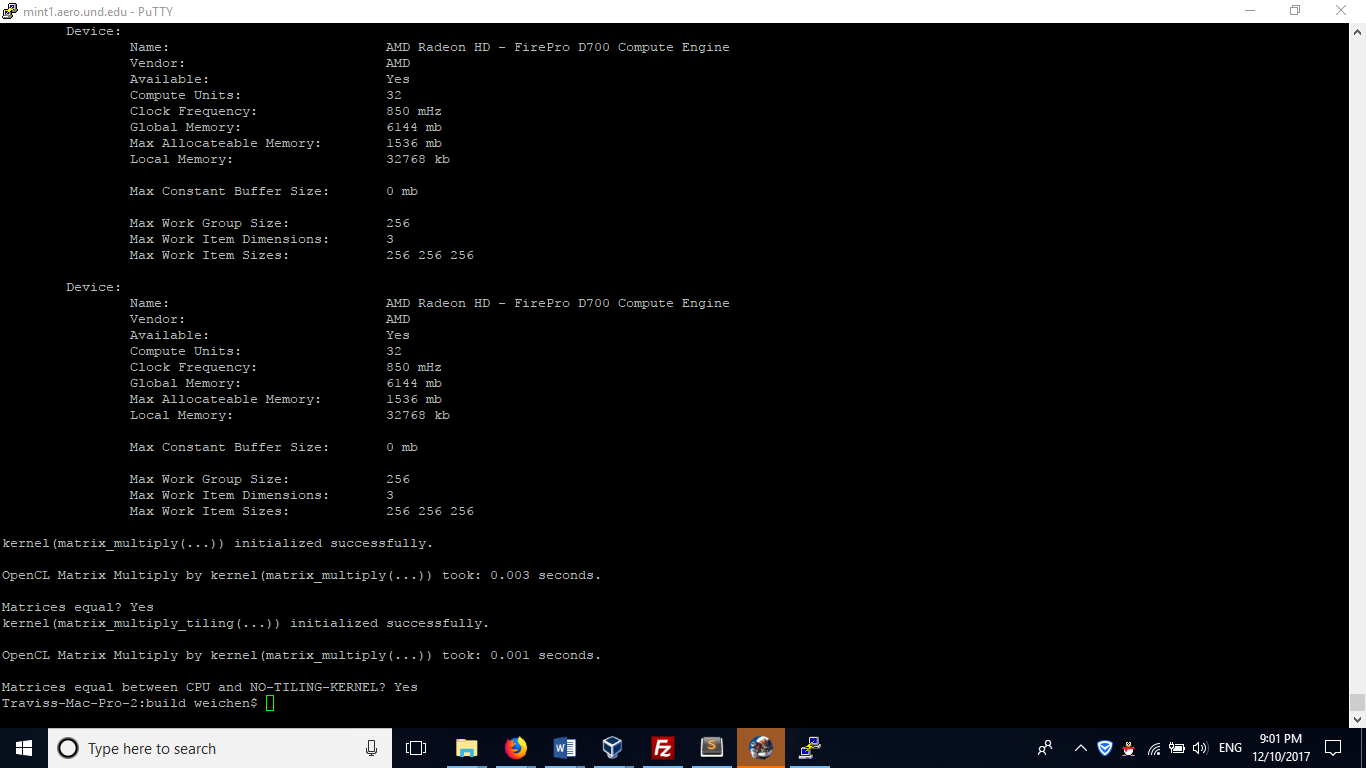
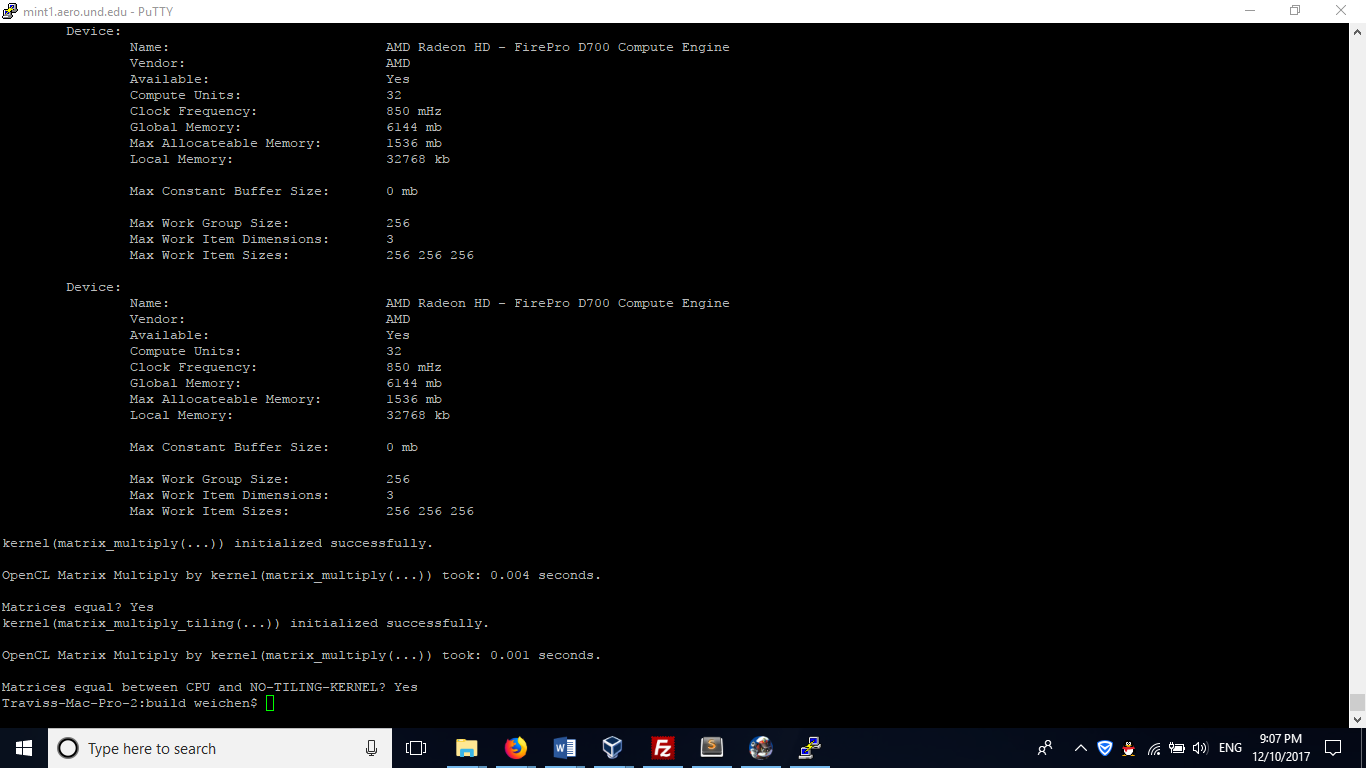
**Experiment 1: two 256\*256 matrix with kernel block size 16**

