Michael Smith

CSE 460

Homework 3

2/13/18

Total Points: 60

1. ( 10 points )  
   The aging algorithm with a = 1/2 is being used to predict run times. The previous four runs, from oldest to most recent are 40, 20, 40, and 15 msec. What is the next run time?

Using the four previous run times, the prediction is able to be produced:

(((40+20)/2+40)/2+15)/2

= ((30+40)/2+15)/2

=(35+15)/2

= 25

While using the previous two run times, the prediction becomes

(40\_15)/2

= 27.5

1. ( 10 points )   
   Measurement of a certain system have shown that the average process runs for a time T before blocking on I/O. A process switch requires a time S, which is effectively wasted ( overhead ). For round robin scheduling with quantum Q, give a formula for the CPU efficiency for each of the following.
   1. Q = infinity
   2. Q > T
   3. S < Q < T
   4. Q = S
   5. Q nearly 0

Number of times switch = (T/Q)

Time Wasted switching = (T/Q)\*S

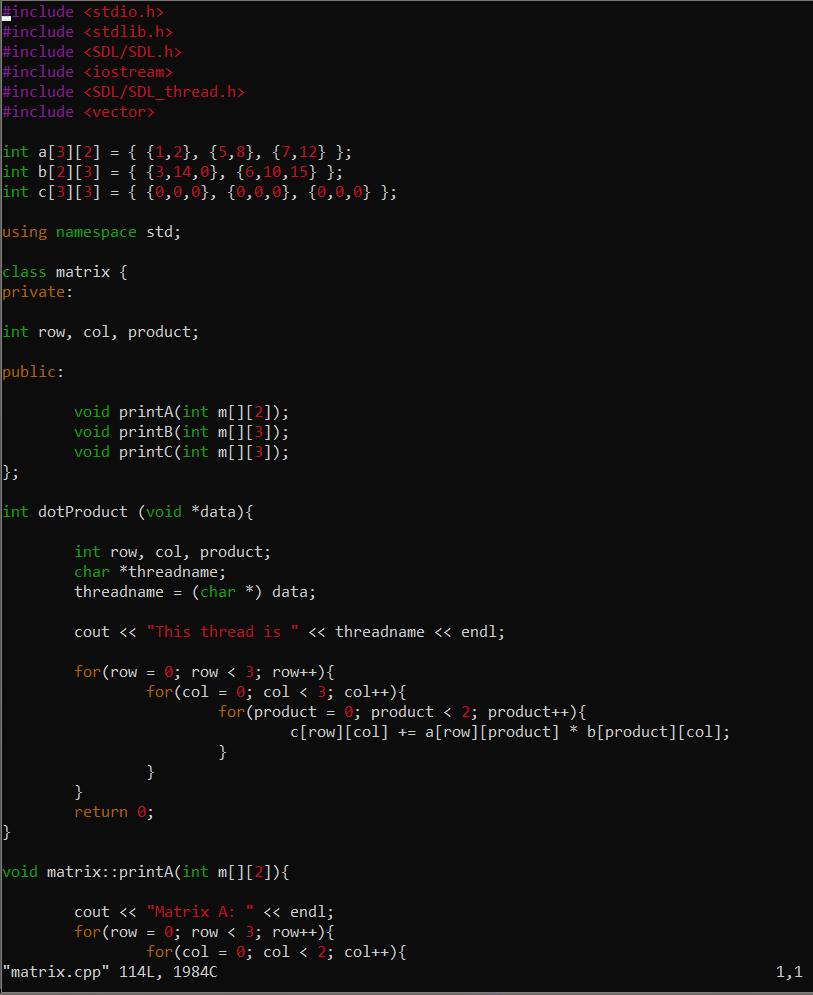
So efficiency becomes = T / (T + ST/Q)

