Ethan Wilkes & Justin Sprouse

Homework 1

Part 1 Writeup

(a) Describe the parent/child process synchronization approach implemented in the

program.

- We used two pipes to synchronize the parent and child processes. By making each pipe a one way communicator, we are able to make each process wait at the appropriate times. For example, when the child process tries to read from one pipe, it cannot continue execution until the parent process finishes writing to the pipe. This keeps the two processes in sync, and each process cannot continue until it gets the information it needs.

(b) Describe the inter-process communication (IPC) approach you have selected to

exchange data (both input and output) among parent and child processes. Provide

reasoning for your selection.

- The IPC method was also the pipes. We use pipes as a one way write system to transfer information between the pipes. One pipe let the parent write and let the child read, the other pipe let the parent read and let the child write.

(c) Briefly describe a different inter-process communication approach that can be used in

this program. Provide reasoning for NOT implementing it in current program.

* Another IPC method that could have been implemented was shared memory. We chose not to use shared memory because, in our opinion, shared memory was the most useful when you had multiple processes that needed to access the shared memory. Shared memory allows multiple subscribers to access the memory after it has been written to (pub/sub). On the other hand, each pipe only allows communication between two processes. Since the pipes were simpler and our situation only needed two processes, we decided to go with pipes

Part 2 Writeup

(a) Describe the parent/child thread synchronization approach implemented in the

program.

-To synchronize the parent and child threads, we waited until we got a line, passed it into the function ran by the thread, and then waited for the thread to finish its execution and return the appropriate values before continuing the parent threads execution.

(b) Describe the approach you have selected to exchange data (both input and output)

among parent and child threads. Provide reasoning for your selection.

* To input data to the child thread, we are passing it in as a variable when the thread is created. To get output from the thread, we are waiting for the thread to finish executing and storing it’s returned value into a variable. This is a linear approach, which means we are not really getting any performance benefit, but we are meeting the requirements of the homework assignment using this method. Since it is essentially a synchronous method, passing data in as a variable and storing it after thread execution is just a really simple, easy way to pass data between threads.

Part 3 Writeup

1. Compare and contrast the two implementations: Part – 1 (process-based) and Part – 2   
   (thread-based). Is one approach (always) preferable over other? Provide reasoning for   
   your views.
   * It seems to me that it is easier to define multiple threads than it is to define multiple processes. What I mean by this is that it is easier to define each thread and specify what it should execute. This is because you just pass in the function you want the thread to execute in its creation statement. With processes, you have to do the if else statements, which is more confusing the more you create.
   * However, each approach has pros and cons. If you do not have to scale up this implementation, then processes might be a great option. I think they both exist for a reason, and one is not always preferable over the other due to each having pros and cons.