Experiment 11

Aim: Write a program to create a thread in Linux OS.

- A **thread** is the smallest unit of execution within a process. Threads allow a program to perform multiple tasks concurrently, sharing the same memory space.
- In Linux, threads can be created using the **POSIX Threads (pthreads)** library.
- Each thread runs independently and can perform different parts of a program simultaneously.
- Multithreading is especially useful for improving the performance of applications that require parallel processing.

Sample Code:

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

void* thread_function(void* arg) {
   int i;
   printf("Inside thread\n");
   for (i = 0; i < 5; i++) {
      printf("i: %d\n", i);
      sleep(1);
   }
   return NULL;
}

int main() {
   pthread_t a_thread; // Thread declaration
   int j;

   pthread_create(&a_thread, NULL, thread_function, NULL);
   pthread_join(a_thread, NULL); // Process waits for thread to finish

   printf("Inside main program\n");
   for (j = 20; j < 25; j++) {
      printf("j: %d\n", j);
      sleep(1);
   }

   return 0;
}
```

Sample Screenshots:

Creating a vi file named threads.c

```
localhost:~/Ritika# vi threads.c
```

Writing the C program in threads.c

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

void *thread_function(void *arg);  // Function prototype
int i, j;

void *thread_function(void *arg)
{
    printf("Inside thread\n");
    for (i = 0; i < 5; i++)
    {
        printf("i: %d\n", i);
        sleep(1);
    }
    return NULL;
}

int main()
{
    pthread_t a_thread;  // Thread declaration

    pthread_create(&a_thread, NULL, thread_function, NULL);
    pthread_join(a_thread, NULL);  // Process waits for thread to finish

    printf("Inside main program\n");
    for (j = 20; j < 25; j++)
    {
        printf("j: %d\n", j);
        sleep(1);
    }

    return 0;
}
```

Compilation and Output execution:

```
localhost:~/Ritika# gcc threads.c -o threads -pthread
localhost:~/Ritika# ./threads
Inside thread
i: 0
i: 1
i: 2
i: 3
i: 4
Inside main program
j: 20
j: 21
j: 22
j: 23
j: 24
localhost:~/Ritika#
```