(command: ./matrix\_op -m 256 -n 256 -p 256 -q 256)



So, get the non-tiling running time is 0.003 seconds, tiling running time is 0.001 seconds, so tiling is faster than non-tiling method.

Try run with the same matrix and kernel block size **second** time:



the non-tiling running time is 0.004 seconds, tiling running time is 0.001 seconds, still tiling is faster

I then continuously run this for 7-8 times, and get the results:

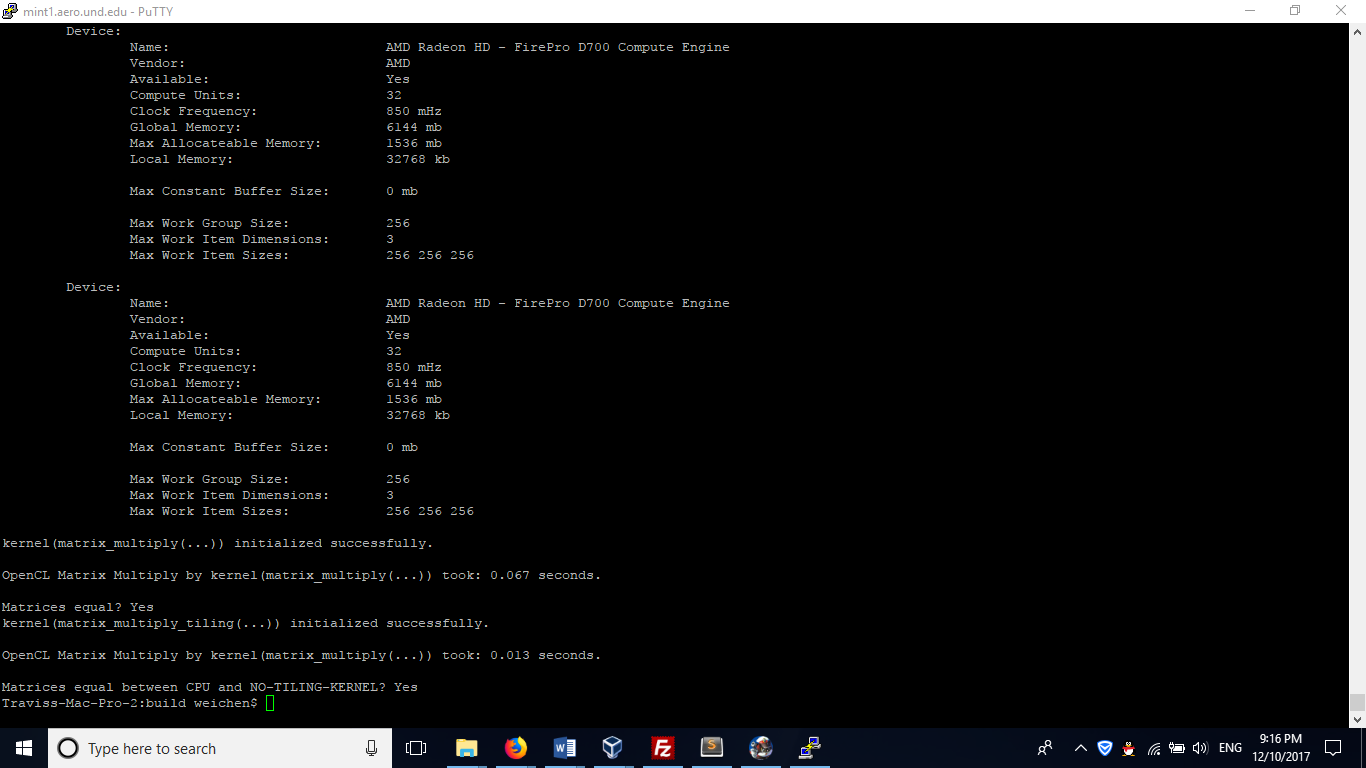
Non-tiling: 0.003/0.004 seconds

Tiling: 0.001 seconds

Conclusion: **Tiling is a little bit faster than Non-tiling**

**Experiment 2: With Larger matrix compared to Experiment 1**

This one increases the matrix size: run two 1024\*1024 matrix with kernel block size 16



We can see Non-tiling is 0.067 seconds while tiling is 0.013 seconds, and tiling is much faster than Non-tiling.

I then run this case for 7-8 times, all the results are same. So, just skip the same screenshots.

*So, we can get the conclusion from experiment 1 and experiment 2:*

**Tiling method is faster than Non-tiling method, and the running time difference becomes remarkable by increasing matrix size. Tiling method is more effective especially with large-scale matrix calculation.**

**Furthermore, we can see: using local memory can provide well-equipped memory access locality for large-scale matrix calculation, which leads to the dramatic improvement of calculation speed compared to purely using global memory.**