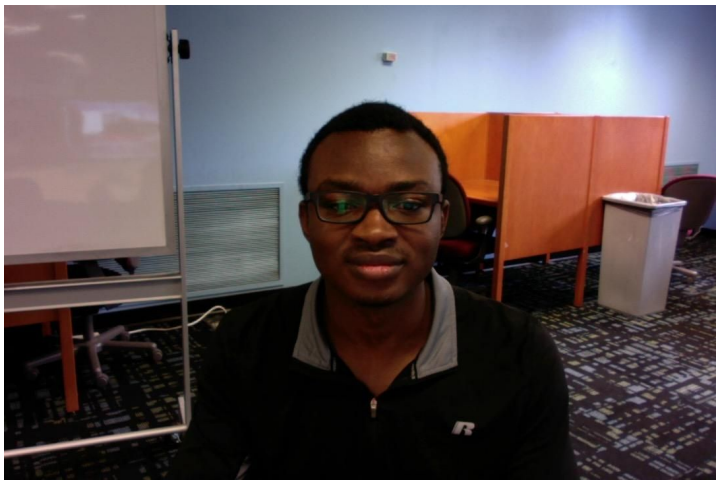
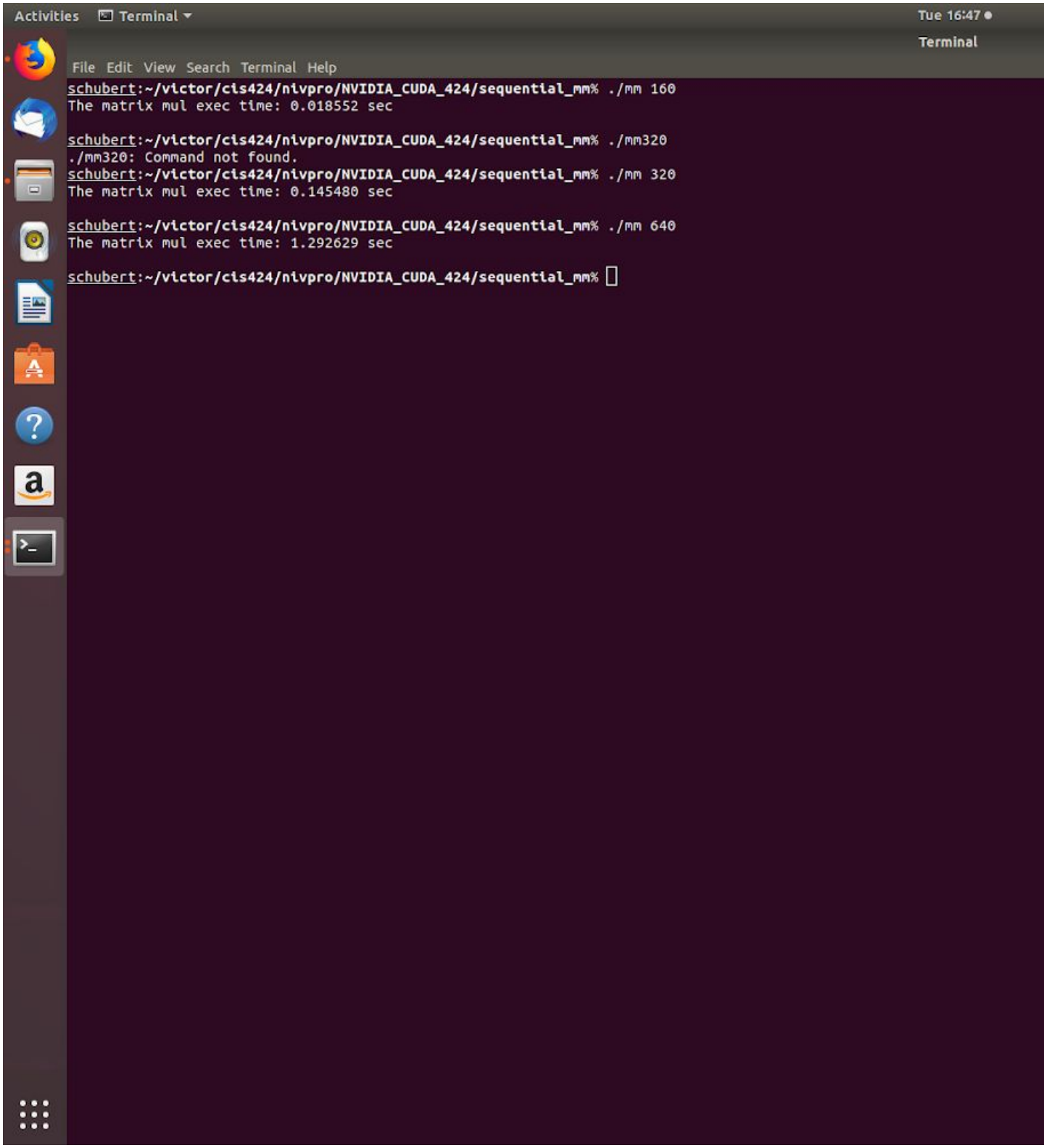


