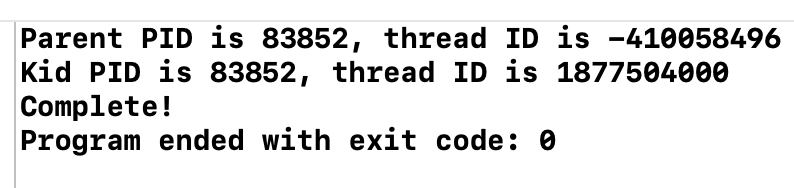
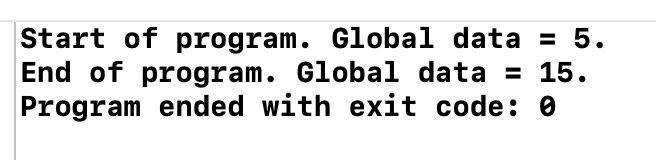
Q1)



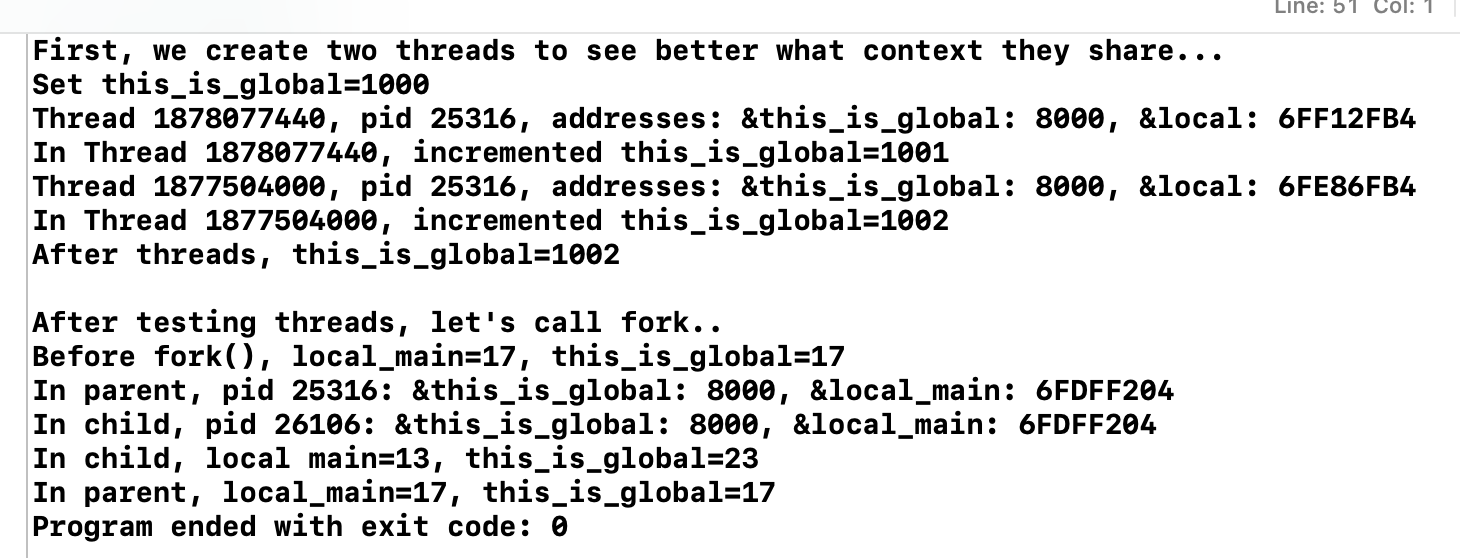
The two PIDs is same because it’s under same process, however, different threadIDs because it’s under different threads (as code written “pthread\_create”, it create a new thread that run kidfunc().

Q2)



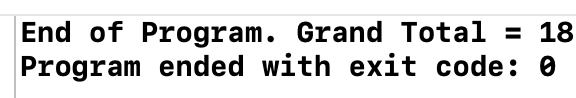
Global data will be non deterministic, because the child thread will run a function kidfunc(), and modify the global data. For instance, the pthread\_join, will wait child function to finish, hence, lastly, the global data will be 15.

Q3)

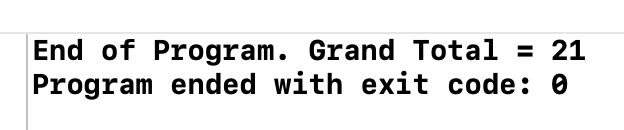


1. The variable “this\_is\_global” equals to 1002 in thread test because each thread increment this\_is\_global by 1 using ++, so there is 2 thread, and hence, the result after 2 thread is finished will be 1002
2. Yes, the address of variables in 2 threads are identical, since they are in same process, just different thread. When in same process, they share same address space. (NOPE, it’s different, local thread on own stack)
3. The modification in child process doesn’t affect parent process is because they are different process, different process have different memory space, hence their variable is not same in term of memory space.
4. This is because of “copy on write” , the address are same because the value of them are same. And the address showing, it’s virtual address, it will be same for both process since child is copied from parent. However, need to note that the physical address of both child and parent process is different.

Q4)

Text

Description automatically generated with low confidence



After 4 executions, each time, the result is different, hence, not deterministic.

This is because the program doesn’t use any synchronization mechanism, so that it may happen situation where 2 or more threads access the same global variable at the same time. Where it’s not a sequential order