

Makefile (0,5)

Crea un archivo *Makefile* para compilar todos los programas del examen, usando reglas independientes para cada uno. Incluye una regla llamada *clean* para eliminar todos los ejecutables, archivos objeto y/o ficheros temporales. Los programas deben compilarse si y sólo si se han actualizado los archivos de código fuente que los componen.

Control de errores y función Usage (0,5)

Todos los programas deben incluir un control adecuado de los argumentos usando la función `usage` y deben controlar los errores en TODAS las llamadas al sistema (a excepción del write por pantalla).

1. Cuenta (2,75 puntos)

Te proporcionamos un código fuente llamado *cuenta_basic.c* que incluye una serie de funciones necesarias para hacer los siguientes ejercicios. Haz una copia de este fichero y llámalo **cuenta.c** para implementar un programa que cuente cuántas veces aparece un carácter en un fichero utilizando N procesos, de forma que cada uno revise una parte del fichero.

El programa recibirá tres parámetros: número de procesos a crear (`num_procs`), nombre del fichero (`fileName`), carácter a buscar (`c`)

\$./cuenta num_procs fileName char_to_find

Se pide que el programa ofrezca la siguiente funcionalidad:

- Comprobar el número de argumentos utilizando una función `Usage`.
- Comprobar que el tamaño del fichero es divisible por el número de procesos, en caso contrario escribe un mensaje explicativo y terminar el programa. Para esto se os ofrece la función `FileSize`.
- Crear `num_proc` procesos concurrentes. Cada uno de ellos llamará a la función `CountCharacter`, con los parámetros específicos del proceso, y terminará su ejecución.
- El proceso padre esperará la finalización de sus hijos, esperándolos de forma pasiva. Para cada uno de ellos, al detectar que ha terminado, ejecutará la función `ReadPartialCount`. El valor que devuelve esta función se debe ir acumulando y cuando terminen todos los procesos se escribirá el resultado por la pantalla.

2. Cuenta_exec (1,5 puntos)

Copia el programa `cuenta.c` en **cuenta_exec.c**. Modifica el programa `cuenta_exec.c` para que, en lugar de utilizar la función `CountCharacter`, cada proceso hijo mute (reemplace la imagen del proceso actual) al ejecutable `cuenta_trozo` (del cual también te proporcionamos el código fuente) y que ofrezca la misma funcionalidad, pero en un programa aparte.

3. Cuenta_sync (3,5 puntos)

Copia el programa cuenta.c en **cuenta_sync.c**. Modifica cuenta_sync.c para que el proceso padre, después de crear a los procesos hijos, espere, de forma pasiva, a la recepción de un signal SIGALRM debido a una alarma de 5 segundos. Al recibirlo, enviará un signal SIGUSR1 a sus hijos. Los hijos, antes de llamar a la función CountCharacter, realizarán una espera activa hasta recibir el SIGUSR1. Se valorará que cada proceso tenga capturado únicamente los signals que va a tratar. Puedes hacer este ejercicio asumiendo que habrá un máximo de 10 procesos hijos. Además, también se valorará que se introduzcan los mecanismos necesarios para asegurar que los signals se bloqueen en aquellos periodos de tiempo iniciales donde todavía el proceso no está preparado para recibirlo.

4. Cuenta_dyn (1,25 puntos)

Copia cuenta.c en **cuenta_dyn.c**. Modifica cuenta_dyn.c para que el proceso padre espere, de forma activa, la finalización de sus procesos hijo y libere el PCB de los procesos hijo que finalicen mediante la reprogramación del signal SIGCHLD. De tal forma que, dentro de la función de reprogramación de este signal, se encuentre la llamada al sistema que libera el PCB. Además, la función mostrará un mensaje por pantalla para indicar si la causa de finalización del proceso hijo ha sido voluntaria o bien debido a un signal.

Se valorará en los programas

- Seguir las especificaciones de los ejercicios.
- El uso correcto de las llamadas al sistema.
- Control de errores en las llamadas al sistema.
- La claridad del código y una indentación adecuada.
- Que las reglas del Makefile tengan las dependencias y los objetivos requeridos correctos.
- La función Usage() muestra por pantalla, en caso de invocación incorrecta, la línea de comandos correcta para ejecutar el programa.

Qué se debe entregar

Un único fichero *tar.gz* con el código fuente y el Makefile. Para generarlo podéis usar el comando siguiente:

```
tar zcvf clab1.tar.gz *.c Makefile
```