Name: Victor Ipinmoroti

Student Id: 2681928

Username: viipinmo

The project was a very interesting one, there was not much problem doing majority of the things required except for two problem. The first problem was getting the time from the mm.c file i was not entirely sure how to use the <sys/time.h> to get the time so i decided to use the time.h library so i can use he clock function then convert into seconds. And i was able to get an output.

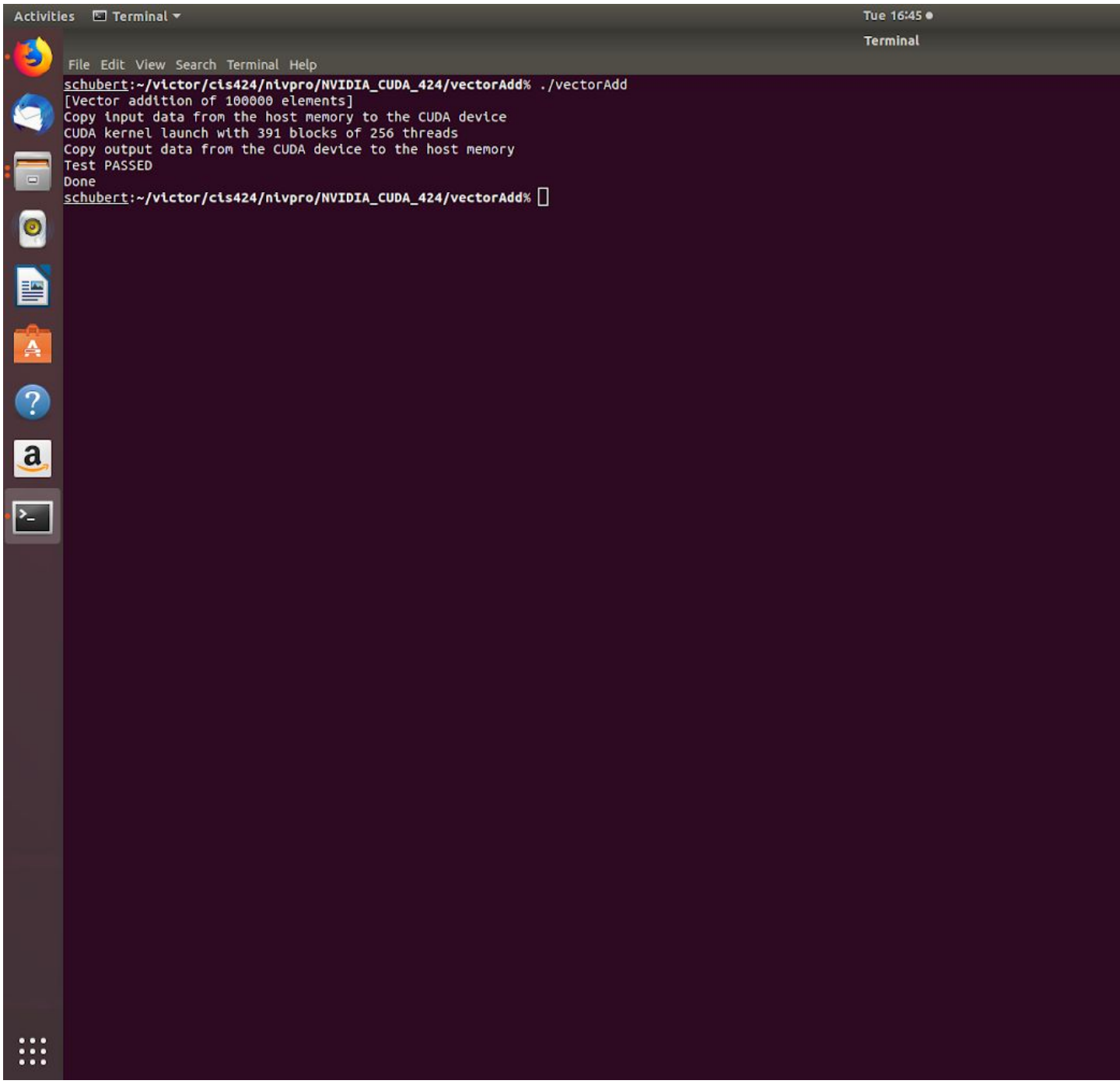

Activities
Terminal
Tue 16:47
Terminal
File Edit View Search Terminal Help
schubert:~/victor/cis424/nivpro/NVIDIA_CUDA_424/sequential_mm% ./mm 160
The matrix mul exec time: 0.018552 sec
schubert:~/victor/cis424/nivpro/NVIDIA_CUDA_424/sequential_mm% ./mm320
./mm320: Command not found.
schubert:~/victor/cis424/nivpro/NVIDIA_CUDA_424/sequential_mm% ./mm 320
The matrix mul exec time: 0.145480 sec
schubert:~/victor/cis424/nivpro/NVIDIA_CUDA_424/sequential_mm% ./mm 640
The matrix mul exec time: 1.292629 sec
schubert:~/victor/cis424/nivpro/NVIDIA_CUDA_424/sequential_mm%

mm.c output

```
schubert:~/victor/cis424/nivpro/NVIDIA_CUDA_424/deviceQuery% ./deviceQuery
./deviceQuery Starting...

 CUDA Device Query (Runtime API) version (CUDART static linking)

Detected 1 CUDA Capable device(s)

Device 0: "Quadro P620"
  CUDA Driver Version / Runtime Version          9.2 / 9.1
  CUDA Capability Major/Minor version number:    6.1
  Total amount of global memory:                 1991 MBytes (2087649280 bytes)
  ( 4) Multiprocessors, (128) CUDA Cores/MP:     512 CUDA Cores
