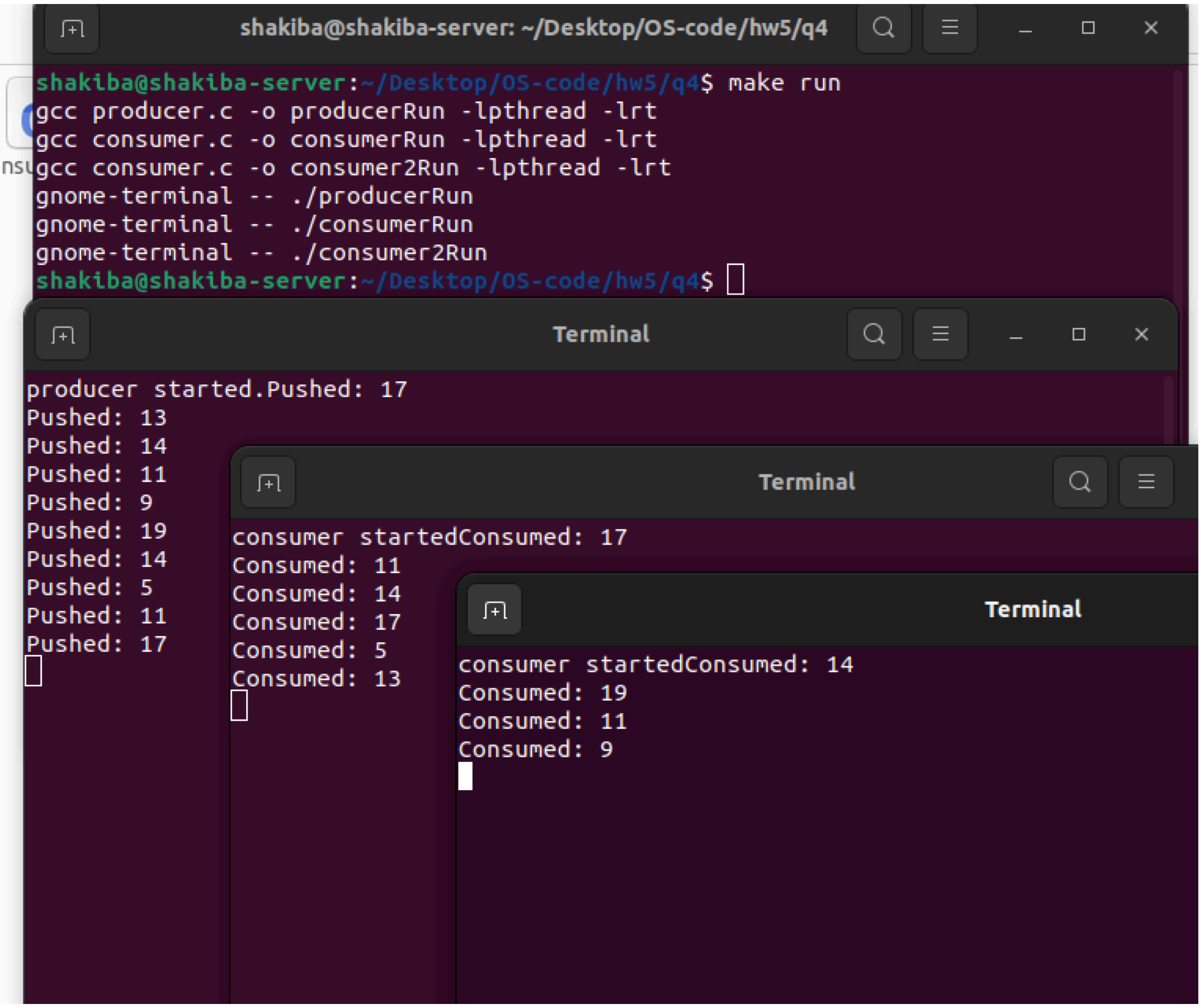
OS HW5 Q4 report
Shakiba Anaraki
99442047

- partA

The codes and Makefile are stored in partA folder.
By command **“make run”** a producer and two consumers will start running in separated terminals.
You can also delete all generated exe files by **“make clean”** command.