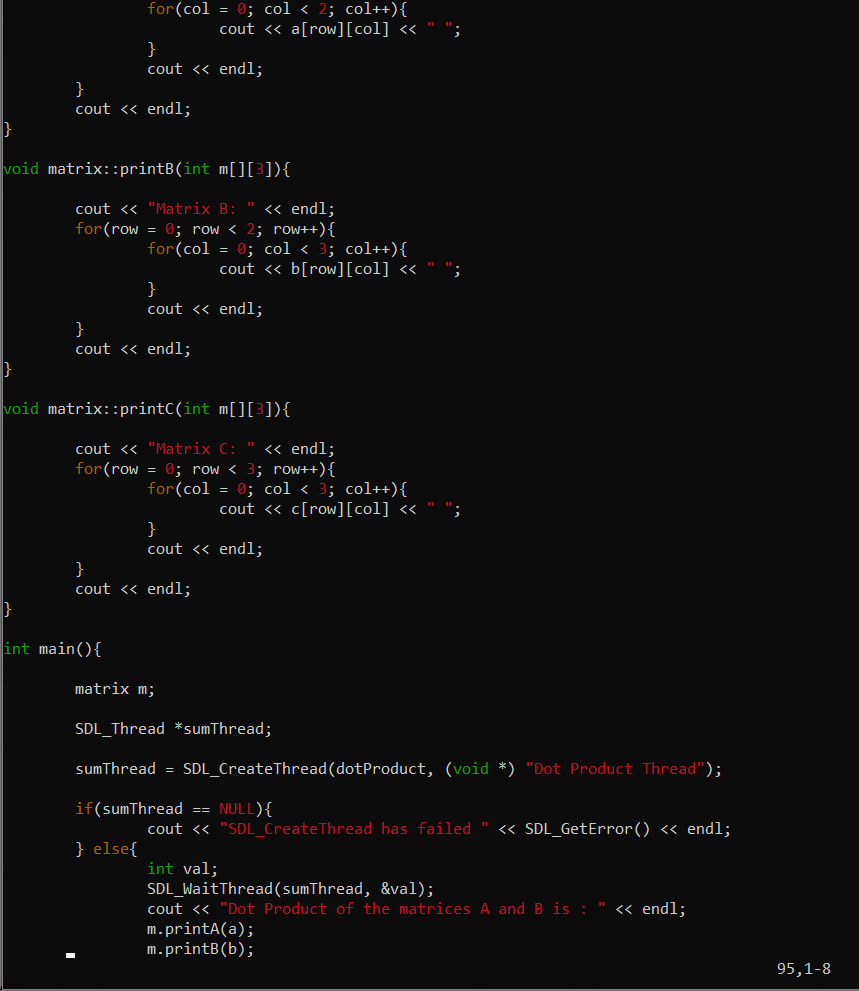
* 1. S = 0 while the CPU efficiency is T/T = 100%
  2. S = 0 while the CPU efficiency is T/T = 100%
  3. Using the efficiency equation, T / (T + ST/Q), the value would vary between 100% to 50% depending on how much less Q is than T.
  4. S = Q so the efficiency becomes T / (T + ST/Q) = T /(2T) = 50%
  5. With Q ~= 0 the efficiency becomes T / (T + ST/Q) = T / ~infinity = Undefined or 0%

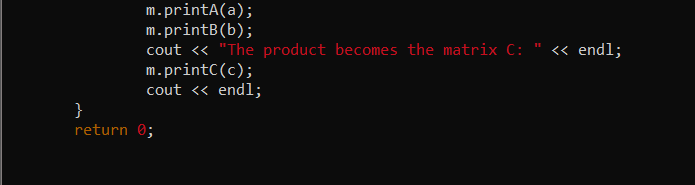
1. ( 20 points )   
   Write a multithreaded program using SDL threads or POSIX threads. The program uses a number of threads to multiply two matrices. The multiplication of an M X L matrix A and an L X N matrix B gives an M X N matrix C, and is given by the formula,



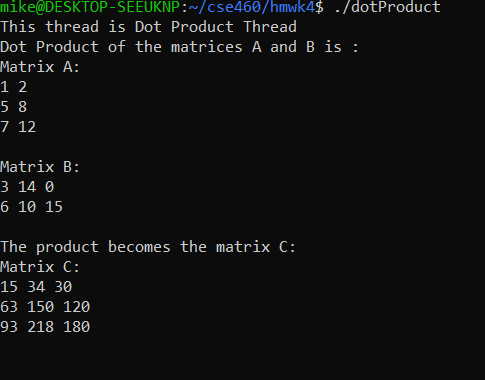
Basically, each element Cij is the dot product of the i-th row vector of A with the j-th column vector of B. The program uses one thread to calculate a dot product. Therefore, it totally needs M x N threads to calculate all the elements of matrix C.







