Lab 6
Interprocess Communication

Task: *Review Race conditions, critical regions,and mutual exclusion with busy waiting from your Textbook.*

6.1 Synchronization

Ex: 6.1: This program demonstrate the solution (strict alternation) for critical region problem.

Strictalt.c

```
#include<stdlib.h>
#include<unistd.h>
#include<pthread.h>
#include<stdio.h>

void *thread1f (void * arg);
void *thread2f (void * arg);

int turn = 1;

int main()
{
pthread_t thid1;
pthread_t thid2;

pthread_create (&thid1, NULL, &thread1f, NULL);
pthread_create (&thid2, NULL, &thread2f, NULL);

pthread_join(thid1, NULL);
pthread_join(thid2, NULL);

return 0;
}

void *thread1f(void *arg)
{
int a = 0;
while(a++<20){
        while(turn!= 1);
        fputc('b',stderr);
        turn = 0;
        }
```

```
}
void *thread2f (void * arg)
{
int b = 0;
while(b++<20){
                while(turn != 0);
                fputc('a', stderr);
                turn = 1;
                }
}
```

Assignment #L6

1. Run the program Ex. 6.1, and analyze the output.
2. Modify the above program to demonstrate lock variables solution, and comment the deficiencies of this solution.
3. Write the problems in the solution of Ex. 6.1 and develop new version of the program using Peterson's solution.