  GPU Max Clock rate:                            1354 MHz (1.35 GHz)
  Memory Clock rate:                             2505 Mhz
  Memory Bus Width:                              128-bit
  L2 Cache Size:                                 524288 bytes
  Maximum Texture Dimension Size (x,y,z)         1D=(131072), 2D=(131072, 65536), 3D=(16384, 16384, 16384)
  Maximum Layered 1D Texture Size, (num) layers  1D=(32768), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers  2D=(32768, 32768), 2048 layers
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  2048
  Maximum number of threads per block:           1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size    (x,y,z): (2147483647, 65535, 65535)
  Maximum memory pitch:                          2147483647 bytes
  Texture alignment:                             512 bytes
  Concurrent copy and kernel execution:          Yes with 2 copy engine(s)
  Run time limit on kernels:                     Yes
  Integrated GPU sharing Host Memory:            No
  Support host page-locked memory mapping:       Yes
  Alignment requirement for Surfaces:            Yes
  Device has ECC support:                        Disabled
  Device supports Unified Addressing (UVA):      Yes
  Device PCI Domain ID / Bus ID / location ID:   0 / 1 / 0
  Compute Mode:
     < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 9.2, CUDA Runtime Version = 9.1, NumDevs = 1, Device0 = Quadro P620
Result = PASS
schubert:~/victor/cis424/nivpro/NVIDIA_CUDA_424/deviceQuery%
```