A code running sample with one producer and two consumers:



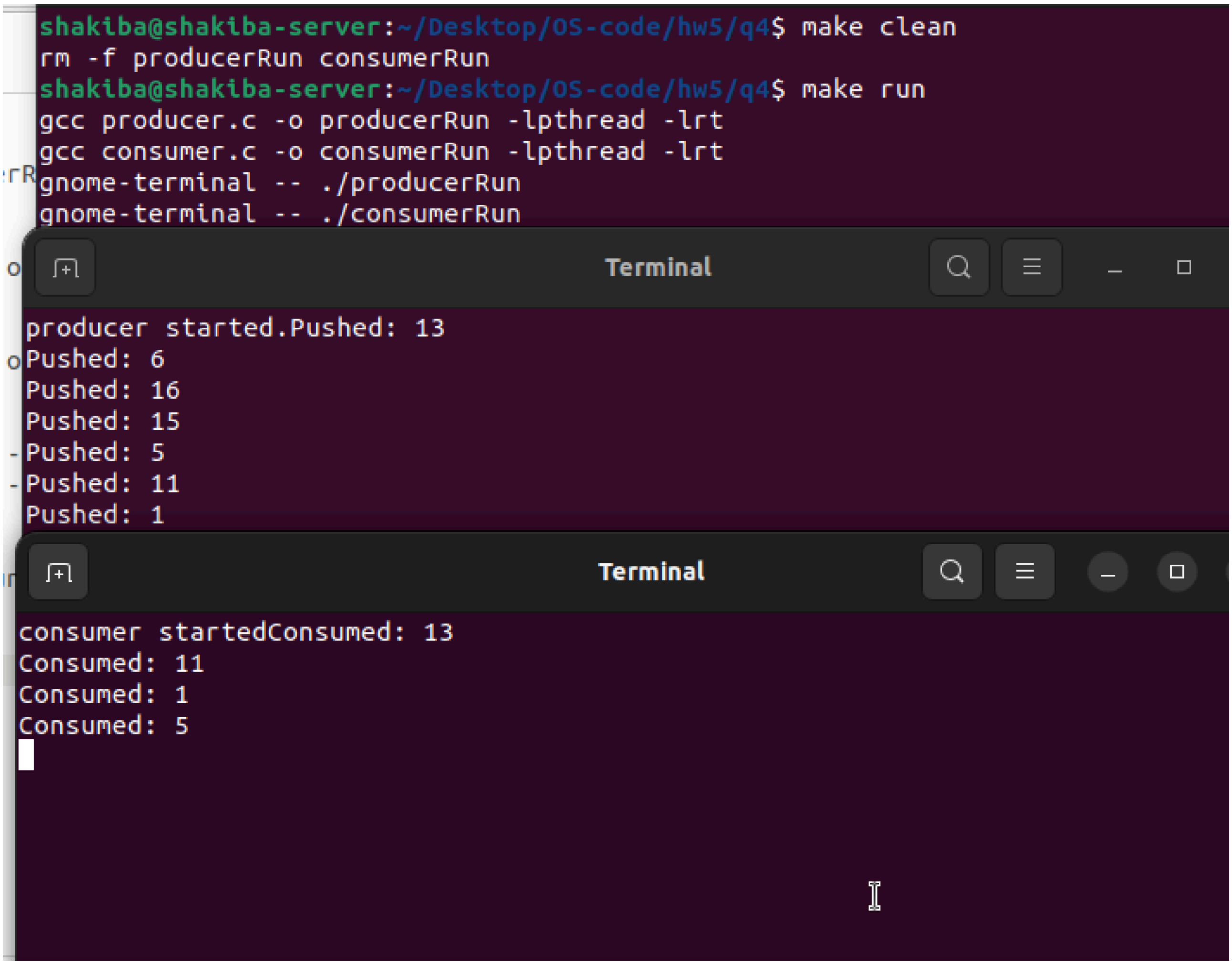
```
shakiba@shakiba-server:~/Desktop/OS-code/hw5/q4$ make run
gcc producer.c -o producerRun -lpthread -lrt
gcc consumer.c -o consumerRun -lpthread -lrt
gcc consumer.c -o consumer2Run -lpthread -lrt
gnome-terminal -- ./producerRun
gnome-terminal -- ./consumerRun
gnome-terminal -- ./consumer2Run
shakiba@shakiba-server:~/Desktop/OS-code/hw5/q4$
```

```
producer started.Pushed: 17
Pushed: 13
Pushed: 14
Pushed: 11
Pushed: 9
Pushed: 19
Pushed: 14
Pushed: 5
Pushed: 11
Pushed: 17
```

```
consumer startedConsumed: 17
Consumed: 11
Consumed: 14
Consumed: 17
Consumed: 5
Consumed: 13
```

```
consumer startedConsumed: 14
Consumed: 19
Consumed: 11
Consumed: 9
```

