ST. XAVIER’S COLLEGE

(Affiliated to Tribhuvan University)
Maitighar, Kathmandu

OS Lab Assignment #5

Threads

SUBMITTED BY:

Milan Rawal

018BSCIT019

2nd Year/4th Sem

SUBMITTED TO:

Er. Rajan Karmacharya (Coordinator)	
Er Rabin Maharjan (Lecturer)	

Department of Computer Science

Threads

As with processes, threads appears to run concurrently; the Linux kernel schedules them asynchronously, interrupting each thread time to time to give others a chance to execute. Threads exists within a process. GNU/Linux implements the POSIX standard thread API (pthreads). All thread functions and data types are declared in the header file <*pthread.h*>. The pthread functions are not included in the standard C library; they are in libpthread, therefore -lpthread should add when linking program.

5.1 Thread Creation

Each thread have their own thread ID as process, thread ID referred by type *pthread_t*. The pthread_create function create new threads. It has following formate.

*int pthread_create (pthread_t *thread, pthread_attr_t *attr, void *(*start_routine) (void*), void *arg);* The pthread_exit function terminates the thread.

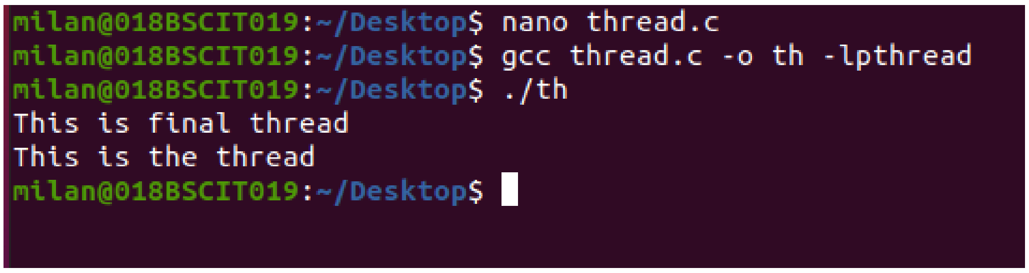
*thread_exit(void *return_val);*

The pthread_join function waits other process for termination – equivalent of wait. *int pthread_join(pthread_t th, void **thread_return);*

Ex 5.1: Thread Creation (threadc.c)

```
#include<stdio.h>
#include<pthread.h>
#include<unistd.h>
void *fun(void *para) {printf("This is the thread\n");}
int main(void){
        pthread_t id; pthread_create(&id,NULL,&fun,NULL);
        printf("This is final thread\n"); pthread_join(id,NULL);
}
```

Output

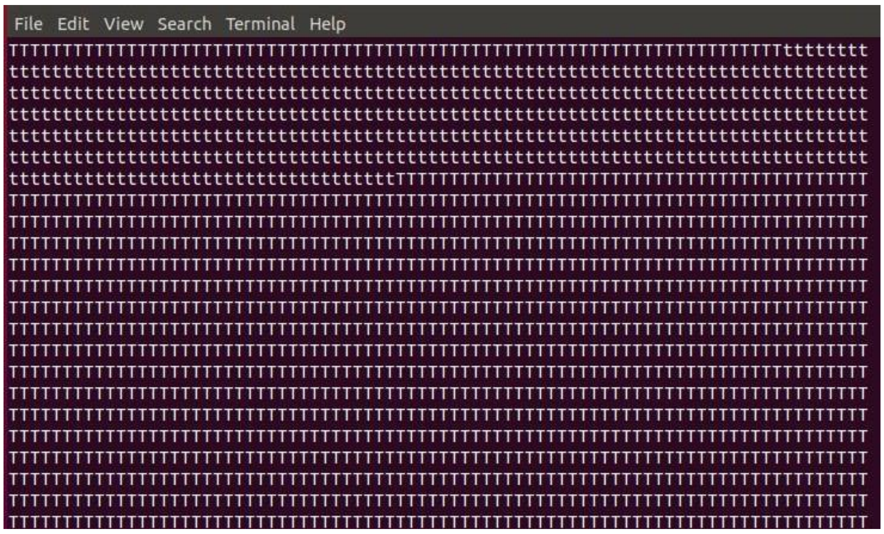


```
milan@018BSCIT019:~/Desktop$ nano thread.c
milan@018BSCIT019:~/Desktop$ gcc thread.c -o th -lpthread
milan@018BSCIT019:~/Desktop$ ./th
This is final thread
This is the thread
milan@018BSCIT019:~/Desktop$ 
```

