Midterm Exam

(October 19th @ 7:30 pm)

- Implement SAXPY (Single-Precision A.X Plus Y), also called Scaled Vector Addition with both *pthreads* and TBB.

$$\vec{y} \leftarrow a\vec{x} + \vec{y}$$

  - ✓ SAXPY is a combination of scalar multiplication and vector addition. It takes as input two n-element input vectors $\vec{x}$ and $\vec{y}$ (whose elements are 32-bit floating point numbers), and a scalar value a. A simple C implementation looks like this:

```
void saxpy(int n, float a, float *x, float *y) {
  for (int i = 0; i < n; i++)
      y[i] = a*x[i] + y[i];
}
```

PROBLEM 1 (60 PTS)

- Implement SAXPY using *pthreads* in C (30 pts)
  - ✓ Your code should read the parameter *nthreads* (number of threads) and the length of the vectors (n).
    - Note that *nthreads* $\in [1, n]$.
  - ✓ Parallelization: each thread i ($i \in [1, n]$) computes a slice of the output vector $\vec{y}$ with the following indices:
    - From $\left\lfloor \frac{i \times n}{nthreads} \right\rfloor$ to $\left\lfloor \frac{(i+1) \times n}{nthreads} \right\rfloor$.
  - ✓ **Input data**: Given the length n, your code should initialize the vectors $\vec{x}$ and $\vec{y}$ as per the following pseudo-code:

```
a = 1.618
for i = 0:n-1
   x[i] = sinh(i*3.416/n);   y[i] = cosh(i*3.416/n);
```

  - ✓ **Verification**: To be fully sure that your results are correct, you need to create a sequential implementation and then compare the results with those of your multi-threaded implementation. This can be achieved by computing the sum of absolute differences (SAD), which should be 0.0:

$$diff = \sum_{i=0}^{n-1} |y_p(i) - y_s(i)|$$

  where $\vec{y}_p$ and $\vec{y}_s$ are the output vectors of the multi-threaded and sequential implementations respectively.

- Compile the code and execute the application on the DE2i-150 Board. Complete Table I (take an average of ~10 executions in order to get the computation time for each case). (20 pts).
  - ✓ Example: `./mysaxpy 1000 10`
    - It will compute SAXPY on 1000-element vectors $\vec{x}$ and $\vec{y}$ using 10 threads.

TABLE I. COMPUTATION TIME (US) VS. NUMBER OF THREADS AND VECTORS LENGTH

	nthreads									
n	1	2	3	4	5	6	7	8	9	10
1,000	390	505.6	657.8	883	878.8	1243.6	1217.4	1493	1616.6	1969.6
10,000	784	739.2	888.8	941.4	1082.2	1166.4	1727.6	1934	2675.4	1783.2
100,000	4807	3205	3569.6	3247.6	2923.2	3221.4	3602.8	3964.2	4173	4338.4
1,000,000	38527.2	27265.8	18203.2	18850	19884.2	17940.4	16735	17942	20726.2	18263
2,000,000	75893.2	54747.4	36935.8	34053.2	34390.6	33449.6	32452.4	32657.4	32893	32745.2

  - ✓ Comment on your results in Table I. Is there an optimal number of threads? At what point increasing the number of threads causes an increase in processing time?

The optimal number of threads is 3-4, More threads after this causes a performance decrease for values of n >= 10,000. After 3-4 threads processing time increases.