**Output**



1. ( 20 points )   
   In the class the implementation of the readers-writers problem using SDL threads has been presented. However, the read and write tasks of the **reader** thread and the **writer** thread are not given. Implement these tasks as reading and writing of a file named *counter.txt*, which contains an integer counter.   
   A **reader** thread

reads the counter from the file, and

prints out its thread name and the value of the counter.

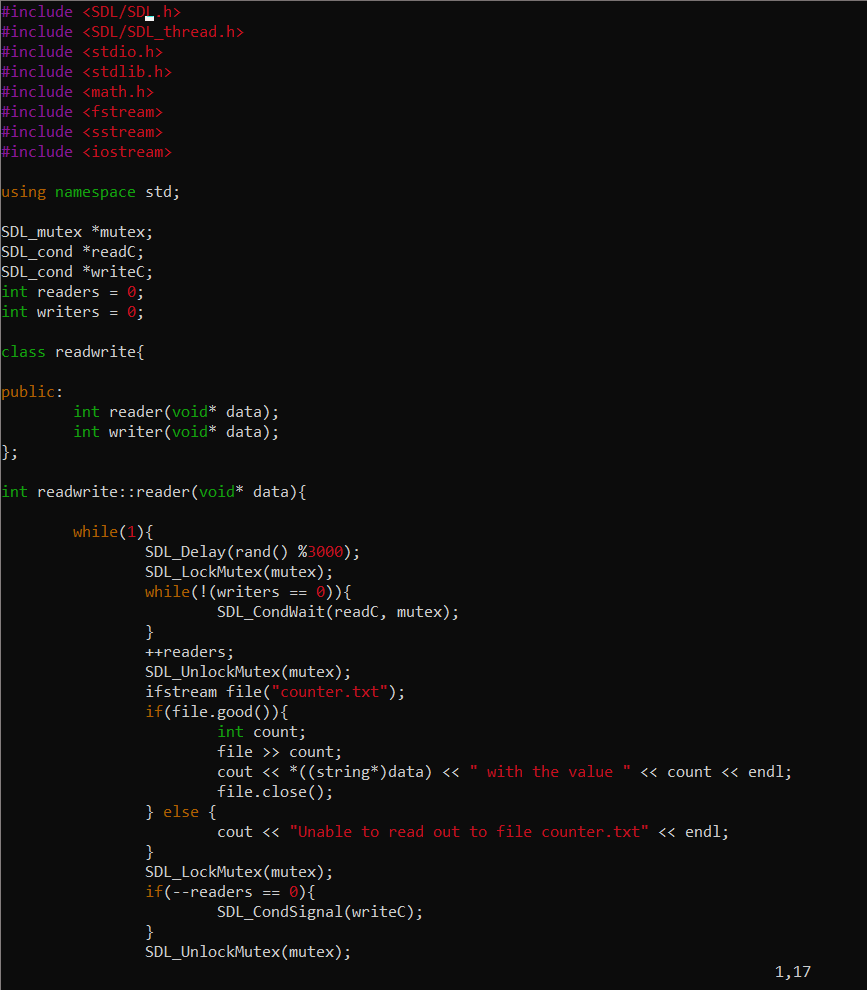
A **writer** thread

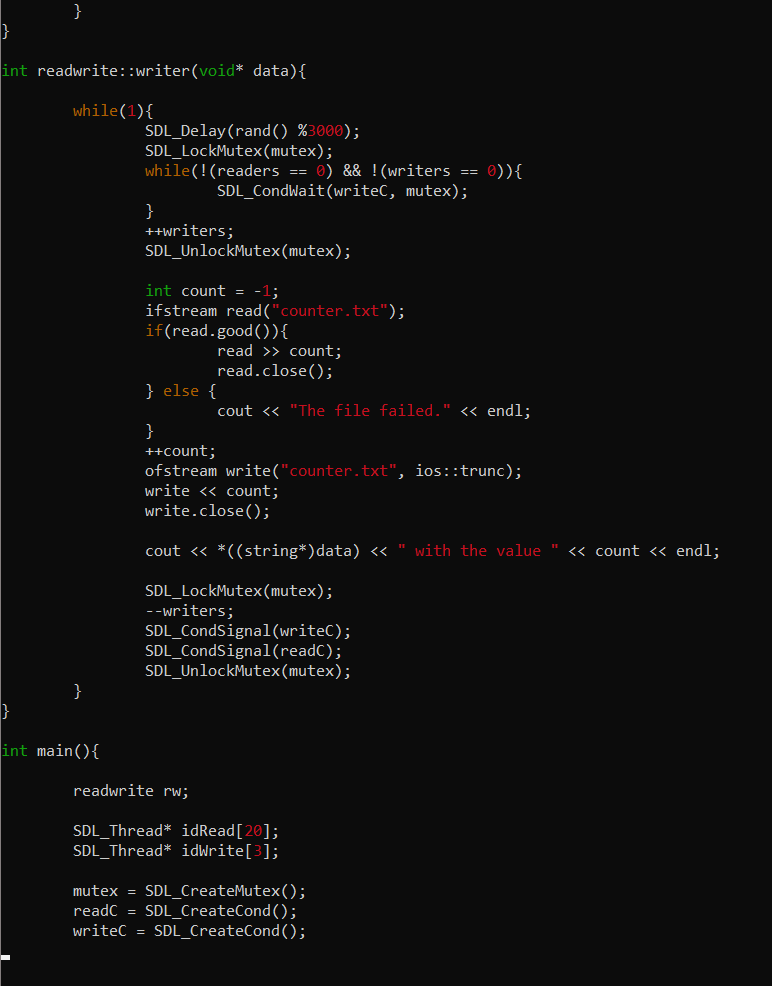
