BITS Pilani - Hyderabad Campus
Advanced Operating Systems (G 623)

Lab Sheet 4

1st Sem 2025-26

Program1: Creating multiple threads and making them add numbers to a global variable, print the final value of the global variable.

```
/* thread1.c */

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

#define N 5

int sum = 0;

void* fun(void *val) { int
       *n = (int *)val;
    sum += *n;
    return NULL;
}

int main(int argc, char* argv[]) {
       pthread_t t[N];
       int data[N], errcode, i;
     for(i = 0; i < N; i ++) {
       data [i] = atoi (argv[i + 1]);
         if (pthread_create (&t[i], NULL, fun, (void*)&data[i])) {
                printf ("Error creating thread\n");

           exit(1);
            }
       }
   for (i = 0; i < N; i ++) {
      pthread_join (t[i], NULL);
     }
   printf("Sum: %d\n", sum);
   exit(0);
}
```

Compiling and running it:

```
$ cc –pthread –o thread1 thread1.c
$ ./thread1 1 2 3 4  5
Sum: 15
```

Program2: Creating multiple threads and manipulating a shared memory, using **mutex**.

```
/* thread2.c*/

#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#define N 5
int sum = 0;
int inc = 5;

pthread_mutex_t lock;

void *fun(void *val)
{
  pthread_mutex_lock(&lock);
  sum += inc;
  printf ("Value: %d\n", sum);
  inc += 5;
  pthread_mutex_unlock(&lock);
  pthread_exit (NULL);
}

int main (int argc, char* argv[])
{
 pthread_t t[N];
 int errcode, i;
 for (i = 0; i < N;  i ++)
 {
  if (pthread_create(&t[i], NULL, fun,   NULL))
  {
   printf ("Error creating thread\n");
   return EXIT_FAILURE;
  }
 }
 for (i = 0; i < N;  i ++)
 {
   pthread_join (t[i], NULL);
 }
return EXIT_SUCCESS;
}
```

Compiling and running it:

```
$ gcc -pthread -o thread2 thread2.c
$ ./thread2
Value: 5
Value: 15
Value: 30
Value: 50
Value: 75
```

Program3. Making two threads communicate using a Pipe. One thread writes into it, and the other reads from it.

```
/* thread3.c */
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <pthread.h>

int pipedata[2];

void *reader_function() {
        while(1) {
           char ch;
           if (read(pipedata[0],&ch,1)!=1) {
                printf ("Read error\n");
                exit(1);
           }
           printf("Thread ID:%lu, Data read: %c\n", pthread_self(),ch);
           if(ch=='q') {
                break;
           }
      }
      pthread_exit(NULL);
}

  void *writer_function() {
      int i = 0;
      char ch = 'a', dummy;
      while (ch != 'q') {scanf
           ("%c", &ch); scanf
           ("%c",&dummy);
           if (write(pipedata[1], &ch,1)!=1)
           {
            printf("Write error\n");
            exit(1);
           }
           printf("Thread ID:%lu,Data written:%c\n",pthread_self(),ch);
      }
      pthread_exit(NULL);
}

int main () {
      pthread_t reader, writer;
      if (pipe(pipedata) < 0)
      {
        printf("Pipe creation error\n");
        exit (1);
      }
      pthread_create(&writer,NULL,writer_function,NULL);
      pthread_create(&reader,NULL,reader_function,NULL);
      pthread_join(writer,NULL);
      pthread_join(reader,NULL);
   exit(0);
}
```

Compiling and running it:

```
$ gcc -pthread -o thread3 thread3.c
$ ./thread3 a
Thread ID: 123145302839296, Data written: a
Thread ID: 123145303375872, Data read: a c
Thread ID: 123145302839296, Data written: c
Thread ID: 123145303375872, Data read: c
```

Task: Create a multi-threaded program to implement the below distributed system with 3 nodes and their communication channels as shown in the below diagram. Assume that the communication channels are FIFO channels. Show the message communication amongst these channels using Pipes as communication medium.