

Submit response?

Your username (**hermann@pdx.edu**) and responses will be recorded when you submit this form.

[SWITCH ACCOUNT](#)[SUBMIT](#)

Assignment: Synchronized Sum

Work alone or with a partner. If you work with a partner, then indicate their name below AND make sure that your partner also submits this form so that they get credit.

Your assignment is to understand, build, run and improve the sum.c program. See the detailed instructions below, and see the comments in the source code.

The name, username and photo associated with your Google account will be recorded when you upload files and submit this form. Not **hermann@pdx.edu**? [Switch account](#)

* Required

Your Name *

Hermann Yepdjio

Partner's Name, if you worked with a partner. Partner should also submit this assignment, with you listed as their partner, so that they receive full credit for the assignment.

N.A

Read and understand all of code for sum.c *



yes, I understand how the code works



Build and run the original sum.c program as described in the source code. Does the original code fail for you? It should! *

☒ the original sum.c code builds, runs and fails as expected

Why does the original sum program fail? What is wrong with it? *

there is no synchronization. more than one thread access the global variable at the same time

Modify the sum.c source code so that it correctly, consistently, successfully calculates the desired sums. Test the code with 4 threads and various numbers of Rounds such as 2, 20, 200, 2000. Keep the code INEFFICIENT in that every thread updates the global Total on every iteration of its inner loop. If your code takes longer than 2 minutes to run, then stop it and run with a lower number of Rounds. *

☒ my modified sum program is functioning correctly

☒ my modified sum program is still inefficient

☒ I ran with 4 threads and with various numbers of rounds such as 2, 20, 200, 2000

Using 4 threads, run your ./sum_fix1 program with enough Rounds to cause it to run longer than 1 minute but less than 2 minutes. For what value of Rounds does your sum_fixed1 program run for longer than 1 minute but shorter than 2 minutes? (we will refer to this value as MAXROUNDS) *

40000



Upload your sum_fixed1.c program, your correct but inefficient version *

 sum_fixed1.c ✕

Make the program more efficient by having each thread calculate its local sum and then modify the shared Total variable when finished calculating its local total.

*



my program now works correctly and is more efficient. it calculates its local sum before updating the global sum

Upload sum_fixed2.c, your more-efficient version of sum.c *

 sum_fixed2.c ✕

Compare the performance of your two versions using 4 threads and 5 separate values of the "Rounds" input, ranging from 100 to MAXROUNDS.

use the "time" shell feature on linux to time each run of your program, like this:

```
time ./sum_fix1 4 100
time ./sum_fix2 4 100
time ./sum_fix1 4 1000
time ./sum_fix2 4 1000
etc....
```

note the "real" time value printed by "time" for each run.

plot the results in a graph that clearly shows the difference in performance from the fix1 version of the program to the fix2 version of the program. upload the graph.

Upload your graph *

 fix1_vs_fix2_runn... ✕



Submit

Never submit passwords through Google Forms.

This form was created inside of Portland State University. [Report Abuse](#)

Google Forms

