

Sistemas Operativos: Proyecto

Curso 2020/2021

Grupo 2292_02
Javier López de la Nieta
Daniel Varela Sánchez
10/05/2021

Análisis del problema

En este proyecto, el objetivo es crear una red de mineros que busquen de forma paralela una solución a un problema matemático dado. Para ello, se nos fue dado el archivo `miner.c` en el cual un único minero buscaba una solución al problema. Para mejorar la eficiencia de esos mineros, cada minero deberá tener un número de hilos o “trabajadores” los cuales busquen la misma solución y una vez encontrada, esta será guardada en un bloque. Para implementar la red de mineros que puedan buscar la misma solución, es necesario unir todos los mineros creados con una lista guardada en un segmento de memoria compartida, segmento el cual a su vez aprovecharemos para implementar el sistema de votación.

Algoritmos utilizados

Para este proyecto ha sido necesario poner en práctica todo lo usado en las anteriores entregas, desde el uso de semáforos, memoria compartida e hilos, a comunicación entre procesos mediante tuberías. Hemos necesitado un bucle que crease un número arbitrario de hilos y luego otro bucle que los esperase, creación de semáforos mutex para controlar el acceso a la memoria compartida, captura de señales para actuar en base a ellas y comunicación entre procesos mediante tuberías para printear toda la información recogida en un archivo `txt`.

Módulos

En esta práctica era inevitable la modularización, ya que en muchos casos lanzaremos dos procesos los cuales contienen un `main`. Por ello, hemos creado un archivo llamado `common`, en el cual guardaremos todas las funciones utilizadas tanto en `miner.c` como en `monitor.c`. Este último, era necesario crearlo, ya que necesitaba ser un proceso propio que se ejecutase a placer del usuario, y este se encarga de implementar la funcionalidad de un monitor. En `monitor.c` hemos decidido que para evitar un `main` excesivamente grande era necesario implementar dos funciones las cuales implementasen el funcionamiento de los procesos padre e hijo del monitor, haciendo este archivo mucho más limpio y legible.

Estructuras de datos

Para este proyecto, se nos fueron dadas varias estructuras de datos, la estructura `Block`, y la estructura `NetData`. La primera se utilizara para guardar las monedas obtenidas por cada minero, el número objetivo, la solución, y además de eso también servirá para crear una lista doblemente enlazada de bloques. Por otro lado, `NetData`, se utilizará para manejar toda la red de mineros implementada. En ella guardaremos las ids de los mineros, las votaciones de los mismos, el número total de mineros, el último minero que ha entrado, el último ganador, y además de eso la id del monitor que se usará para printear toda la información en pantalla.

Pruebas realizadas

En un primer momento, fue necesario aprender el funcionamiento de los mineros para poder trabajar con ellos, por lo que probamos cambiando los valores del problema matemático para comprobar cómo estos afectaban al tiempo de resolución.

Debido a varios errores con la implementación del monitor, comprobamos paso a paso la comunicación entre el proceso miner y el proceso monitor para averiguar por qué había fallos pasando esta información.

También, debido a varios problemas con la memoria compartida, tuvimos que realizar pruebas con cómo se guardaba la información, ya que esta no era correcta. Para ello, mediante prints y debuggeando conseguimos arreglar algunos de estos problemas.

Autoevaluación

Este proyecto ha sido muy complicado para nosotros, no por la dificultad del mismo, la cual también era mucha, sino por el poco tiempo disponible, coincidencias con otros exámenes y trabajos, etc. Solo hemos conseguido implementar de manera 100% funcional los dos primeros niveles del trabajo, los cuales funcionan perfectamente siempre y sin dar ningún error. Y pese a que hemos intentado avanzar en lo que restaba de práctica, no paraban de surgir errores en niveles por todo el código e incluso en niveles anteriores. Por ello solo entregamos los dos primeros niveles del proyecto, lo que corresponde con un 3, que es la nota que creo que nos merecemos, que aunque nos gustaría haber sido capaces de aspirar a más nota, sé que no hemos trabajado lo suficiente, o quizás no de la forma más eficaz para ello. Pese a eso, hemos comentado todas y cada una de las funciones, además de comentarios sobre el funcionamiento del código. Al fin y al cabo es una asignatura que nos gusta y cuyas prácticas son entretenidas, y creo que eso se puede ver en las notas de las anteriores entregas, pero la rabia por no haber sido capaces de avanzar más en esta última es innegable.

También me siento en parte culpable (Javier Lopez), ya que como te comenté por correo electrónico, tuve ciertos problemas familiares y no pudimos avanzar en la práctica durante una semana. Este tipo de prácticas son en equipo, y si uno de los miembros no trabaja por el motivo que sea, esto tiene muchísima repercusión en el resultado final. Sé que no es una excusa, y no pretendo que lo sea, pero creo que es una práctica que me habría gustado muchísimo hacerla con tiempo y poder disfrutar de ello como se merece. Suena un tanto ridículo, pero aunque a veces sea exasperante estar horas programando, cuando consigues tu objetivo la satisfacción es indescriptible.