- Take (and attach) a screenshot of the software running in the terminal for *nthreads=5, n=20*. It should show the computation times (for both the sequential and the *pthreads* implementations), the input vectors $\vec{x}$ and $\vec{y}$, the output vector $\vec{y}$, and the sum of absolute differences (SAD). Fig. 1 shows an execution example. (10 pts)

```
daniel@daniel-Inspiron-1545: ~/Dropbox/mystuff/work_ubuntu/labs/midterm/saxpy_pthreads
x(input)        y(input)                y(output)
x[0]=0.0000     y[0]=1.0000             y[0]=1.0000
x[1]=0.1716     y[1]=1.0146             y[1]=1.2923
x[2]=0.3483     y[2]=1.0589             y[2]=1.6224
x[3]=0.5351     y[3]=1.1342             y[3]=2.0000
x[4]=0.7376     y[4]=1.2426             y[4]=2.4360
x[5]=0.9617     y[5]=1.3874             y[5]=2.9433
x[6]=1.2138     y[6]=1.5727             y[6]=3.5367
x[7]=1.5015     y[7]=1.8040             y[7]=4.2335
x[8]=1.8331     y[8]=2.0881             y[8]=5.0541
x[9]=2.2183     y[9]=2.4333             y[9]=6.0224
x[10]=2.6683    y[10]=2.8496            y[10]=7.1670
x[11]=3.1964    y[11]=3.3492            y[11]=8.5210
x[12]=3.8180    y[12]=3.9468            y[12]=10.1243
x[13]=4.5512    y[13]=4.6598            y[13]=12.0237
x[14]=5.4175    y[14]=5.5091            y[14]=14.2746
x[15]=6.4423    y[15]=6.5194            y[15]=16.9430
x[16]=7.6554    y[16]=7.7205            y[16]=20.1069
x[17]=9.0924    y[17]=9.1473            y[17]=23.8588
x[18]=10.7953   y[18]=10.8416           y[18]=28.3084
x[19]=12.8139   y[19]=12.8529           y[19]=33.5859
Sum of absolute differences: 0.0000

Time measurements
***************
pthreads implementation - nthreads = 10
start: 555010 us        end: 555766 us
Elapsed time: 756 us
Sequential implementation
start: 556037 us        end: 556038 us
Elapsed time: 1 us
daniel@daniel-Inspiron-1545:~/Dropbox/mystuff/work_ubuntu/labs/midterm/saxpy_pthreads$
```

Figure 1. SAXPY execution showing three 20-element sets of values. Computation times obtained from execution on a Dell Inspiron laptop.

PROBLEM 2 (40 PTS)

- Implement SAXPY using TBB *parallel_for* in C++ (15 pts)
  - ✓ Follow the same procedure as in Problem 1, but instead of using *pthreads* to implement slices of the output vector, use *parallel_for* to fully parallelize the sequential SAXPY. Make sure to include a sequential implementation in C++.
  - ✓ Your code should read the parameter input data set size (n).

- Compile the code and execute the application on the DE2i-150 Board. Complete Table II (take an average of ~10 executions for each case). (15 pts)
  - ✓ Example: `./mysaxpy_tbb 1000`
    - It will compute SAXPY on 1000-element vectors $\vec{x}$ and $\vec{y}$.

TABLE II. COMPUTATION TIME (US) VS. VECTORS LENGTH

	n				
Implementation	10,000	100,000	1,000,000	2,000,000	5,000,000
Sequential	313.2	3764.6	29215.6	58394.2	191814
TBB	4143.8	6198.2	26679	51181.8	116464.2

  - ✓ Comment on your Table II results. Is there any point at which the TBB implementation is faster than the sequential one? Yes or No? If No, can you venture a guess as to why this is happening?

Yes, TBB is faster than Sequential for large vector sizes $n > 1{,}000{,}000$. We see that after this value, computation time is lower for TBB.

- Take (and attach) a screenshot of the software running in the terminal for *n=20*. It should show the computation times (both sequential and the TBB implementations), the input vectors $\vec{x}$ and $\vec{y}$, the output vector $\vec{y}$ and the SAD (as in Fig. 1). (10 pts)

SUBMISSION

- Demonstration: In this Midterm, the requested screenshots of the software routines running in the Terminal suffices.

- Submit to Moodle (an assignment will be created):
  - ✓ Two .zip files (one for Problem 1 and one for Problem 2).
    - Problem 1: The .zip file must contain the source files (`.c, .h, Makefile`).
    - Problem 2: The .zip file must contain the source files (`.cpp, .h, Makefile`).
  - ✓ Your Midterm work (a PDF file): This must include the completed Tables I and II, your comments, as well as the requested screenshots (2).

