Control Laboratorio 2 2012QT

0) **(0,5 puntos)** Control de errores y utilización de la función Usage en todos los códigos.

1) **(0.5 puntos)** Crea un ***Makefile*** que permita generar ***todos*** los programas del enunciado a la vez y cada uno de ellos por separado. Añade una regla (***clean***) para borrar todos los binarios y/o ficheros objeto y dejar solo los ficheros fuentes. Los programas deben generarse si, y solo si, ha habido cambios en los ficheros fuentes.

2) **(0.5 puntos)** Implementa un programa que llamaremos **my_wcl** cuyo funcionamiento será similar a ejecutar el comando wc con la opción -l. El programa tendrá el siguiente formato:

```
$my_wcl
```

El programa lee de la entrada std, cuenta los saltos de línea existentes (carácter \n) y escribe el resultado por la salida std. La lectura debe realizarse byte a byte. El mensaje tendrá el formato "El número de líneas es XXXX"

Por ejemplo:

```
$my_wcl
linea 1
linea 2
linea 3 (ctrl D)
El numero de lineas es 3
```

3) **(0.75 puntos)** Copia el programa my_wcl en **my_wcl_v2**. Modifica este último para que las lecturas se realicen utilizando un buffer de 32bytes.

4) **(1.5 puntos)** Implementa una tercera versión **my_wcl_v3** en la que el programa recibe dos nombres de fichero como parámetro que utilizará como entrada y salida de datos.

El formato es el siguiente:

```
$ my_wcl_v3 ficheroentrada ficherosalida
```

- El parámetro ficheroentrada se utilizará como fichero de entrada. Se considera un error que el fichero no exista previamente.
- El parámetro ficherosalida, se utilizará como fichero de salida. Si el fichero no existe se creará. Si existe, se debe truncar su contenido.

5) **(0.25 puntos)** Anota en el fichero **respuestas.txt** la línea de comandos que utilices para contestar a esta pregunta y los resultados obtenidos:

¿Que comando utilizarías para ver la cantidad de llamadas a sistema "read" que realizas con la versión my_wcl y my_wcl_v2?

Calcúlalo utilizando como fichero de entrada el fichero my_wcl_v3.c en los dos casos.

6) **(3 puntos)** Queremos hacer un programa que implemente el equivalente a lo que haría la shell al realizar la siguiente combinación de comandos:

```
$ grep palabra fichero | wc -l
```

El programa se llamará **cuantaslineas**, y tendrá el siguiente formato:

```
$cuantaslineas palabra fichero
```

7) **(2 puntos)** Queremos hacer un programa (**1deN**) que lea de la entrada std números enteros (en representación interna de la máquina) y muestre 1 de cada N (donde N será un parámetro del programa). Después de leer cada número, lo mostraremos en formato ascii por la salida std.

   El formato es el siguiente:

   ```
   $ 1deN N
   ```

   Utilizad para probar el fichero NumerosConsecutivos que contiene números consecutivos en representación interna de la maquina, por ejemplo:

   ```
   $1deN 2 < NumerosConsecutivos
   0 2 4 6 .... 1000
   ```

8) **(1 punto)** El programa sumavector (similar al de la S10), calcula la suma de los elementos de un vector. En este caso, hemos transformado el código para que sea muy sencillo pasar de una ejecución secuencial (como os damos) a una concurrente utilizando pthreads. Añade las llamadas a la librerías de pthreads para que la función SumaParcial se ejecute de forma concurrente para los N threads (NTHREADS) y el resultado sea correcto. En este caso no hay problemas de concurrencia, solo hay que añadir la gestión de los threads. Llama al programa **sumavector_pth.**

Qué se tiene que hacer

- El makefile
- Los códigos de los programas en C
- La función Usage() para cada programa
- El fichero respuestas.txt con las respuestas a las preguntas

Qué se valora

- Que sigas las especificaciones del enunciado
- Que el uso de las llamadas a sistema sea el correcto
- Que se comprueben los errores de **todas** las llamadas al sistema
- Código claro y correctamente indentado
- Que el makefile tenga bien definidas las dependencias y los objetivos
- La función Usage() que muestre en pantalla cómo debe invocarse correctamente al programa en caso que los argumentos recibidos no sean adecuados.
- El fichero respuestas.txt

Qué hay que entregar

- Un único fichero tar con el código de *.c, respuestas.txt y el Makefile
  - tar zcvf control-L2.tar.gz *.c respuestas.txt Makefile