

## Enunciado

En esta práctica deberás realizar una combinación de programas en lenguaje “C” y macros (*shell scripts*) con la *bash* con el objetivo de trabajar con las redirecciones y canalizaciones (*pipes*).

### Primer programa: alturas.sh

Un programa (realizado en *bash*) recibe como parámetros obligatorios por línea de comandos 3 valores enteros, que son respectivamente:

- Altura mínima (ALT\_MIN, expresada en mm).
- Altura máxima (ALT\_MAX, expresada en mm).
- N° de alturas a generar

El programa genera una línea con los siguientes valores obtenidos aleatoriamente (separados por tabulador): altura (n° aleatorio entre ALT\_MIN y ALT\_MAX), categoría (“A” si está por encima del 90% de ALT\_MAX, “B” si está hasta un 25% por encima de ALT\_MIN, y “N” para el resto), y país (aleatoriamente uno de los siguientes: ES, FR, DE, IT, UK, US). Ejemplos:

```
1756 N ES
1875 A IT
```

...

### Segundo programa: filtro.c

Un programa (realizado en “C”) lee líneas por la entrada estándar (STDIN) con la estructura del apartado anterior y analiza los datos que recibe para escribir por la salida estándar (STDOUT) los registros correspondientes a categoría “A” y por la de error (STDERR) a los de categoría “B”. Los de categoría “N” son omitidos.

### Tercer programa: estadisticas.c

Se trata de un programa en “C” denominado “estadistica”, y se encarga de leer todos los datos que recibe por la entrada estándar STDIN. Entiende el mismo formato de los apartados anteriores, y para cada país recibido, calcula los valores mínimo, máximo y medio (redondeado sin decimales), imprimiendo esos valores al finalizar. Ejemplo:

```
ES 1521 1987 1745
UK 1498 1916 1722
```

...

## Realización

1. Crea un directorio para esta práctica, denominado “*practica\_1*”.
2. Crea “*alturas.sh*”, y verifica su funcionamiento, pasándole diferentes parámetros. Prueba también a no pasarle nada. Dale permisos de ejecución (utiliza el octal 755 y justifica el uso de este valor).
3. Crea “*filtro.c*”, compila y depura el programa. Pruébale primero generando un fichero que contenga la salida de la macro y poniéndola como entrada de este. Después, utiliza la canalización de datos (pipe) para encadenar la salida de la macro anterior con la entrada de este programa.
4. Separa la salida de este programa en dos ficheros, uno correspondiente a la salida estándar (*A\_datos.stdout*) y otro a la salida de error (*B\_datos.stderr*). Utiliza para ello las redirecciones que has visto en el laboratorio (> y 2>) Verifica que los ficheros de datos son correctos.
5. A partir de los ficheros generados en el apartado anterior, pásalos por la entrada estándar al programa “*estadisticas*” y obtén las estadísticas respectivas en dos ficheros (*A.stats* y *B.stats*).
6. Mediante instrucciones de la *bash*, obtén los siguientes valores:
  - a. Número de líneas de cada fichero (*A.stats*, *B.stats*) y totales.
  - b. Ordena el fichero (*A.stats*) por altura media y filtra para quedarte con el más alto.
  - c. Ordena el fichero (*B.stats*) por altura media y filtra para quedarte con el más bajo.

*./alturas.sh 60 100 20 | ./filtro 1>A\_datos.stdout 2>B\_datos.stderr*

7. Usa el comando `tee` entre la ejecución de la macro `alturas.sh` y el programa `filtro` para ver la información intermedia y guárdalo en el fichero "`alturas.debug`".
8. Generar un fichero `make` para compilar de forma modular este proyecto, de tal forma que pueda compilar ambos programas.
9. Cuando lo tengas debes entregar un único fichero de formato `tar` con lo que has realizado (explicación en comentarios, fuentes, datos, constantes, `make`, entrada, macro, ...) a través de Moodle. Se te valorará la calidad del código, que todo funcione y la macro y `make` realizados, así como la documentación / comentarios.

## Evaluación

Esta práctica será una súper-versión del examen posterior y te servirá tanto de referencia como de evaluación (30%). En las siguientes sesiones se te pedirá que muestres su correcto funcionamiento y cualquier aspecto que el profesor crea oportuno.

## Ayudas

1. El canal 0 es `stdin`, el canal 1 es `stdout`, el canal de errores 2 es `stderr`. Los primeros se toman por defecto. Si queremos escribir por el de errores deberemos usar la instrucción:

```
fprintf(stderr, mismo formato que printf que sería stdout);
```

2. Para leer un *string* se utiliza el `fgets` y para trocear una línea se utiliza `strtok`. Tienes un ejemplo de uso en el `man`.
3. La generación de números aleatorios en bash se puede realizar se puede hacer con ayuda de la función `$RANDOM` (no es una variable), que devuelve un número entre 0 y 32.767 cada vez que se utiliza. Por ejemplo, para obtener un valor aleatorio entre 0 y 10 basta con hacer:

```
ALEATORIO=$(( $RANDOM % 11 ))
```

4. Ten en cuenta que en una macro se pueden definir array, puede servirte este ejemplo:

```
# definicion de un array de 6 elementos
mi_array=(uno dos tres cuatro cinco seis)
# imprimir el primer elemento
echo ${mi_array[0]}
```

5. Para ver cómo funciona el comando `tar` usa el manual. Ejemplos de uso:

- Crea un tar con cuatro ficheros

```
tar -cf fichero.tar uno dos tres cuatro
```
- Lista el contenido de un tar de forma detallada (verbose)

```
tar -tvf fichero.tar
```
- Extrae en el directorio los ficheros que contiene el tar

```
tar -xf archive.tar
```