Ex 5.2: Thread Creation (threadc.c)

```
#include <unistd.h>
#include <pthread.h>
struct param{ char ch; int count;};
void *printc(void *parameter){
        struct param *p = (struct param*) parameter;
        for(int i=0;i<p->count;++i) fputc(p?ch,stderr);
}
int main(void){
	pthread_t thread1_id, thread2_id;
	struct param thread1_args, thread2_args;
	thread1_args.ch = 'T';
	thread2_args.count = 3000;
	pthread_create(&thread1_id,NULL,&printc,&thread1_args);
	thread2_args.ch ='t'; thread2_args.count = 2000;
	pthread_create(&thread2_id,NULL,&printc,&thread2_args);
	pthread_join(thread1_id,NULL); pthread_join(thread2_id,NULL);
}
```
Warning! : *Run this program as : gcc -o threadc threadc.c –lpthread*

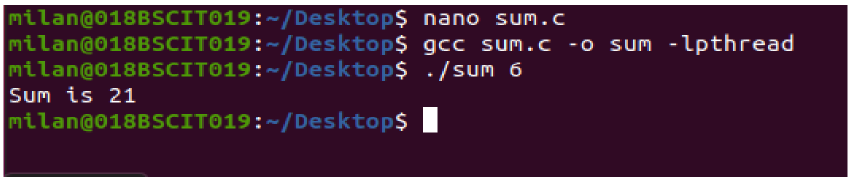
Output



```
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTtttttttt
tttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttt
tttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttt
tttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttt
tttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttt
tttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttttt
ttttttttttttttttttttttttttttttttttttTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
```

STATEMENT: WRITE A PROGRAM USING THREADS THAT PRINTS SUM OF NUMBERS UP TO GIVEN POSITIVE NUMBER.

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
int sum = 0;
void* sum_runner(void* arg){
	int *limit_ptr = (int*)arg; int limit =  *limit_ptr; int i;
	for(i = 0; i <= limit; i++) sum += i;
}
int main(int argc, char **argv) {
	if (argc < 2){   printf("Usage: %s <num>\n", argv[0]); exit(-1); }
	int limit = atoi(argv[1]); pthread_t tid; pthread_attr_t attr;
	pthread_attr_init(&attr); pthread_create(&tid, &attr, sum_runner, &limit);
	pthread_join(tid, NULL); printf("Sum is %d\n", sum);
}
```
OUTPUT:

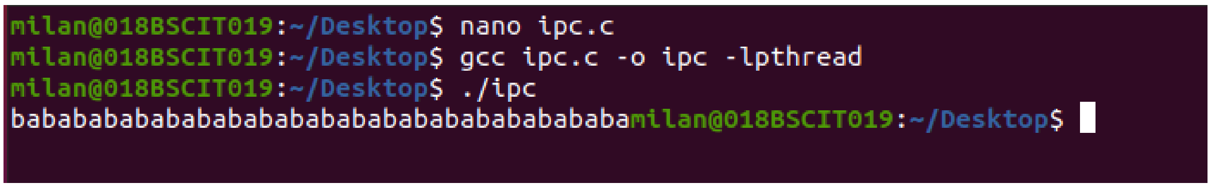


```
milan@018BSCIT019:~/Desktop$ nano sum.c
milan@018BSCIT019:~/Desktop$ gcc sum.c -o sum -lpthread
milan@018BSCIT019:~/Desktop$ ./sum 6
Sum is 21
milan@018BSCIT019:~/Desktop$ 
```

STATEMENT: A PROGRAM DEMONSTRATE THE SOLUTION (STRICT ALTERNATION) FOR CRITICAL REGION PROBLEM.

```c
#include<stdlib.h>
#include<unistd.h>
#include<pthread.h>
#include<stdio.h>
int turn = 1;
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;
        }
}
int main(void){
        pthread_t thid1, thid2;
        pthread_create (&thid1, NULL, &thread1f, NULL);
        pthread_create (&thid2, NULL, &thread2f, NULL);
        pthread_join(thid1, NULL); pthread_join(thid2, NULL);
}
```

OUTPUT:



```
milan@018BSCIT019:~/Desktop$ nano ipc.c
milan@018BSCIT019:~/Desktop$ gcc ipc.c -o ipc -lpthread
milan@018BSCIT019:~/Desktop$ ./ipc
babababababababababababababababababababamilan@018BSCIT019:~/Desktop$ 
```