Another sample with one producer and one consumer:



```
shakiba@shakiba-server:~/Desktop/OS-code/hw5/q4$ make clean
rm -f producerRun consumerRun
shakiba@shakiba-server:~/Desktop/OS-code/hw5/q4$ make run
gcc producer.c -o producerRun -lpthread -lrt
gcc consumer.c -o consumerRun -lpthread -lrt
gnome-terminal -- ./producerRun
gnome-terminal -- ./consumerRun
```

```
producer started.Pushed: 13
Pushed: 6
Pushed: 16
Pushed: 15
Pushed: 5
Pushed: 11
Pushed: 1
```

```
consumer startedConsumed: 13
Consumed: 11
Consumed: 1
Consumed: 5
```

Hard coded limits in this part are :
Maximum shared memory stack size: 5
Producer generates a random number for at most 10 times.
Consumer reads numbers from the shared memory up to a maximum of 20 times.

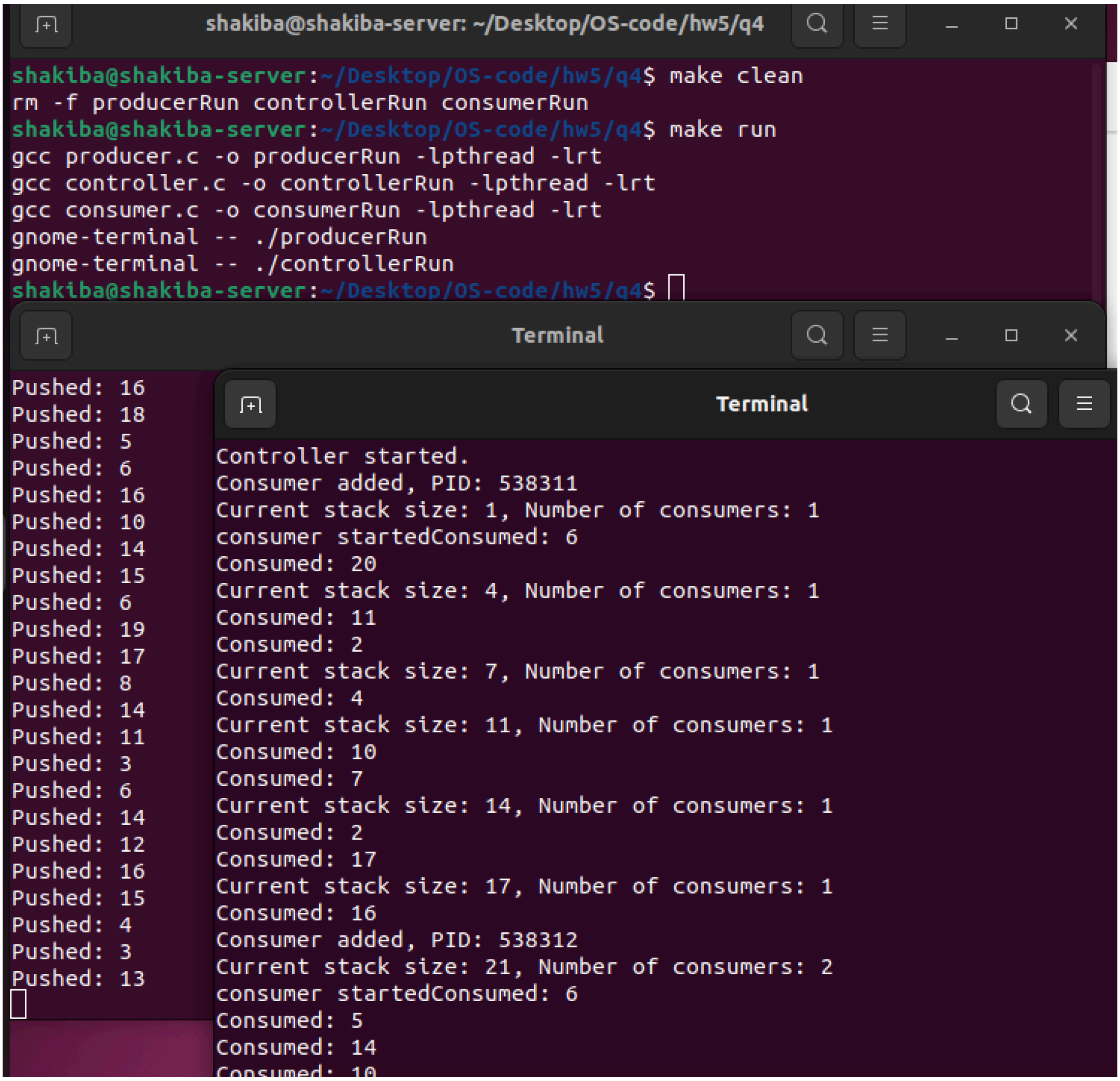
Shared memory data structure defined in the shared_mem.h file:

```
typedef struct {
    int data[MAX_STACK_SIZE];
    int top;
} SharedStack;
```

