SIMLAB1 (adapt. CLAB 2122Q1) SO2324q1

Makefile

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 que tú generes, y dejar sólo los ficheros fuente. Los programas deben generarse si, y solo si, ha habido cambios en los ficheros fuente.

Control de errores

Para todos los programas que se piden a continuación deben comprobarse los errores de TODAS las llamadas a sistema (excepto el write por pantalla). Para mostrar los errores y acabar la finalización por error de llamada a sistema, en el paquete junto al enunciado tenéis el fichero **error_exit.h** que contiene la especificación de la función que realiza esta tarea. Debéis invocarla en caso de que vuestro programa detecte errores en las llamadas a sistema. El código que implementa la función de *error_exit.h* lo tenéis ya compilado en el fichero **error_exit.o**

Todos los programas también deben controlar los argumentos de entrada y definir la función Usage().

Ejercicio 1

Escribe un programa (jerarquia1.c) que reciba una secuencia de entre 1 y 10 nombres de fichero como argumento. El programa debe crear de forma **concurrente** tantos procesos hijo como nombres de fichero haya recibido. Cada proceso hijo se encarga de tratar uno de los ficheros. En este primer apartado el tratamiento consiste simplemente en mostrar por pantalla un mensaje con el nombre del fichero que le ha tocado y acabar la ejecución pasando como parámetro de finalización el número de orden de su creación.

Por su parte, el proceso padre sólo tiene que esperar a que acaben todos los procesos hijos. La espera la tiene que hacer **en orden de creación** (hasta que no libere el PCB del primer proceso creado no puede pasar a liberar el PCB del segundo). Para cada hijo muerto, deberá mostrar en pantalla un mensaje con el pid del proceso hijo y con el parámetro que el hijo ha pasado al exit.

Ejercicio 2

Haz una copia de jerarquia1.c y llámala jerarquia2.c. Modifica el código de jerarquia2.c para que cada hijo haga ahora un tratamiento diferente. **Cada hijo**, después de mostrar el mensaje por pantalla con fichero que tiene que tratar, creará un grupo de tres procesos que se ejecutarán de manera **secuencial. Cada grupo de 3 "hermanos"** mutará de manera que ejecutarán la siguiente secuencia de comandos (siendo nombre_fichero el nombre del fichero que le toca tratar y nombre_nuevo se crea concatenando nombre_fichero y el sufijo "sin_repes")**:**

1) wc -l nombre_fichero

   Muestra por pantalla el número de líneas de "nombre_fichero"

2) sort -u nombre_fichero -o nombre_nuevo

   Guarda en el fichero "nombre_nuevo" el contenido ordenado de "nombre_fichero" y sin líneas repetidas

3) wc -l nombre_nuevo

   Muestra por pantalla el número de líneas de “nombre_nuevo”

Qué hay que hacer

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

Qué se valora

- Que sigas las especificaciones del enunciado
- Que el uso de las llamadas al sistema sea el correcto
- Que se comprueben los errores de **todas** las llamadas al sistema
- Que el código sea claro y correctamente indentado
- Que el Makefile tenga bien definidas las dependencias y objetivos
- Que la función Usage() muestre por pantalla como debe invocarse correctamente el programa en el caso que los argumentos recibidos no sean los adecuados

Qué hay que entregar

Un único fichero tar.gz con el código de todos los programas, el Makefile, y las respuestas en respuestas.txt:

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