

MEMORIA PROYECTO DE SOPER

Claudia González Arteaga y Laura Iniesta Rodríguez
Grupo 2261

El sistema que hemos implementado consta de un proceso principal, este proceso se va a encargar de crear otros procesos, en concreto 512 trabajadores y un ilustrador, todos ellos comparten memoria, el proceso principal tiene una estructura con datos que inicializa leyendo del fichero que se indique al ejecutar el programa, estos datos se van a tener que ordenar y para ello se van a dividir los trabajos de ordenación en niveles y partes, cada nivel tiene varias partes, cada una se va a mandar a los procesos para que las hagan.

El trabajador recibe esa tarea mediante una cola y se encarga de resolver la ordenación, cuando recibe una alarma (que se envía cada segundo) informará mediante tuberías(trab_ilus) al ilustrador de su estado, indicando el nivel en el que está trabajando, la parte y su id, el ilustrador recibe esta información de todos los trabajadores y la muestra, tras ello avisa por otra tubería (ilus_trab) de que ya ha recibido la información y los trabajadores continúan con la tarea que estaban realizando.

Hemos implementado memoria compartida con shm_open y mapeando con mmap, hemos comprobado que funciona correctamente ya que antes de implementar otra cosa probamos a ejecutarlo y funcionaba con un proceso, hemos inicializado una cola de mensajes por la que se pasa una estructura llamada estado que indica la tarea que tiene que realizar un trabajador.

Hemos creado las tuberías para que se comuniquen los trabajadores con el ilustrador trab_ilus y el ilustrador con los trabajadores ilus_trab.

Los trabajadores mandan su estado por trab_ilus y se quedan a la espera de la respuesta del ilustrador por ilus_trab (un int con valor 1).

Para que no haya mezcla de tuberías, cuando llamamos al trabajador le enviamos sus dos tuberías.

Hemos hecho un fork para crear al proceso ilustrador y si es el ilustrador llamamos a su función que mientras no acabe el proceso principal y le envíe sigterm imprime la información que recibe por tuberías.

Hemos hecho un bucle en el que se genera un trabajador por max_parts que es el máximo de trabajadores que van a poder actuar, si es el trabajador se llama a su función en la que va a leer de la cola las tareas que tenga que realizar hasta que se active la alarma y tenga que enviar su información al ilustrador, una vez enviada se queda a la espera en un while de leer un 1 en la tubería ya que eso significa que el ilustrador habría acabado y puede continuar con la ejecución, para implementar esto hemos hecho dos while uno que espera a recibir la señal sigterm del proceso principal y dentro otro que espera a recibir sig_alarm, en este último se va a leer de cola hasta recibir la señal en cuyo caso la variable global se pone a 0, se guarda la información que se va a enviar por tubería y se hace un break de forma que

dejamos de leer pero solo hasta que recibamos un 1 por tubería que volveríamos a entrar en el while sig alarm igual a 0.

Todos los datos de números a ordenar se encuentran en la memoria compartida, estructura sort, que también contiene un semáforo con el que controlamos el acceso a la memoria haciendo un wait antes de usar sort y un post justo después para evitar superponer información en nuestra estructura.

Hemos creado una estructura "Estado" que usamos para informar al ilustrador por tuberías y para enviar tareas por la cola a los procesos, en ella tenemos un string con el estado de la tarea, el nivel y la parte de la tarea y el id del proceso trabajador que la realiza

Hemos tenido bastantes problemas con la inicialización de las tuberías pero lo hemos solucionado, se debía a que creabamos max parts tuberías y hay que crear una por cada parte de cada nivel y con la cola de mensajes ya que no recibimos bien los mensajes pero hemos logrado solucionarlo usando un semáforo, hemos descubierto que el fallo era porque los trabajadores nada más ser creados intentaban leer de la cola pero todavía no había nada escrito por lo que fallaban, hemos inicializado el semáforo de sort a 0 para que cuando se envíe un task se haga post y los trabajadores tienen un wait antes de empezar a leer en cola de forma que hasta que no se ha escrito algo no van a poder leer. No hemos podido solucionar los problemas que nos daba la comunicación entre tuberías por lo que las hemos dejado comentadas.

Cuando ejecutamos con small data ordena el nivel 0 y el 1 pero no más, no sabemos en qué parte se queda a la espera de un semáforo o señal que no llega, no hemos conseguido solucionarlo.

Debido a la falta de tiempo por la gran presión de todas las prácticas y exámenes del final del cuatrimestre y sintiéndolo mucho no hemos podido dedicarle el tiempo suficiente para lograr que funcione la implementación del proyecto.

Memoria Compartida

Proceso Principal

