

Problem 2: Threads Synchronisation and Communication

Consider an application with two cooperating user threads A and B. Each consists of a sequence of three function calls whose execution is dependent as shown in Figure 2.1. The threads share access to the integer variables: A_i, B_i .

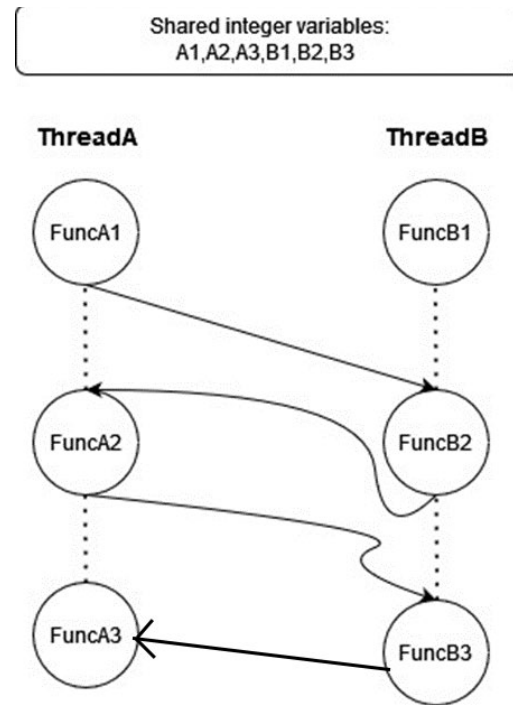


Figure 2.1

Hypothetically, we assume that the thread functions update the shared variables A_i and B_i as follows:

- FuncA1: $A1 = \sum_{i=0}^{500} i$
- FuncB1: $B1 = \sum_{i=0}^{250} i$
- FuncB2: $B2 = A1 + \sum_{i=0}^{200} i$
- FuncA2: $A2 = B2 + \sum_{i=0}^{300} i$
- FuncB3: $B3 = A2 + \sum_{i=0}^{400} i$
- FuncA3: $A3 = B3 + \sum_{i=0}^{400} i$

- Using the mathematical expressions of the functions and their execution dependence as shown in Figure 2.1, give the final correct values of the shared variables¹
- Give the additional synchronization code needed to implement the thread functions execution

¹ $\sum_{i=0}^n i = n * (n + 1) / 2$

dependence irrespective of the Operating System's threads scheduling. Active wait (i.e. `while(!cond){}` which wastes CPU cycles) or Thread `sleep` method should not be used in your solution. Explain in detail your answer.

- c) Using Java, code the application with your additional synchronisation code given in your answer to (b). The sum $\sum_{i=0}^n i$ should be collected from a static method in a dedicated utility class and should be coded using a loop.
- d) Extend the `main()` function code to run the two threads for a high number of iterations to verify the correctness of the implementation. Explain your answer.

What to submit

- Your java programs
- A report explaining the implementation of the synchronisation with reference to the relevant lines of your code along with the testing result. Maximum 3 pages.