

# CMSC 257 Lab 7

## Shared Memory & Process Control

### Summary

- Create a new directory: “lab7”. All the files for this lab should be saved in this directory only.
- Copy the files shmtest.c, shmtest2.c, shmtest2d.c from /home/dahlbergra/example\_code/for\_lab/ directory to your lab7 directory.

### Checkpoint 1

Consider **shmtest.c**.

Rename the shared memory partition “/brs\_memory” to your “username\_memory”. Figure out how to compile this code.

Execute this code and explain what it is doing.

*What are the command line parameters doing in this code?*

### Checkpoint 2

Consider **shmtest2.c**.

Rename the shared memory partition “/brs\_memory” to your “username\_memory”.

Execute this code and explain what it is doing.

*How is the process control part different in this code than shmtest.c? How much shared memory was allocated in this code?*

### Checkpoint 3

Consider **shmtest2d.c**.

Rename the shared memory partition “/brs\_memory” to your “username\_memory”.

Execute this code and explain what it is doing.

*How much shared memory was allocated in this code?*

### Checkpoint 4

Consider **shmtest2d.c**.

Rename the shared memory partition “/brs\_memory” to your “username\_memory”.

Allocate a shared memory partition for a 2-d array: `mem[n][n]`; `n` should be passed as a command line argument in addition to the number of threads.

Convert the 2-d array implementation to a 1-d array representation. You will still allocate shared memory for `n2` ints, and each reference to **mem[i][j]** should change to **mem[i\*n+j]** in the 1-d array representation.

## Checkpoint 5

Make sure you don't have any rogue processes running.

Do a “**ps -eaf | grep username**” to list all processes under your username. If you find rogue processes still being executed, kill them using:

**kill -9 <process\_id>**