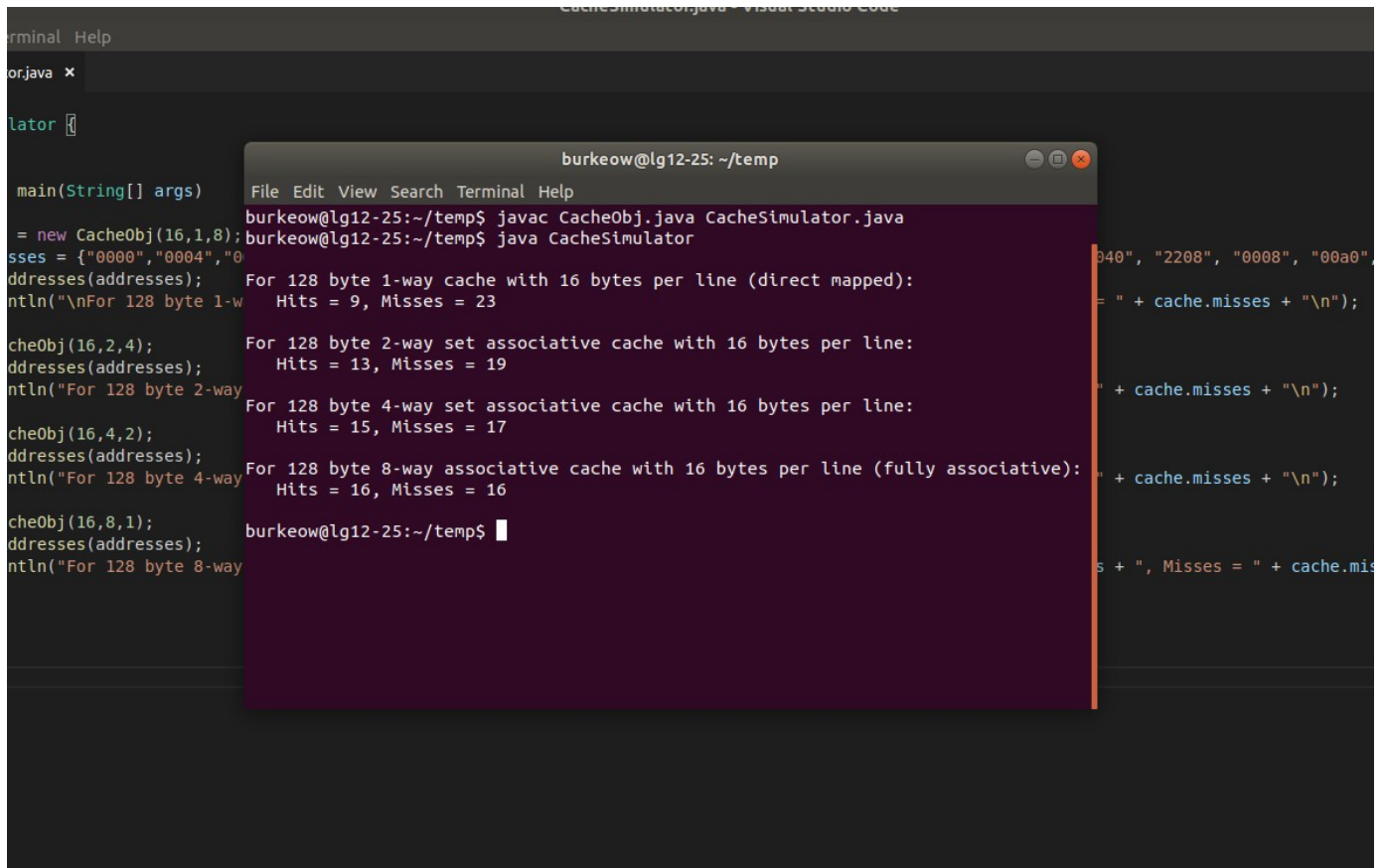


CS3021 Tutorial 6

Below is proof that my program generates the correct results for Q2:



```
burkeow@lg12-25: ~/temp
File Edit View Search Terminal Help
burkeow@lg12-25:~/temp$ javac CacheObj.java CacheSimulator.java
burkeow@lg12-25:~/temp$ java CacheSimulator

main(String[] args)
= new CacheObj(16,1,8);
sses = {"0000", "0004", "0
ddresses(addresses);
ntln("\nFor 128 byte 1-w
    Hits = 9, Misses = 23

For 128 byte 1-way cache with 16 bytes per line (direct mapped):
    Hits = 9, Misses = 23

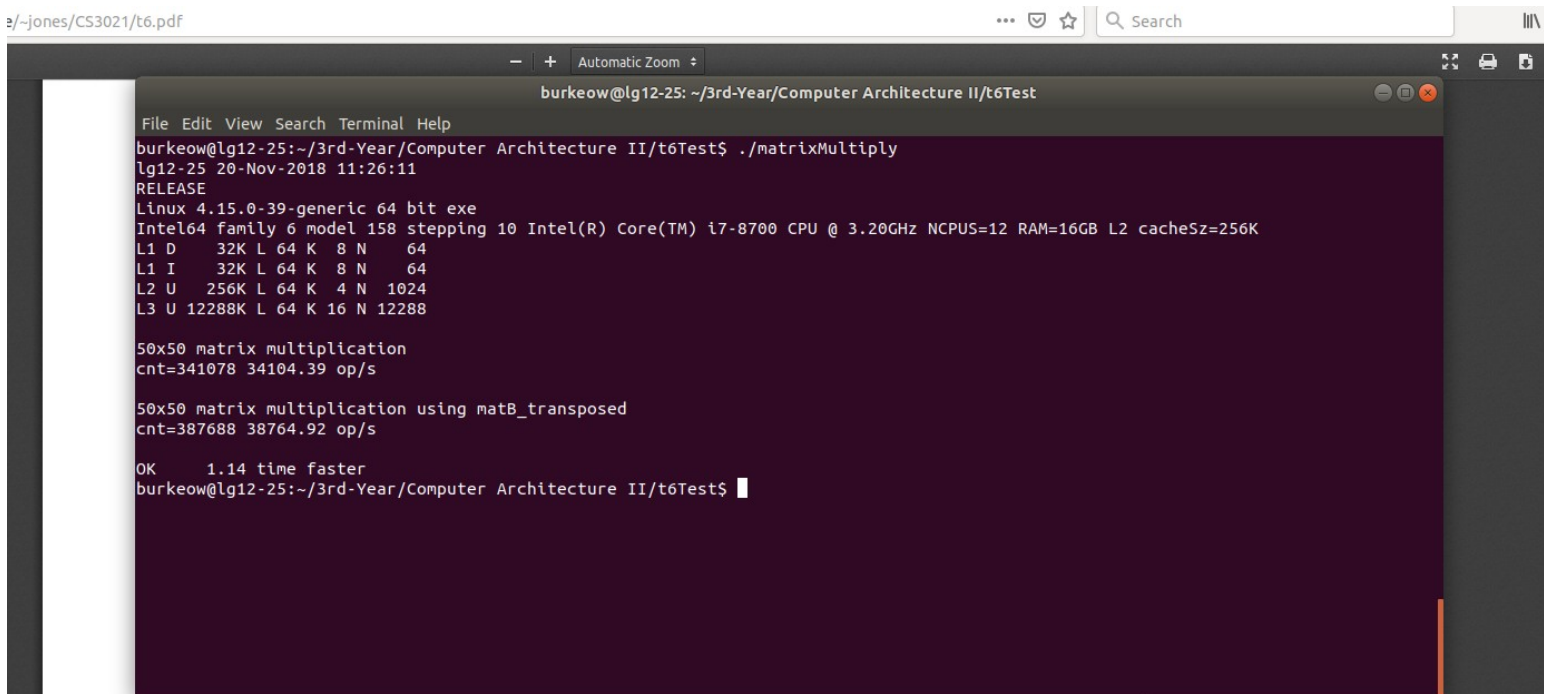
For 128 byte 2-way set associative cache with 16 bytes per line:
    Hits = 13, Misses = 19

For 128 byte 4-way set associative cache with 16 bytes per line:
    Hits = 15, Misses = 17

For 128 byte 8-way associative cache with 16 bytes per line (fully associative):
    Hits = 16, Misses = 16

burkeow@lg12-25:~/temp$
```

Below is a screenshot of the output for Q3:



```
burkeow@lg12-25: ~/3rd-Year/Computer Architecture II/t6Test
File Edit View Search Terminal Help
burkeow@lg12-25:~/3rd-Year/Computer Architecture II/t6Test$ ./matrixMultiply
lg12-25 20-Nov-2018 11:26:11
RELEASE
Linux 4.15.0-39-generic 64 bit exe
Intel64 family 6 model 158 stepping 10 Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz NCPUS=12 RAM=16GB L2 cacheSz=256K
L1 D 32K L 64 K 8 N 64
L1 I 32K L 64 K 8 N 64
L2 U 256K L 64 K 4 N 1024
L3 U 12288K L 64 K 16 N 12288

50x50 matrix multiplication
cnt=341078 34104.39 op/s

50x50 matrix multiplication using matB_transposed
cnt=387688 38764.92 op/s

OK 1.14 time faster
burkeow@lg12-25:~/3rd-Year/Computer Architecture II/t6Test$
```

This Tutorial is worth double marks (marked out of 20). Submit the source code for Q2 and Q3 and one .pdf document containing (i) evidence that your program generates the correct results for Q2 and (ii) a screen shot of the output of Q3. Please submit your answer via Blackboard by 9am Fri 30-Nov-18.