CSE 3100: PS 7
Instructor: Kriti Bhargava

What you need to implement!

Open the file `matrix.c`. This file contains a set of functions to create and multiply matrices. You must implement the following functions:

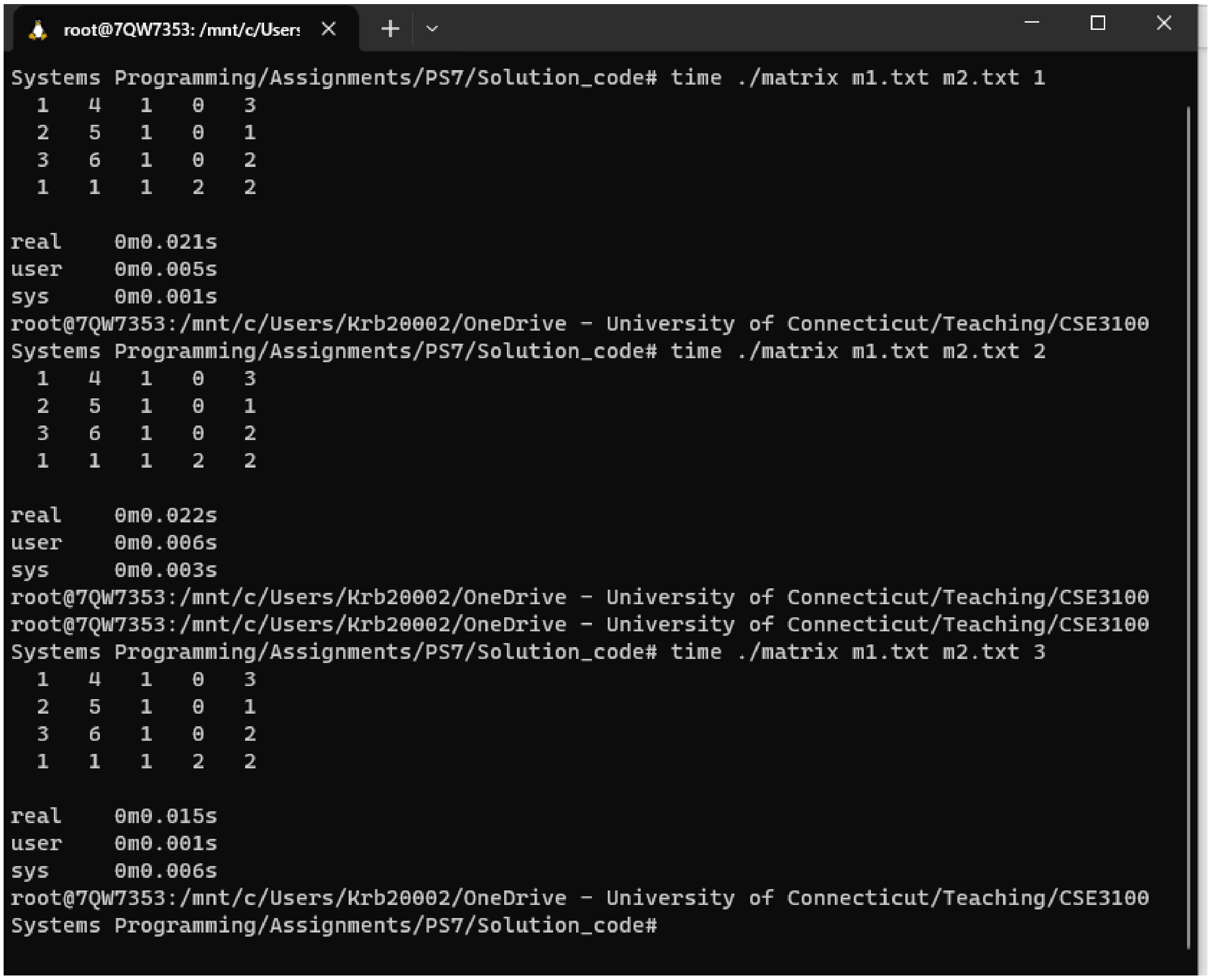
1. `size_t sizeMatrix(int r,int c)` -
   returns the number of bytes needed to hold a matrix of r x c values (type int).
2. `Matrix* makeMatrix(int r,int c)` -
   allocates space to hold a matrix of r x c values (type int).
3. `Matrix* makeMatrixMap(void* zone,int r,int c)` -
   allocates space for a matrix of r x c from a given memory block at the address zone.
4. `void freeMatrix(Matrix* m)` - deallocates the space used by matrix m.
5. `void printMatrix(Matrix* m)` - prints the matrix on the standard output. One row per line, values for the row separated by white spaces.
6. `Matrix* multMatrix(Matrix* a,Matrix* b,Matrix* into)` -
   computes the product of A * B and stores the result in \`into\`.
   The computation here is sequential and is only meant to be used as a check for your parallel code.
7. `Matrix* parMultMatrix(int nbW,sem_t* sem,Matrix* a,Matrix* b,Matrix* into)` – computes the product of A * B and stores the result in \`into\`.
   The computation is done in parallel with nbW worker processes that you fork from here.

The functions to read a value from a file descriptor (`readValue`) and reading a matrix from a file (`readMatrix`) have been implemented for you.

A test file, `main.c`, has already been implemented for you. Read through the comments to see how the computation is set up using shared memory and semaphores.

Below are some sample sessions of running the program with 1, 2 and 3 processes. You can check the time it takes to compute the multiplication of matrices using the `time` command as shown. Ideally, the time taken to compute the multiplication should be less when multiple processes are involved. The difference is apparent for large matrices. For small matrices, however, the difference is not always realized due to context switching (that we shall discuss later).

CSE 3100: PS 7
Instructor: Kriti Bhargava



```
root@7QW7353: /mnt/c/User:
Systems Programming/Assignments/PS7/Solution_code# time ./matrix m1.txt m2.txt 1
   1   4   1   0   3
   2   5   1   0   1
   3   6   1   0   2
   1   1   1   2   2

real    0m0.021s
user    0m0.005s
sys     0m0.001s
root@7QW7353:/mnt/c/Users/Krb20002/OneDrive - University of Connecticut/Teaching/CSE3100
Systems Programming/Assignments/PS7/Solution_code# time ./matrix m1.txt m2.txt 2
   1   4   1   0   3
   2   5   1   0   1
   3   6   1   0   2
   1   1   1   2   2

real    0m0.022s
user    0m0.006s
sys     0m0.003s
root@7QW7353:/mnt/c/Users/Krb20002/OneDrive - University of Connecticut/Teaching/CSE3100
root@7QW7353:/mnt/c/Users/Krb20002/OneDrive - University of Connecticut/Teaching/CSE3100
Systems Programming/Assignments/PS7/Solution_code# time ./matrix m1.txt m2.txt 3
   1   4   1   0   3
   2   5   1   0   1
   3   6   1   0   2
   1   1   1   2   2

real    0m0.015s
user    0m0.001s
sys     0m0.006s
root@7QW7353:/mnt/c/Users/Krb20002/OneDrive - University of Connecticut/Teaching/CSE3100
Systems Programming/Assignments/PS7/Solution_code#
```