Tarea 1

Sistemas Operativos

Integrantes:
Rodrigo Álvarez - Valeria Fuentes

Profesor:
Martín Gutierrez
Ayudante:
Cristian Villavicencio

11 de septiembre de 2021

Índice general

CAPÍTULO 1

Ejercicio 1

En este ejercicio se solicita crear K procesos los cuales son transformados aleatoriamente en procesos zombies, huérfanos o simplemente terminan.

En nuestra solución hemos generado un bucle `for` que recorre desde el **0** hasta el **K-1** generando procesos mediante el syscall `fork`. Para cada proceso hijo creado se ha generado un número aleatorio entre el 0 y el 2, que representa la transformación que tomará el proceso.

Cabe destacar que para evitar problemas, se ha decidido mantener al proceso original inalterado.

```
if (r == 0) // OPCION 0 -> HUERFANO
{
    if(parent_pid>=padre_pid){
        kill(parent_pid, 9);
        sleep(5);
    }
    else{
        printf("OYE %d ya eres HUERFANO, no mates a tu padre (que es importante para el computador)\n", getpid());
    }
}
```

Para el caso de que un proceso deba transformarse en huérfano usaremos el comando kill para matar al padre. Luego de esto se usará el comando sleep para mantener el proceso vivo y checkear que efectivamente el proceso se convierte en huérfano desde la consola. Lo anterior mencionado se ejecutara únicamente si el pid del padre corresponde a uno de los procesos creados en el código, en caso contrario se mostrara un mensaje indicando que el proceso ya se encuentra huérfano.

```
else if (r == 1) // OPCION 1 -> ZOMBIE
{
    kill(parent_pid, SIGSTOP);
    exit(0);
}
```

Para el caso de un proceso que se deba transformar en zombie, mandaremos una señal de STOP al padre para evitar que este proceso muera, y usaremos el comando `exit(0)` para que finalice el proceso hijo, transformándolo en zombie.

```
else // OPCION 2 -> MUERE/ TERMINA
{
    sleep(3);
    exit(0);
}
```

Para el ultimo caso el proceso esperará 3 segundos antes de terminar. Esto permitirá que no exista problema para aquellos procesos que se transformen en zombie, ya que para estos dependen directamente de que el padre continué ejecutándose.

1.1 | Observaciones

Existen pequeños errores que pueden ocurrir en la ejecución de este programa, los cuales son:

1 - El proceso principal nunca se transforma directamente según las opciones indicadas.

2 - Se crean K+1 procesos (el proceso principal más los creados mediante forks).

3 - Para el caso en que un proceso padre decide ser zombie y su hijo decide ser huérfano, en ciertas ocasiones sucederá que el padre se "zombieficará" antes que el hijo lo mate y por tanto el hijo será adoptado por otro proceso. Cuando esto ocurra, el hijo no tendrá porqué matar a su padre debido a que ya es un proceso huérfano.

4 - Para el caso que un proceso hijo quiera convertirse en zombie y el proceso padre previamente haya elegido el camino de la muerte se eligió dar prioridad al hijo zombie mediante la implementación de una espera de 3 segundos para el padre (y en general todo proceso) que elija la opción de simplemente terminar, con esto se evita que el proceso padre muera y permite que el proceso hijo se convierta efectivamente en zombie.