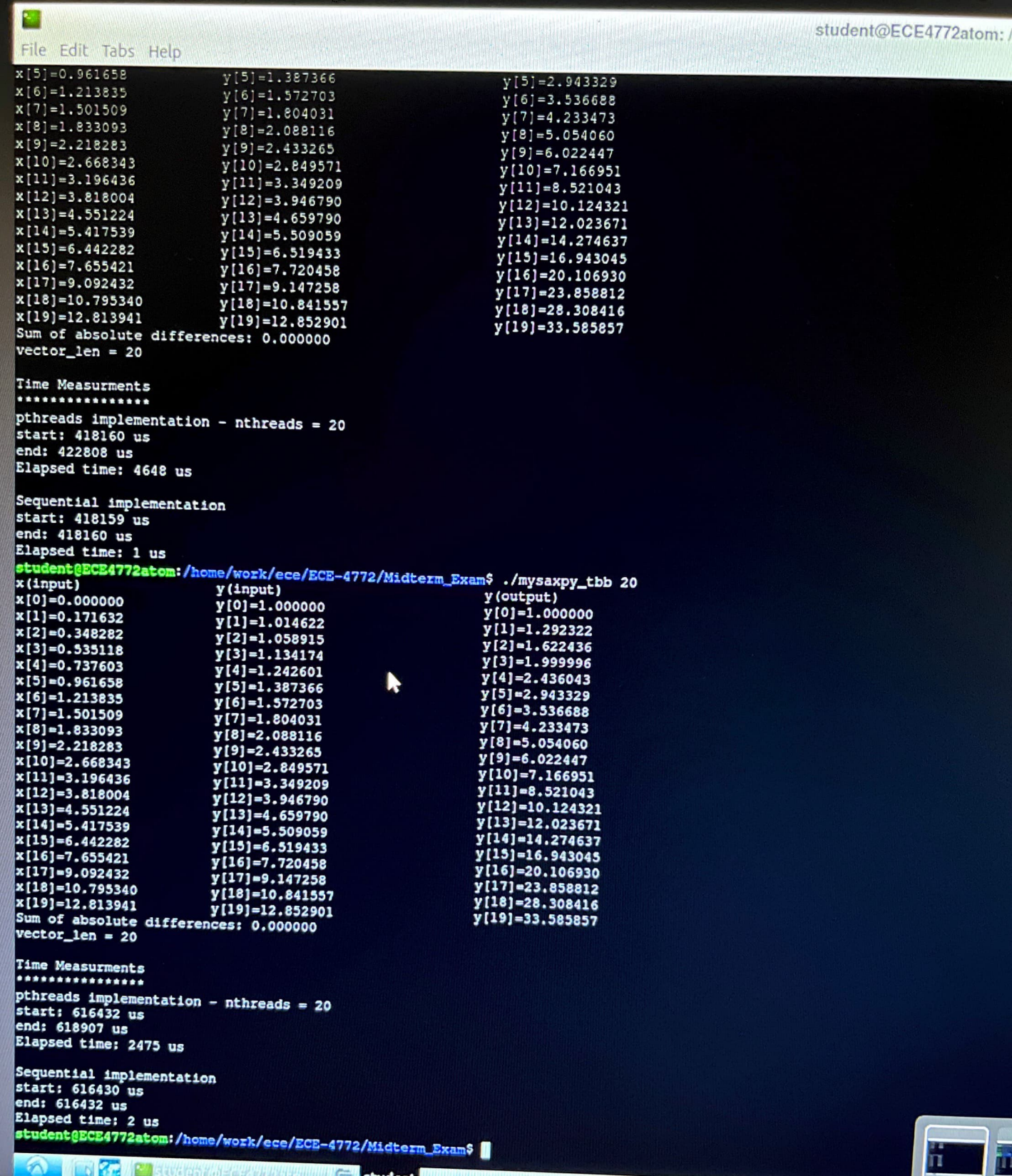
File Edit Tabs Help

```
 create mode 100644 Midtem_Exam/saxpy_pthreads.c
 create mode 100644 Midtem_Exam/saxpy_pthreads.exe
student@ECE4772atom:/home/work/ece/ECE-4772$ cd Midterm_Exam
bash: cd: Midterm_Exam: No such file or directory
student@ECE4772atom:/home/work/ece/ECE-4772$ ls
 Homework2  'Lab 1'   Lab2   Lab3   Lab4   Midtem_Exam
student@ECE4772atom:/home/work/ece/ECE-4772$ cd Midtem_Exam
student@ECE4772atom:/home/work/ece/ECE-4772/Midtem_Exam$ gcc -o mysaxpy saxpy_pthreads.c
/tmp/ccsy9mG1.o: In function `main':
saxpy_pthreads.c:(.text+0x2dc): undefined reference to `sinh'
saxpy_pthreads.c:(.text+0x321): undefined reference to `cosh'
saxpy_pthreads.c:(.text+0x3b4): undefined reference to `pthread_create'
saxpy_pthreads.c:(.text+0x3e7): undefined reference to `pthread_join'
collect2: error: ld returned 1 exit status
student@ECE4772atom:/home/work/ece/ECE-4772/Midtem_Exam$ gcc -o mysaxpy saxpy_pthreads.c -lm
/tmp/ccV9x9AS.o: In function `main':
saxpy_pthreads.c:(.text+0x3b4): undefined reference to `pthread_create'
saxpy_pthreads.c:(.text+0x3e7): undefined reference to `pthread_join'
collect2: error: ld returned 1 exit status
student@ECE4772atom:/home/work/ece/ECE-4772/Midtem_Exam$ gcc -o mysaxpy saxpy_pthreads.c -lm - pthreads
gcc: error: pthreads: No such file or directory
student@ECE4772atom:/home/work/ece/ECE-4772/Midtem_Exam$ gcc -o mysaxpy saxpy_pthreads.c -lm -pthreads
gcc: error: unrecognized command line option '-pthreads'; did you mean '-pthread'?
student@ECE4772atom:/home/work/ece/ECE-4772/Midtem_Exam$ gcc -o mysaxpy saxpy_pthreads.c -lm -pthread
student@ECE4772atom:/home/work/ece/ECE-4772/Midtem_Exam$ ls
mysaxpy  saxpy_pthreads.c  saxpy_pthreads.exe