getting the device query was not a problem and was relatively straightforward as one can see there 512 cuda cores and gpu is clocked at 1.35 ghz

vectorADD

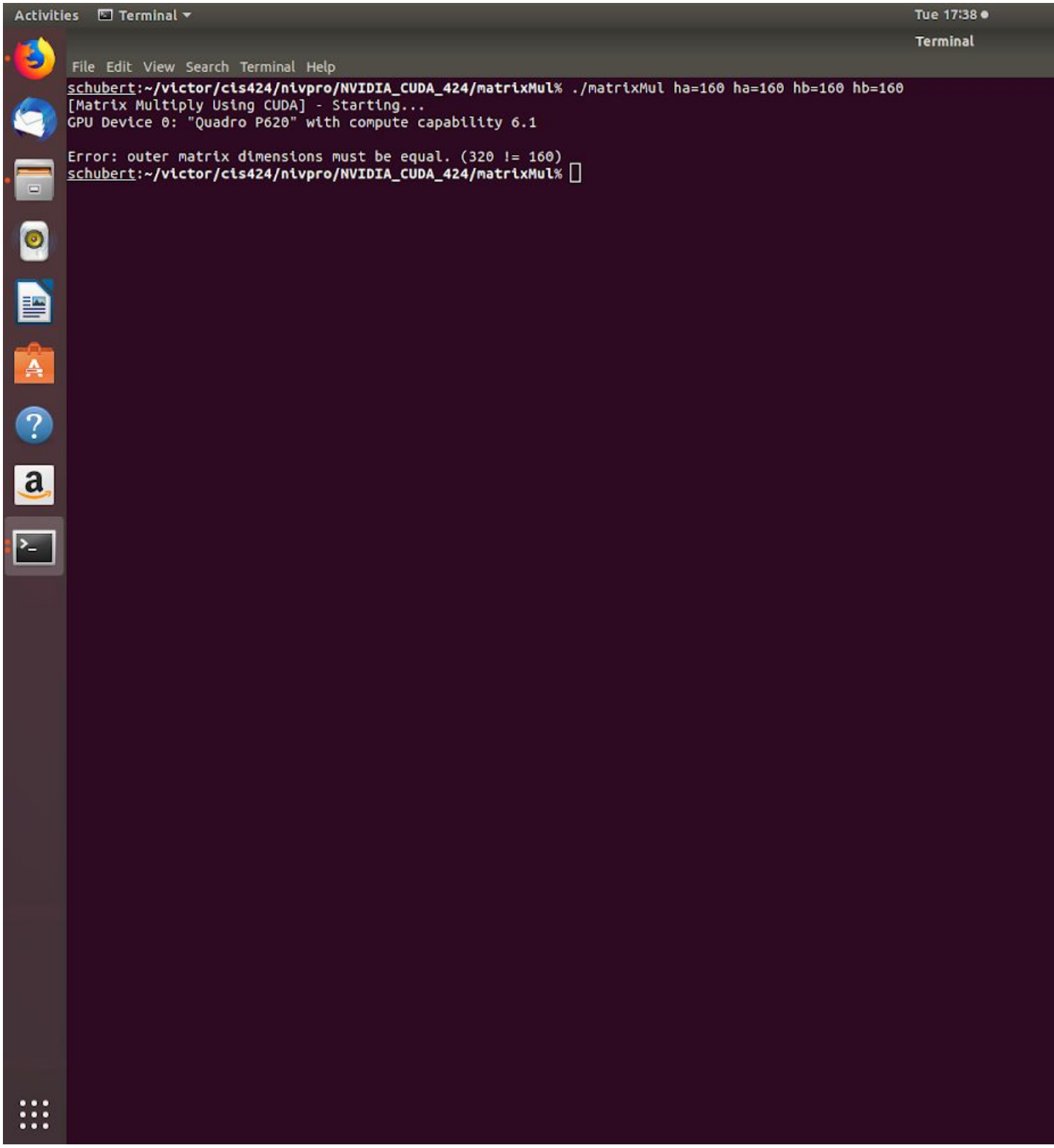


The vectoradd did not give any problem i followed exactly what was said in class and it came out as expected.