increments the value of the counter in the file,

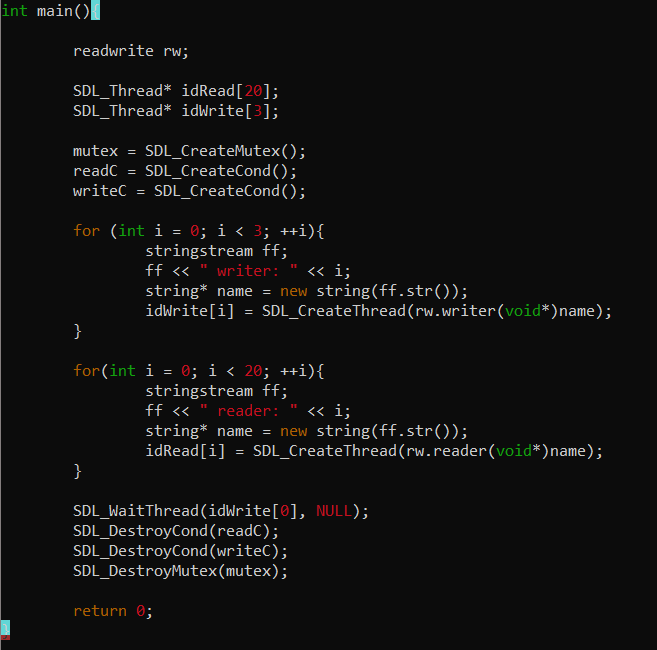
prints out its thread name and the new value of the counter.

Each thread repeats its task indefinitely in a random amount of time between 0 and 3000 ms. Your **main** program should create 20 **reader** threads and 3 **writer** threads.

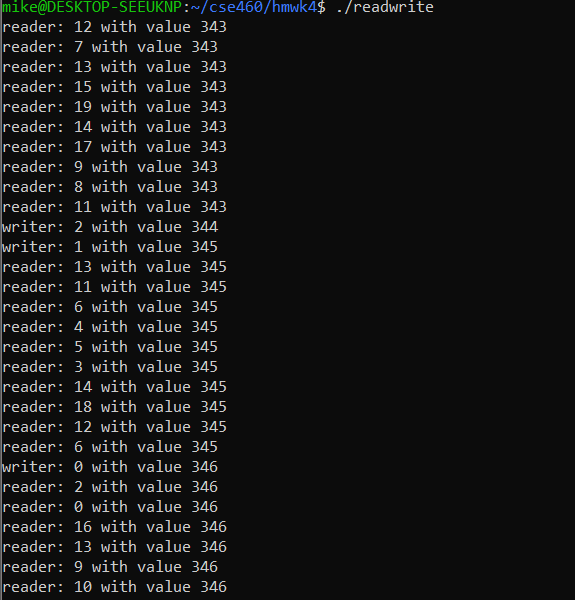
Code:







Output



Report: I was able to complete each section of the homework assignment, while producing correct results. I believe I have earned a 60/60 grade.