Control Laboratorio 1 SO1819q2

Makefile (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 sólo los ficheros fuente. Los programas deben generarse si, y solo si, ha habido cambios en los ficheros fuente.

Control de errores (0.5 puntos)

Para todos los programas que se piden a continuación deben comprobarse los errores de TODAS las llamadas a sistema (excepto el write por pantalla), controlar los argumentos de entrada y definir la función Usage().

Ejercicios (9 puntos)

A) **[2.5 puntos]** Escribe un programa (hermano.c) que cree *n* procesos de forma concurrente, donde *n* es el primer y único argumento del programa. Cada hijo debe escribir por la salida estándar un mensaje con el PID del anterior proceso creado por el padre (es decir, su hermano mayor) y acabar. El primer hijo debe escribir el PID del padre, al no tener hermano mayor. Antes de finalizar, el padre debe bloquear todos los signals y esperar a que acaben todos sus hijos para liberarlos.

B) **[2.5 puntos]** Escribe un programa (signal.c) que reciba como primer y único argumento el PID de un proceso. El programa debe esperar 1 segundo utilizando una espera bloqueante y a continuación enviar un SIGUSR1 al proceso con el PID pasado como argumento. Si no se puede mandar el signal porque el proceso con PID no existe, el programa debe mostrar un mensaje de error y acabar.

C) **[2 puntos]** Haz una copia del programa hermano.c y renómbralo a hermano2.c para que ahora cada hijo después de mostrar el mensaje, mute al programa **signal** pasándole como argumento el PID de su hermano mayor (o el del padre en el caso del primer hijo). Modifica también el padre para que después de esperar a cada uno de sus hijos muestre por la salida estándar un mensaje con el PID del hijo que ha muerto y la causa de su muerte (exit o signal).

D) **[1 punto]** Haz una copia del programa hermano2.c y renómbralo a hermano3.c para que ahora almacene la(s) variable(s) que necesite para guardar el PID del hermano mayor (o el del padre en el caso del primer hijo) en memoria dinámica utilizando las funciones de librería de C. Toda la memoria reservada debe liberarse antes de finalizar.

E) **[1 punto]** Apunta en el fichero "respuestas.txt" el tamaño del heap del proceso padre después de reservar memoria y antes de liberarla. Explica que ficheros o código has utilizado para calcular el tamaño del heap. ¿El tamaño del heap es igual o diferente al tamaño del bloque de memoria reservado? Argumenta tu respuesta en el fichero "respuestas.txt".

Control Laboratorio 1 SO1819q2

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
- El fichero respuestas.txt

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 clab1.tar.gz Makefile respuestas.txt *.c
```