**THREADING & SYNCHRONIZATION APPLICATIONS**

**Ex.No:10**

# AIM:

To write a c program to implement Threading and Synchronization Applications.

# ALGORITHM:

Step 1: Start the process

Step 2: Declare process thread, thread-id.

Step 3: Read the process thread and thread state.

Step 4: Check the process thread equals to thread-id by using if condition. Step 5: Check the error state of the thread.

Step 6: Display the completed thread process. Step 7: Stop the process

# PROGRAM:

#include<stdio.h> #include<string.h> #include<pthread.h> #include<stdlib.h> #include<unistd.h> pthread\_t tid[2];

void\* doSomeThing(void \*arg)

{

unsigned long i = 0; pthread\_t id = pthread\_self();

if(pthread\_equal(id,tid[0]))

{

printf("\n First thread processing\n");

}

else

{

printf("\n Second thread processing\n");

}

for(i=0; i<(0xFFFFFFFF);i++);

return NULL;

}

int main(void)

{

int i = 0;

int err;

while(i < 2)

{

err = pthread\_create(&(tid[i]), NULL, &doSomeThing, NULL); if (err != 0)

printf("\ncan't create thread :[%s]", strerror(err)); else

printf("\n Thread created successfully\n");

i++;

}

sleep(5); return 0;

}