|  |  |
| --- | --- |
| Name | Abdulaziz Hamid Ebrahim |
| ID | 1946282 |

# Ex1:

1. Run the above program, and observe its output:

A screen shot of a computer code

Description automatically generated with low confidence

1. Are the process ID numbers of parent and child threads the same or different? Why?

No, they are not. We know that every thread has its own unique ID, and the program’s output confirms that.

# Ex2:

1. Run the above program several times; observe its output every time. A sample output follows:

A screen shot of a computer

Description automatically generated with medium confidence

1. Does the program give the same output every time? Why?

Yes, the program's output is not guaranteed to be the same every time. It seems to be the same in this instance because my system completed the thread's job before the main thread could modify glob\_data. However, this is not a guaranteed outcome on every system or every run.

1. Do the threads have separate copies of glob\_data?  
   No, Threads do not have separate copies of glob\_data. They share memory.

# Ex3:

1. Run the above program several times and observe the outputs:

A screenshot of a computer

Description automatically generated with medium confidence

1. Do the output lines come in the same order every time? Why?

No, the output lines do not appear in the same order each time. This variability results from differences in thread creation times and priorities, which change with every run.

# Ex4:

1. Run the above program and observe its output. Following is a sample output:

A picture containing text, screenshot, font

Description automatically generated

1. Did this\_is\_global change after the threads have finished? Why?

Yes, the value of this\_is\_global does change after the threads have finished. This change occurs because it was modified after calling the join() function for all threads..

1. Are the local addresses the same in each thread? What about the global addresses?

Each thread has a unique local address. However, global addresses are identical across threads.

1. Did local\_main and this\_is\_global change after the child process has finished? Why?

Yes, the values of local\_main and this\_is\_global change after the child process has finished because the parent can modify these values after the child completes its task.

1. Are the local addresses the same in each process? What about global addresses? What happened?

Yes, in each process, the local addresses can be the same because processes do not share the same memory space and thus have separate memory addresses. This difference in memory allocation explains why processes can have the same local addresses but not global ones

# Ex5:

1. Run the above program several times and observe the outputs, until you get different results.

A screen shot of a computer program

Description automatically generated with medium confidence

1. How many times the line tot\_items = tot\_items + \*iptr; is executed?

The line "tot\_items = tot\_items + \*iptr" is executed 2,500,000 times. This count results from having 50 threads each running the command 50,000 times.

1. What values does \*iptr have during these executions?

Points to the data’s value in the tidrec struct for the thread

In other word, \*iptr points to the data's value in the tidrec struct for the current thread

1. What do you expect Grand Total to be?

The expected value for Grand Total is 63,750,000. This total is achieved by summing the numbers from 1 to 50 and then multiplying by 50,000.

1. Why you are getting different results?

Different results arise due to a race condition with the tot\_items variable. As all threads try to modify tot\_items simultaneously, the value does not increment as orderly as it should. Consequently, the output varies and does not represent the expected total.