- **partB**

The codes and Makefile are stored in partB folder.
By command **“make run”** a controller and a producer will start running in separated terminals and also a consumer exe file will be generated in order to be used by the controller for creating consumers.
You can also delete all generated exe files by **“make clean”** command.

A running sample:



```
shakiba@shakiba-server:~/Desktop/OS-code/hw5/q4$ make clean
rm -f producerRun controllerRun consumerRun
shakiba@shakiba-server:~/Desktop/OS-code/hw5/q4$ make run
gcc producer.c -o producerRun -lpthread -lrt
gcc controller.c -o controllerRun -lpthread -lrt
gcc consumer.c -o consumerRun -lpthread -lrt
gnome-terminal -- ./producerRun
gnome-terminal -- ./controllerRun
shakiba@shakiba-server:~/Desktop/OS-code/hw5/q4$
```

```
Pushed: 16
Pushed: 18
Pushed: 5
Pushed: 6
Pushed: 16
Pushed: 10
Pushed: 14
Pushed: 15
Pushed: 6
Pushed: 19
Pushed: 17
Pushed: 8
Pushed: 14
Pushed: 11
Pushed: 3
Pushed: 6
Pushed: 14
Pushed: 12
Pushed: 16
Pushed: 15
Pushed: 4
Pushed: 3
Pushed: 13
```

```
Controller started.
Consumer added, PID: 538311
Current stack size: 1, Number of consumers: 1
consumer startedConsumed: 6
Consumed: 20
Current stack size: 4, Number of consumers: 1
Consumed: 11
Consumed: 2
Current stack size: 7, Number of consumers: 1
Consumed: 4
Current stack size: 11, Number of consumers: 1
Consumed: 10
Consumed: 7
Current stack size: 14, Number of consumers: 1
Consumed: 2
Consumed: 17
Current stack size: 17, Number of consumers: 1
Consumed: 16
Consumer added, PID: 538312
Current stack size: 21, Number of consumers: 2
consumer startedConsumed: 6
Consumed: 5
Consumed: 14
Consumed: 10
```

Hard coded limits in this part are :
Maximum shared memory stack size: 100
Maximum number of consumers generated by controller simultaneously: 10
Minimum desired stack items count : 20
Maximum desired stack items count : 40
Controller sleep interval: 5 seconds
Producer sleep interval: 1 seconds
Consumer sleep interval: 3 seconds
Bothe producer and consumers process work in unlimited loops before being killed by the controller.