

National Textile University

Department of Computer Science

Subject:

Operating System

Submitted to:

Sir Nasir

Submitted by:

Khizar Hameed

Reg number:

23-NTU-CS-1169

Lab no:

6

Semester:

5th

LAB 6

| **Peterson’s Algorithm** | **Mutex Lock** |
| --- | --- |
| This code use logic and shared varibles like flag and turn to make sure only one thread run at a time. | This code use a builtin lock called pthread\_mutex\_t which control thread access automaticly. |
| The other thread keep checking in loop untill it get it’s turn, so CPU always busy (busy waiting). | When one thread take the lock, others just wait quitly without using cpu (blocking wait). |
| It only work properly for 2 threads and not good when we use more threads. | It can handle many threads easy and give good performence. |
| CPU keep working even when waiting so it waste time and resources. | CPU free when waiting because system handle the waiting part, so it is more efficent. |
| It is little hard to understand and make without mistake because all condtions are done manualy. | It is very easy to use, we just lock before enter and unlock when finish. |
| Program run slower because of loop checking again and again. | Run faster and smoother because OS control everything in background. |
| This code is mostly use for study purpose to understand how mutual exclution works. | This code use in real life programs to make sure threads work safe together. |
| It work fine for small example but not good for big systems. | It is use in big softwares and multi-threaded apps without problem. |
| Not very reliable and can fail when more threads used. | Very reliable and safe method that developer use in real life. |
| Help to learn concept of mutual exclution in simple way. | Show the real practical way to do mutual exclution in computer system. |