Practice 1 – Shared buffer problem

```
sky@sky-VirtualBox:~/Documents/os/labOS/lab3/labSync-student/labSync-student/p1
mutex$ ./run.sh
rm -rf *.o shrdmem
gcc -c shrdmem.c
gcc shrdmem.o -o shrdmem -lpthread
Thread 1: holding 1000000000
Thread 2: holding 2000000000
```

Practice 2 – Bounded buffer problem

```
sky@sky-VirtualBox:~/Documents/os/labOS/lab3/labSync-student/labSync-student/p2
pc$ ./pc
Producer 0 put data 0
Consumer 0 get data 0
Producer 0 put data 1
Consumer 0 get data 1
Producer 0 put data 2
Consumer 0 get data 2
Producer 0 put data 3
Consumer 0 get data 3
Producer 0 put data 4
Consumer 0 get data 4
Producer 0 put data 5
Consumer 0 get data 5
Producer 0 put data 6
Consumer 0 get data 6
Producer 0 put data 7
Consumer 0 get data 7
Producer 0 put data 8
Consumer 0 get data 8
Producer 0 put data 9
Consumer 0 get data 9
```

Problem 3 – Dining Philosopher problem

```
sky@sky-VirtualBox:~/Documents/os/labOS/lab3/labSync-student/labSync-student/p3
dinPhil$ ./run.sh
rm -rf *.o dinPhil
gcc -c dinPhil.c
gcc dinPhil.o -o dinPhil -lpthread
Philosopher 2 has entered room
Philosopher 2 takes fork 2 and 3
Philosopher 2 is eating
Philosopher 0 has entered room
Philosopher 0 takes fork 0 and 1
Philosopher 0 is eating
Philosopher 1 has entered room
Philosopher 3 has entered room
Philosopher 4 has entered room
Philosopher 2 puts fork 3 and 2 down
Philosopher 2 is thinking
Philosopher 3 takes fork 3 and 4
Philosopher 3 is eating
Philosopher 0 puts fork 1 and 0 down
Philosopher 0 is thinking
Philosopher 1 takes fork 1 and 2
Philosopher 1 is eating
Philosopher 3 puts fork 4 and 3 down
Philosopher 3 is thinking
Philosopher 1 puts fork 2 and 1 down
Philosopher 1 is thinking
Philosopher 4 takes fork 4 and 0
Philosopher 4 is eating
Philosopher 2 takes fork 2 and 3
```


Problem 1 – Sequence lock

```
sky@sky-VirtualBox:~/Documents/os/labOS/lab3/labSync-student/labSync-student/ex
1seqlock$ ./run.sh
rm -rf *.o seqlock
gcc -c seqlock.c
gcc seqlock.o -o seqlock -lpthread
val = 1
```

Problem 2 – Aggregated Sum

```
gcc  -g -pthread -I./ -L.  -c utils.c -o utils.o
ar rcs libutils.a utils.o
gcc  -std=c++11 -g -pthread -I./ -L. -lutils main.o libutils.a -o main
rm utils.o
 number : 20    valid  (and represents all characters read)
 number : 4    valid  (and represents all characters read)
 number : 4    valid  (and represents all characters read)
aggsum runs with <arrsz>=20      <tnum>=4        <seednum>=4

[0,4]   [5,9]   [10,14] [15,19]

[ 78,    30,      7,     32,     71,     10,     85,     11,     85,     26,
11,      96,     70,     80,     21,     52,     85,     65,     62,     20,  ]
sequence sum results 997
aggsum gives sum result 997
```


Problem 3 – Interruptible system logger

```
sky@sky-VirtualBox:~/Documents/os/labOS/lab3/labSync-student/labSync-student/ex
3logbuf$ ./run.sh
rm -rf *.o logbuf
gcc -c logbuf.c
gcc logbuf.o -o logbuf -lpthread
Slot  0: 1
Slot  1: 2
Slot  2: 0
Slot  3: 5
Slot  4: 4
Slot  5: 6
Slot  0: 12
Slot  1: 3
Slot  2: 15
Slot  3: 9
Slot  4: 8
Slot  5: 7
Slot  0: 13
Slot  1: 14
Slot  2: 18
Slot  3: 17
Slot  4: 11
Slot  5: 19
Slot  0: 16
Slot  1: 22
Slot  2: 20
Slot  3: 24
```