Licenciatura en Computación
PROGRAMACIÓN CONCURRENTE
Práctica 1

Trimestre 23I

1. Objetivo

Conocer y utilizar los llamados al sistema fork(), getpid(), getppid() y wait() para construir programas concurrentes.

2. Introducción

Revisar el estándar POSIX (http://pubs.opengroup.org/onlinepubs/9699919799/nframe.html) referente a los llamados al sistema mencionados en el objetivo.

3. Actividades

A. Compilar, ejecutar y comprender el programa EjemploForkWait.c disponible en el aula virtual. Observe el no determinismo ejecutando varias veces el programa.
B. Descomentar la instrucción wait, compile, ejecute y comprenda qué sucede.
C. Determine cuántas veces debería escribirse el mensaje “Hola mundo” en los siguientes fragmentos de código. Suponga que n = 3 para los programas que utilizan esta variable.

`if ((cpid = fork())== 0)`
`  printf(“Hola mundo”);`
	`for( i = 0; i < n; i++) {`
	`  cpid = fork();`
	`  if (cpid == 0) {`
	`  printf(“Hola mundo”);`
	`  exit();`
	`  }`
	`}`

`if ((cpid = fork()) != 0)`
`  printf(“Hola mundo”);`
	`for( i = 0; i < n; i++) {`
	`  cpid = fork();`
	`  if (cpid != 0) {`
	`  printf(“Hola mundo”);`
	`  wait(----);`
	`  exit(----);`
	`  }`
	`}`

D. Descargue los programas EjemploA.c y EjemploB.c compile, ejecute y, a partir de las salidas, dibuje la estructura de procesos que resulta de cada uno de ellos. Para esto, haga una gráfica poniendo un vértice por cada proceso y entre cada par de procesos padre-hijo dibuje un arco dirigido que salga del padre y llegue al hijo.

E. Elabore los siguientes programa concurrentes usando:
   a. Una lista de procesos que dados tres enteros positivos c, x *y* n, calcule cx^n distribuyendo el trabajo de la siguiente manera: el proceso original lee los parámetros, su descendiente calcula x^2, el descendiente de este último calcula x^3 y así sucesivamente hasta que se calcula x^n para que finalmente, un último proceso final calcule cx^n y lo imprima. Aproveche el copiado de memoria de padre a hijo para hacerle llegar a cada proceso hijo el cálculo que ya hizo su padre.
   b. Un abanico de procesos que dado un número entero positivo M determine cuántos números primos hay en el intervalo [1,M]. El trabajo se divide por igual entre los n procesos del abanico. Cada proceso hijo recibe un subintervalo, cuenta cuántos primos hay en su subintervalo y le regresa ese dato a su proceso padre vía el llamado exit. El proceso padre, después de haber recibido los resultados de todos los procesos hijos, los suma e imprime el resultado.
   c. ** Un árbol binario de procesos que definido un arreglo de tamaño 2^k, k>1 de números enteros desordenados, determine cuál es el elemento mayor. El algoritmo procede con cada proceso interno del árbol dividiendo el arreglo en dos, creando dos procesos hijos y asignándole a cada uno la tarea de encontrar el mayor en una de las mitades; una vez que los procesos hijos terminan recibe los resultados, los compara y envía el mayor a su proceso padre o a pantalla en el caso del proceso raíz. Los procesos hoja son aquellos que reciben un arreglo de tamaño 1 y proceden a regresar el número contenido en el subarreglo que les corresponde.

F. Entregable:

Elaborar un reporte en formato pdf de la práctica que contenga: sus respuestas para el ejercicio C. Los grafos del inciso D y los programas fuente de los ejercicios E.a y E.b (opcionalmente E.c) así como el enlace a gdb online en donde se encuentren esos programas. Considere que, al ejecutarse, en la salida de cada programa, se deben imprimir los datos con los que se hará el cálculo y cada proceso debe imprimir su resultado precedido de su identificador y el de su proceso padre. Un salto de línea debe seguir a toda línea impresa por un proceso.

G. Entrega: subir los entregables en el enlace previsto para tal efecto en el aula virtual.

Elaboró: Elizabeth Pérez Cortés