que para cada fichero pasado como parámetro, calcule la mediana de la serie de números enteros positivos contenidos en cada uno, a razón de uno por cada línea. La sintaxis será la siguiente:

```
>>mediana <fichero1> [<fichero2>....<ficheroN>]
```

Un ejemplo de resultado de ejecución del comando sería (en el caso de que se introduzcan varios ficheros):

Fichero	Mediana
-----	-----
fichero1.txt	5
ficherolargo.txt	6
ficherolargo333.txt	34

La definición de mediana es:

Punto medio de los valores después de ordenarlos de menor a mayor, o de mayor a menor. Se tiene que 50 % de las observaciones se encuentran por arriba de la mediana y 50 % por debajo de ella.

- Por ejemplo, dado un conjunto impar de números: 5, 3, 2, 1, 4:
 - Los números se ordenan: 1, 2, 3, 4, 5.
 - La mediana sería el 3 ya que tiene dos números por debajo de él (1 y 2) y dos números por encima (4 y 5).
- Si el conjunto contuviese un número par de números: 6, 3, 2, 7, 1, 5:
 - Los números se ordenan: 1, 2, 3, 5, 6, 7.
 - En este caso hay dos números en el centro (3 y 5). Entonces la mediana se calcula como la media aritmética de esos dos números. Es decir, la mediana sería 4 $((3+5)/2)$.

2. Implementar un guión de bash (monton) que imprima en su salida estándar (por defecto, la pantalla) un triángulo isósceles formado por asteriscos de texto, simétrico verticalmente, y de altura igual al número que se le pasa como único y obligatorio parámetro. Por ejemplo, si se teclea

```
$ ./monton 5
```

el programa deberá mostrar en su salida lo siguiente:

```
  *
 ***
*****
*****
*****
```

El script deberá comprobar que fue llamado con corrección (es decir, que tiene un parámetro y sólo uno, y que éste es un entero mayor que cero). Obviamente, el programa deberá funcionar para cualquier entero positivo (ya que, aunque la salida no cupiese en pantalla, podría redireccionarse sin problemas a un fichero dado).

3. Escribir un script que reciba como único parámetro un nombre de fichero de texto, y que escriba en la salida estándar, para cada línea de dicho fichero, la lista de palabras de la misma, todas en minúsculas, separadas por espacios, ordenadas alfabéticamente, sin repeticiones, y sin ningún tipo de carácter especial (puntuación, etc.). El guión debe comprobar la corrección de la llamada (esto es, que recibe sólo un parámetro, y que éste es un fichero existente).

Ejemplo:

```
$ cat quijote.txt
```

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lentejas los viernes, algún palomino de anywhere los domingos, consumían las tres partes de su hacienda.

```
$ ./ordpal quijote.txt
```

acordarme cuyo de en ha la lugar mancha mucho no nombre quiero un adarga antigua astillero de en hidalgo lanza los que tiempo un vivía algo carnero corredor de flaco galgo mas olla que rocín una vaca y duelos lentejas las los mas noches quebrantos sabados salpicon y algun anywhere consumían de domingos las los palomino tres viernes de hacienda partes su

4. Cree un guión shell, llamado quiniela, que nos genere de forma aleatoria una quiniela simple de 1 apuesta, es decir un resultado para cada partido. Cuando ejecute el script:

```
>>quiniela
```

El resultado debe ser similar al siguiente:

```
1.- 1
2.- X
3.- X
...
14.- 2
```

Observe la alineación de los números y que todos los resultados del mismo tipo están en la misma columna.

Nota: La variable de entorno \$RANDOM de bash puede utilizarse para generar un número aleatorio entero entre 0 y 32767, distinto cada vez que es invocada.

5. Escriba un script que admita palabras (seguidas de enter) por parte del usuario hasta que este teclee "salir", después debe aparecer en pantalla el número de palabras que se han introducido y una despedida.

6. Modifique el programa anterior de forma que las palabras se vayan guardando en un fichero "palabras.txt"

7. Modifique de nuevo el programa. Consiga que si el usuario introduce un número en lugar de una palabra, se le devuelva la palabra que introdujo en esa posición. Por ejemplo si introduce "5", se le devolverá la palabra número cinco del fichero "palabras.txt"

8. Implementar un guión de bash (totalvsz) que imprima en su salida estándar (por defecto, la pantalla) un listado de todos los usuarios que se encuentran ejecutando algún proceso en el sistema, junto con un número indicando el tamaño total de memoria virtual ocupado por todos los procesos de dicho usuario.

Nota: Para saber como funciona el comando, ves a:

<http://www.linuxito.com/gnu-linux/nivel-alto/38-ps-aux-explicado>

Ejemplo: si la salida de un comando ps aux fuese, por ejemplo:

USER	PID	%CPU	%MEM	VSZ	RSS	TT	STAT	STARTED	TIME	COMMAND
root	18580	21,5	0,0	2461452	912	??	U	10:44AM	0:10.69	/
judit	18465	2,5	0,9	3485716	152552	??	S	10:39AM	0:36.95	/
judit	6131	2,4	1,7	3182536	277872	??	S	dom04PM	13:47.37	/
judit	403	2,4	0,2	1171008	25436	??	S	sáb08AM	26:39.05	/

Entonces una llamada totalvsz a nuestro programa debería mostrar en su salida lo siguiente:

USER	VSZ
root	2461452
judit	7839260