The only real problem i encountered was with the gpu matrix multiplication, this i believe is due to the way i am giving the input. the only valid nbyn matrix i could get was 320 by 320. as shown below.

```
schubert:~/victor/cis424/nivpro/NVIDIA_CUDA_424/matrixMul% ./matrixMul ha=320 ha=320 hb=320 hb=320
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "Quadro P620" with compute capability 6.1

MatrixA(320,320), MatrixB(640,320)
Computing result using CUDA Kernel...
done
Performance= 167.36 GFlop/s, Time= 0.783 msec, Size= 131072000 Ops, WorkgroupSize= 1024 threads/block
Checking computed result for correctness: Result = PASS

NOTE: The CUDA Samples are not meant for performance measurements. Results may vary when GPU Boost is enabled.
schubert:~/victor/cis424/nivpro/NVIDIA_CUDA_424/matrixMul% ./matrixMul ha=160
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "Quadro P620" with compute capability 6.1

MatrixA(320,160), MatrixB(640,320)
Computing result using CUDA Kernel...
done
Performance= 160.97 GFlop/s, Time= 0.407 msec, Size= 65536000 Ops, WorkgroupSize= 1024 threads/block
Checking computed result for correctness: Result = PASS

NOTE: The CUDA Samples are not meant for performance measurements. Results may vary when GPU Boost is enabled.
schubert:~/victor/cis424/nivpro/NVIDIA_CUDA_424/matrixMul% ./matrixMul ha=640
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "Quadro P620" with compute capability 6.1

MatrixA(320,640), MatrixB(640,320)
Computing result using CUDA Kernel...
done
Performance= 167.93 GFlop/s, Time= 1.561 msec, Size= 262144000 Ops, WorkgroupSize= 1024 threads/block
Checking computed result for correctness: Result = PASS

NOTE: The CUDA Samples are not meant for performance measurements. Results may vary when GPU Boost is enabled.
schubert:~/victor/cis424/nivpro/NVIDIA_CUDA_424/matrixMul% ha=160 ha=160 hb=160 hb=160
ha=160: Command not found.
schubert:~/victor/cis424/nivpro/NVIDIA_CUDA_424/matrixMul% ./matrixMul ha=160 ha=160 hb=160 hb=160
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "Quadro P620" with compute capability 6.1

Error: outer matrix dimensions must be equal. (320 != 160)
schubert:~/victor/cis424/nivpro/NVIDIA_CUDA_424/matrixMul% ./matrixMul ha=160
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "Quadro P620" with compute capability 6.1

MatrixA(320,160), MatrixB(640,320)
Computing result using CUDA Kernel...
done
Performance= 159.92 GFlop/s, Time= 0.410 msec, Size= 65536000 Ops, WorkgroupSize= 1024 threads/block
Checking computed result for correctness: Result = PASS

NOTE: The CUDA Samples are not meant for performance measurements. Results may vary when GPU Boost is enabled.
schubert:~/victor/cis424/nivpro/NVIDIA_CUDA_424/matrixMul%
```

and even then the matrix don't seem to be square matrices if i use anything else i get error like so



giving this i do not think the time speed calculation is valid but here are the result

45.482 160 matrix

185.8 320 matrix

828.08 640 matrix

conclusion

while we encounter hiccups that really affect our calculation it is still a good approximation that the gpu is faster than the cpu at least for matrix multiplication task as 320 by 640 is still a large matrix that is done faster than the cpu 320 by 320.