

TAREA 3

PROGRAMACIÓN AVANZADA

9 de marzo de 2015

La tarea debe ser entregada antes de las 11:59pm del 16 de marzo de 2015. Sólo deben entregar los programas fuente. Los programas que entreguen deben estar documentados y deben explicar claramente cómo funcionan. La documentación representará un 20% de la calificación de la tarea.

Problema 1. El problema consiste en escribir un programa que reciba de la línea de comandos el número de procesos hijo que debe crear usando `fork()`. Cada proceso hijo deberá generar un número aleatorio entre 1 y 15, que será el número de segundos que el proceso deberá ponerse a «dormir». Mientras tanto, el proceso padre deberá esperar a los hijos y cada vez que un hijo termine, deberá imprimir en la pantalla el identificador del proceso que terminó y el número de segundos que estuvo dormido. La comunicación entre padre e hijos deberá hacerse usando tuberías (pipes).

Problema 2. El objetivo es implementar una solución al problema de productor - consumidor que se vio en clase. El proceso productor debe producir un cierto número de valores (10 por ejemplo) y debe escribirlos, uno a uno, en una variable compartida. El consumidor debe irlos leyendo y debe imprimir en la pantalla los valores leídos. Los procesos deben sincronizarse para que el consumidor no pierda un sólo valor, esto es, que el productor no sobre escriba ningún valor antes de que el consumidor lo haya leído y para que el consumidor no se salte ningún valor. Para esto deben usar semáforos y memoria compartida en el estándar de System V.

Problema 3. Escriba un programa que cree un subprocesso (proceso hijo) con el cual se comunicará a través de una tubería. El proceso padre deberá entonces leer de la pantalla mensajes de texto que le enviará al proceso hijo. El proceso hijo deberá entonces imprimir el mensaje en la pantalla. Cuando el proceso padre reciba el mensaje «termina», deberá enviarle un mensaje al proceso hijo para que termine. El hijo deberá imprimir el mensaje `proceso id terminando`, donde `id` debe ser el identificador del proceso, y el proceso padre deberá imprimir al final lo mismo que el proceso hijo, usando su propio identificador.