Krishaun Bartlett

Ryan Reynolds

CIS 345

Assignment 2

Code Description:

The main() function takes in a number from the command line converts the char string to an integer using the atoi() function. The integer is split into 3rds and the range is stored into 3 integer arrays to be passed to the Sumsqrt function. Two threads are created with the start routine Sumsqrt() and passed their respective range of squares roots they need to sum. The main thread calls the Sumsqrt() function with the range covering the remaining 3rd to be square rooted then summed. The execution time from start to finish was added to show that concurrent multithreading is present.

The Sumsqrt() function takes in a pointer of an unknown type to allow it to be passed from the thread creation. The function sums each square root from the range its passed. A mutex lock is not implemented around the for loop that sums the range to allow the main, 2nd, and 3rd threads to execute their range of square root sums concurrently. The mutex lock is placed following the for loop and prior to adding the partial sum, psum, to the global variable representing the total sum, ssum. Then a mutex unlock is placed following the sum. The global variable ssum is the only resource that is shared between the threads. The mutex lock here avoids a race condition if two or more threads exit the for loop at the same time and attempt to add their partial sum to the total sum.

Group Participation:

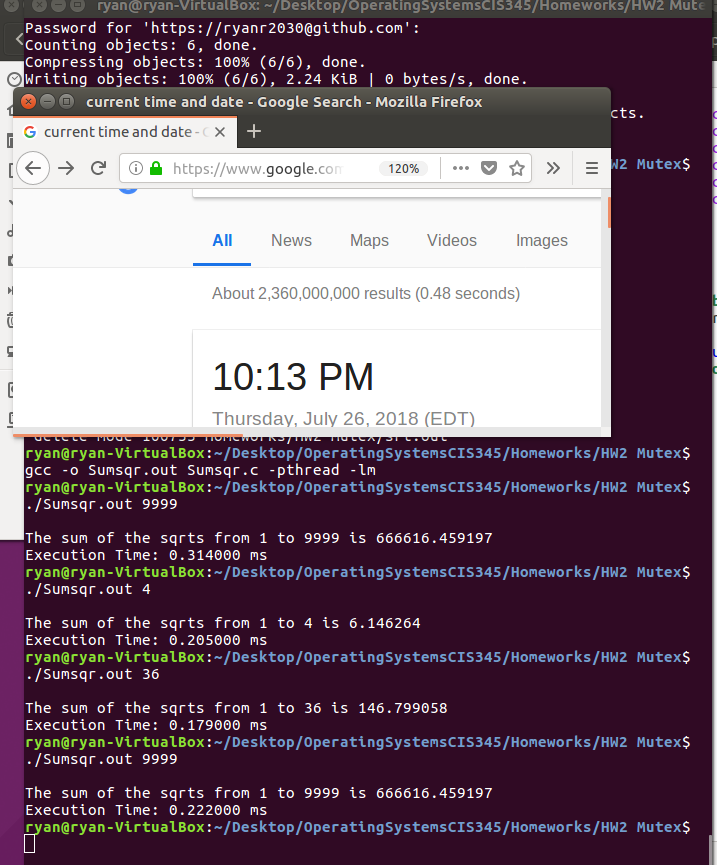
Krishaun (50%)

Wrote main() function. Converted command line argument from char string to int. Declared, created, and passed correct arguments to each thread. Wrote main function code description.

Ryan (50%)

Wrote Sumsqrt() function. Declared mutex lock and implemented to allow for concurrent sum and square root calculation for each thread. Added execution time to the main function. Debugged various warnings and segmentation faults. Wrote Sumsqrt code description.

Output:



Source Code:

|  |
| --- |
|  |
|  | #include<stdio.h>  #include <pthread.h> |
|  | #include <math.h> |
|  | #include <stdlib.h> |
|  | #include <unistd.h> |
|  | #include <time.h> |
|  |  |
|  |  |
|  | int sp1[2]; |
|  | int sp2[2]; |
|  | int sp3[2]; |
|  | double ssum=0; |
|  | pthread\_mutex\_t sumlock; |
|  |  |
|  | //function for each thread to call and sum their respective n values |
|  | void \* sumSqrt(void\* i){ |
|  | double psum=0; |
|  | int \*start=i; |
|  |  |
|  | //Calculate the square root then sum it |
|  | for(int j=start[0];j<start[1]; j++){ |
|  | double result=sqrt((double)j); |
|  | //printf("Square Root of %i is %f\n", j, result); |
|  | psum+=result; |
|  |  |
|  | } |
|  | //place mutex lock here to allow concurrent calculation of squares for each thread |
|  | //Entering Critical Region lock shared variable ssum (sum of all sqr roots) |
|  | pthread\_mutex\_lock(&sumlock); |
|  | ssum=ssum+psum; |
|  | //Exiting Critical Region unlock shared variabl ssum (sum of all sqr roots) |
|  | pthread\_mutex\_unlock(&sumlock); |
|  |  |
|  | } |
|  |  |
|  | int main(int argc, char \*argv[]){ |
|  |  |
|  | //error check remind user to enter a number in command line |
|  | if (argc!=2){ |
|  | printf("No number entered"); |
|  | return EXIT\_FAILURE; |
|  | } |
|  | clock\_t begin=clock(); |
|  | //convert char string in arg at index 1 from chars to integer |
|  | int n = atoi(argv[1]); |
|  |  |
|  | //Allocation of address space for threads t1 and t2 |
|  | pthread\_t t1, t2; |
|  |  |
|  | //1st 1/3 start and end point |
|  | sp1[0]=1; |
|  | sp1[1]=n/3+1; |
|  |  |
|  | //2nd 1/3 start and end point |
|  | sp2[0]=n/3+1; |
|  | sp2[1]=2\*n/3+1; |
|  |  |
|  | //3rd 1/3 start and end point |
|  | sp3[0]=2\*n/3+1; |
|  | sp3[1]=n+1; |
|  |  |
|  | //create thread 2 the 2nd 1/3 of n square roots to be summed |
|  | pthread\_create(&t1, NULL, sumSqrt, sp2); |
|  | //create thread 3 the 3rd 1/3 of n square roots to be summed |
|  | pthread\_create(&t2, NULL, sumSqrt, sp3); |
|  | //run the 1st 1/3 of n square roots to be summed in the main thread |
|  | sumSqrt(sp1); |
|  |  |
|  | //Wait till threads 1 and 2 are terminated to print the result |
|  | pthread\_join(t1,NULL); |
|  | pthread\_join(t2,NULL); |
|  | printf("\nThe sum of the sqrts from 1 to %i is %f\n", n, ssum); |
|  |  |
|  | //execution time calculation |
|  | clock\_t end=clock(); |
|  | double executionTime=(double)(end-begin)/CLOCKS\_PER\_SEC\*pow(10,3); |
|  | printf("Execution Time: %f ms\n",executionTime); |
|  |  |
|  | return 0; |
|  | } |
|  |  |
|  |  |