student@ECE4772atom:/home/work/ece/ECE-4772/Midtem_Exam$ ./mysaxpy 5 20
Creating 5 Threads
x(input)                y(input)                y(output)
x[0]=0.000000           y[0]=1.000000           y[0]=1.000000
x[1]=0.171632           y[1]=1.014622           y[1]=1.292322
x[2]=0.348282           y[2]=1.058915           y[2]=1.622436
x[3]=0.535118           y[3]=1.134174           y[3]=1.999996
x[4]=0.737603           y[4]=1.242601           y[4]=2.436043
x[5]=0.961658           y[5]=1.387366           y[5]=2.943329
x[6]=1.213835           y[6]=1.572703           y[6]=3.536688
x[7]=1.501509           y[7]=1.804031           y[7]=4.233473
x[8]=1.833093           y[8]=2.088116           y[8]=5.054060
x[9]=2.218283           y[9]=2.433265           y[9]=6.022447
x[10]=2.668343          y[10]=2.849571          y[10]=7.166951
x[11]=3.196436          y[11]=3.349209          y[11]=8.521043
x[12]=3.818004          y[12]=3.946790          y[12]=10.124321
x[13]=4.551224          y[13]=4.659790          y[13]=12.023671
x[14]=5.417539          y[14]=5.509059          y[14]=14.274637
x[15]=6.442282          y[15]=6.519433          y[15]=16.943045
x[16]=7.655421          y[16]=7.720458          y[16]=20.106930
x[17]=9.092432          y[17]=9.147258          y[17]=23.858812
x[18]=10.795340         y[18]=10.841557         y[18]=28.308416
x[19]=12.813941         y[19]=12.852901         y[19]=33.585857
Sum of absolute differences: 0.000000

nthreads = 5
vector_len = 20

Time Measurments
****************
pthreads implementation - nthreads = 5
start: 341448 us
end: 342415 us
Elapsed time: 967 us

Sequential implementation
start: 341445 us
end: 341447 us
Elapsed time: 2 us
student@ECE4772atom:/home/work/ece/ECE-4772/Midtem_Exam$
```


File Edit Tabs Help
student@ECE4772atom:
x[5]=0.961658 y[5]=1.387366 y[5]=2.943329
x[6]=1.213835 y[6]=1.572703 y[6]=3.536688
x[7]=1.501509 y[7]=1.804031 y[7]=4.233473
x[8]=1.833093 y[8]=2.088116 y[8]=5.054060
x[9]=2.218283 y[9]=2.433265 y[9]=6.022447
x[10]=2.668343 y[10]=2.849571 y[10]=7.166951
x[11]=3.196436 y[11]=3.349209 y[11]=8.521043
x[12]=3.818004 y[12]=3.946790 y[12]=10.124321
x[13]=4.551224 y[13]=4.659790 y[13]=12.023671
x[14]=5.417539 y[14]=5.509059 y[14]=14.274637
x[15]=6.442282 y[15]=6.519433 y[15]=16.943045
x[16]=7.655421 y[16]=7.720458 y[16]=20.106930
x[17]=9.092432 y[17]=9.147258 y[17]=23.858812
x[18]=10.795340 y[18]=10.841557 y[18]=28.308416
x[19]=12.813941 y[19]=12.852901 y[19]=33.585857
Sum of absolute differences: 0.000000
vector_len = 20
Time Measurments
pthreads implementation - nthreads = 20
start: 418160 us
end: 422808 us
Elapsed time: 4648 us
Sequential implementation
start: 418159 us
end: 418160 us
Elapsed time: 1 us
student@ECE4772atom:/home/work/ece/ECE-4772/Midterm_Exam$ ./mysaxpy_tbb 20
x(input) y(input) y(output)
x[0]=0.000000 y[0]=1.000000 y[0]=1.000000
x[1]=0.171632 y[1]=1.014622 y[1]=1.292322
x[2]=0.348282 y[2]=1.058915 y[2]=1.622436
x[3]=0.535118 y[3]=1.134174 y[3]=1.999996
x[4]=0.737603 y[4]=1.242601 y[4]=2.436043
x[5]=0.961658 y[5]=1.387366 y[5]=2.943329
x[6]=1.213835 y[6]=1.572703 y[6]=3.536688
x[7]=1.501509 y[7]=1.804031 y[7]=4.233473
x[8]=1.833093 y[8]=2.088116 y[8]=5.054060
x[9]=2.218283 y[9]=2.433265 y[9]=6.022447
x[10]=2.668343 y[10]=2.849571 y[10]=7.166951
x[11]=3.196436 y[11]=3.349209 y[11]=8.521043
x[12]=3.818004 y[12]=3.946790 y[12]=10.124321
x[13]=4.551224 y[13]=4.659790 y[13]=12.023671
x[14]=5.417539 y[14]=5.509059 y[14]=14.274637
x[15]=6.442282 y[15]=6.519433 y[15]=16.943045
x[16]=7.655421 y[16]=7.720458 y[16]=20.106930
x[17]=9.092432 y[17]=9.147258 y[17]=23.858812
x[18]=10.795340 y[18]=10.841557 y[18]=28.308416
x[19]=12.813941 y[19]=12.852901 y[19]=33.585857
Sum of absolute differences: 0.000000
vector_len = 20
Time Measurments
pthreads implementation - nthreads = 20
start: 616432 us
end: 618907 us
Elapsed time: 2475 us
Sequential implementation
start: 616430 us
end: 616432 us
Elapsed time: 2 us
student@ECE4772atom:/home/work/ece/ECE-4772/Midterm_Exam$