SISTEMAS OPERATIVOS I 2º DE G.I.I.

Segunda práctica

Infinito.c

Primer miembro: JORGE CUADRADO CRIADO (i0924801)

Segundo miembro: DAVID LAVADO GONZALEZ (i0910797A)

28 de noviembre de 2023

UNIVERSIDAD DE SALAMANCA

Lo primero a destacar de nuestro código sería la creación de los manejadores de las señales, hemos recurrido a los “structs”.

Texto

Descripción generada automáticamente

Como se puede observar asignado a cada “handler” la dirección de memoria de cada señal. También con fines de evitar posibles errores y bloqueos hemos creado un “flag”.

Continuamos por las máscaras de señales, hemos creado un conjunto llamado “mask” con fin de almacenar todas las señales. El conjunto será llenado posteriormente con todas las señales posibles y así poder usar máscaras para bloquear y desbloquear señales en el proceso actual.

Texto

Descripción generada automáticamente

Lo siguiente será configurar el manejo de las señales y establecer el comportamiento de cada señal. Para dar robustez al programa hemos comprobado cualquier posible error.

Texto

Descripción generada automáticamente

La creación de procesos es la parte más compleja y longeva del programa por lo que nos limitaremos a explicar nuestra solución resumidamente. Hemos creado una estructura jerárquica de procesos. Lo primero es obtener el ID del proceso principal “pidp” usando “getpid()” y mostrarlo por pantalla. Se realiza una serie de bifurcaciones “fork()” para crear varios procesos hijos (h1, h2, h3 y h4) dentro del árbol de procesos. Cada proceso hijo establece un comportamiento en un bucle infinito utilizando “sigsuspend(&mask)” para esperar una señal específica (SIGUSR1 o SIGUSR2). Después de recibir la señal, envía otra señal (SIGUSR1) a otro proceso hijo usando “kill()”. El proceso padre maneja una señal de alarma “SIGALRM” con la función alarma(). La función alarma() en el proceso padre realizará varias acciones:

Espera a que dos procesos hijos específicos (h2 y h3) mueran usando señales “SIGUSR2” y luego mata a todos los hijos con “SIGKILL”.

Espera a que todos los hijos terminen “wait(NULL)” y muestra el número de veces que ha dado la señal, que se utiliza para controlar el bucle principal.

Texto

Descripción generada automáticamenteCaptura de pantalla de un celular

Descripción generada automáticamente

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Por último, hemos creado la función alarma que será la encargada de controlar y finalizar a los hijos a parte de realizar acciones tras recibir señales concretas. Lo primero que hará será verificar que el proceso actual es el padre. Creará y eliminará señales y después se encargará de matar a los hijos y comprobar que reciben la información. Está función calculará y mostrará también el número de vueltas dadas.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente