

```
pi@raspberrypi: ~/8-Bit/Raejae/Project_Five/openmp1
The following NEW packages will be installed:
  libtbb-dev libtbb2
0 upgraded, 2 newly installed, 0 to remove and 178 not upgraded.
Need to get 396 kB of archives.
After this operation, 2,077 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://raspbian.mirror.constant.com/raspbian buster/main armhf libtbb2 armhf 2018~U6-4 [110 kB]
Get:2 http://raspbian.mirror.constant.com/raspbian buster/main armhf libtbb-dev armhf 2018~U6-4 [286 kB]
Fetched 396 kB in 1s (326 kB/s)
Selecting previously unselected package libtbb2:armhf.
(Reading database ... 132805 files and directories currently installed.)
Preparing to unpack .../libtbb2_2018~U6-4_armhf.deb ...
Unpacking libtbb2:armhf (2018~U6-4) ...
Selecting previously unselected package libtbb-dev:armhf.
Preparing to unpack .../libtbb-dev_2018~U6-4_armhf.deb ...
Unpacking libtbb-dev:armhf (2018~U6-4) ...
Setting up libtbb2:armhf (2018~U6-4) ...
Setting up libtbb-dev:armhf (2018~U6-4) ...
Processing triggers for libc-bin (2.28-10+rpil) ...
pi@raspberrypi:~/8-Bit/Raejae/Project_Five/openmp1 $ make
g++ -o dd_omp dd_omp.cpp -lm -fopenmp -ltbb -lrt
pi@raspberrypi:~/8-Bit/Raejae/Project_Five/openmp1 $
```

If my pictures look different. This stage of the project I decided to get a little investigative (I know late) I was curious on what SSH really was, and while playing around with it I found I could connect to PI from PC, and edit files the same, and then I could use WinSCP to transfer files. It was really helpful as transferring the files in such a manner initially seemed rather daunting. However this moment here was a moment that finally made everything sail smooth again. After compiling sequential properly I was expecting the same for OpenMp and Threads; however, I received bugs on both of them. I did a little searching and found out that I basically didn't have to library installed so after installing it to the PI I could compile properly! The above code it installing LITBTBB2

Everything went smoothly shortly after. While I followed instructions

```
pi@raspberrypi: ~/8-Bit/Raejae/Project_Five/openmp1
pi@raspberrypi:~/8-Bit/Raejae/Project_Five/cplusthreads1 $ time -p ./ dd_serial
-bash: ./: Is a directory
real 0.00
user 0.00
sys 0.00
pi@raspberrypi:~/8-Bit/Raejae/Project_Five/cplusthreads1 $ time -p ./ dd_threads
1
-bash: -p: command not found

real    0m0.006s
user    0m0.002s
sys     0m0.005s
pi@raspberrypi:~/8-Bit/Raejae/Project_Five/cplusthreads1 $ cd ..
pi@raspberrypi:~/8-Bit/Raejae/Project_Five $ cd sequential/
pi@raspberrypi:~/8-Bit/Raejae/Project_Five/sequential $ time -p ./ dd_serial
-bash: -p: command not found

real    0m0.006s
user    0m0.006s
sys     0m0.000s
pi@raspberrypi:~/8-Bit/Raejae/Project_Five/sequential $ cd ..
pi@raspberrypi:~/8-Bit/Raejae/Project_Five $ cd openmp1/
pi@raspberrypi:~/8-Bit/Raejae/Project_Five/openmp1 $ time -p ./ dd_omp 1
-bash: -p: command not found

real    0m0.006s
user    0m0.006s
sys     0m0.000s
pi@raspberrypi:~/8-Bit/Raejae/Project_Five/openmp1 $
```

Well. . . I lied when I said smooth sailing. The first mistake I made was obviously running the command for a directory outside of the one I was in. While that was questionable, I did get a little better.

```
pi@raspberrypi: ~/8-Bit/Raejae/Project_Five/cplusthreads1
pi@raspberrypi:~/8-Bit/Raejae/Project_Five/openmpl $ time -p ./dd_omp 1
max_ligand=1 nligands=120 nthreads=4
OMP defined
maximal score is 1, achieved by ligands
w o y a i i c t y n n w c h c e r a c c e r w r o i h n y c p p w r c r r o n h
y p r r t o p e p c
real 0.02
user 0.03
sys 0.00
pi@raspberrypi:~/8-Bit/Raejae/Project_Five/openmpl $ cd ..
pi@raspberrypi:~/8-Bit/Raejae/Project_Five $ cd seq
-bash: cd: seq: No such file or directory
pi@raspberrypi:~/8-Bit/Raejae/Project_Five $ cd sequential/
pi@raspberrypi:~/8-Bit/Raejae/Project_Five/sequential $ time -p ./dd_serial
^Creal 214.87
user 214.80
sys 0.00
pi@raspberrypi:~/8-Bit/Raejae/Project_Five/sequential $ cd ..
pi@raspberrypi:~/8-Bit/Raejae/Project_Five $ cd cplusthreads1/
pi@raspberrypi:~/8-Bit/Raejae/Project_Five/cplusthreads1 $ time -p ./dd_threads 1
max_ligand=1 nligands=120 nthreads=4
maximal score is 1, achieved by ligands
i e r n p p c c c n n w h c c p o i p r t o r t c e a w r n h a p r o w r r y y c h c i y
e y w o r
real 0.02
user 0.02
sys 0.00
pi@raspberrypi:~/8-Bit/Raejae/Project_Five/cplusthreads1 $
```

Implementation	Time(s)
dd_serial	214.87
dd_omp	0.02
dd_threads	0.02

For serial it never fully went through however I tried it a couple of times, and it always seemed to land on 215. I'm guessing it's slower since it doesn't engage parallel processing.

Implementation	Time(s) 2 Threads	Time(s) 3 Threads	Time(s) 4 Threads
dd_omp	0.02	0.04	0.35
dd_threads	0.02	0.05	0.16

## 2.3 Discussion Questions

1. Which approach is the fastest?

Using the thread solutions was the fastest.

2. Determine the number of lines in each file (use `wc -l`). How does the C++11 implementation compare to the OpenMP implementations?

Sequential Lines: 170

OpenMP Lines: 192

C++11 Lines: 207

3. Increase the number of threads to 5 threads. What is the run time for each?

OpenMP Real Time : 2.02

C++ Real Time: 1.47

4. Increase the maximum ligand length to 7, and rerun each program. What is the run time for each?

OpenMP Time capped out at 127.16

C++11 Time capped out at 81.79