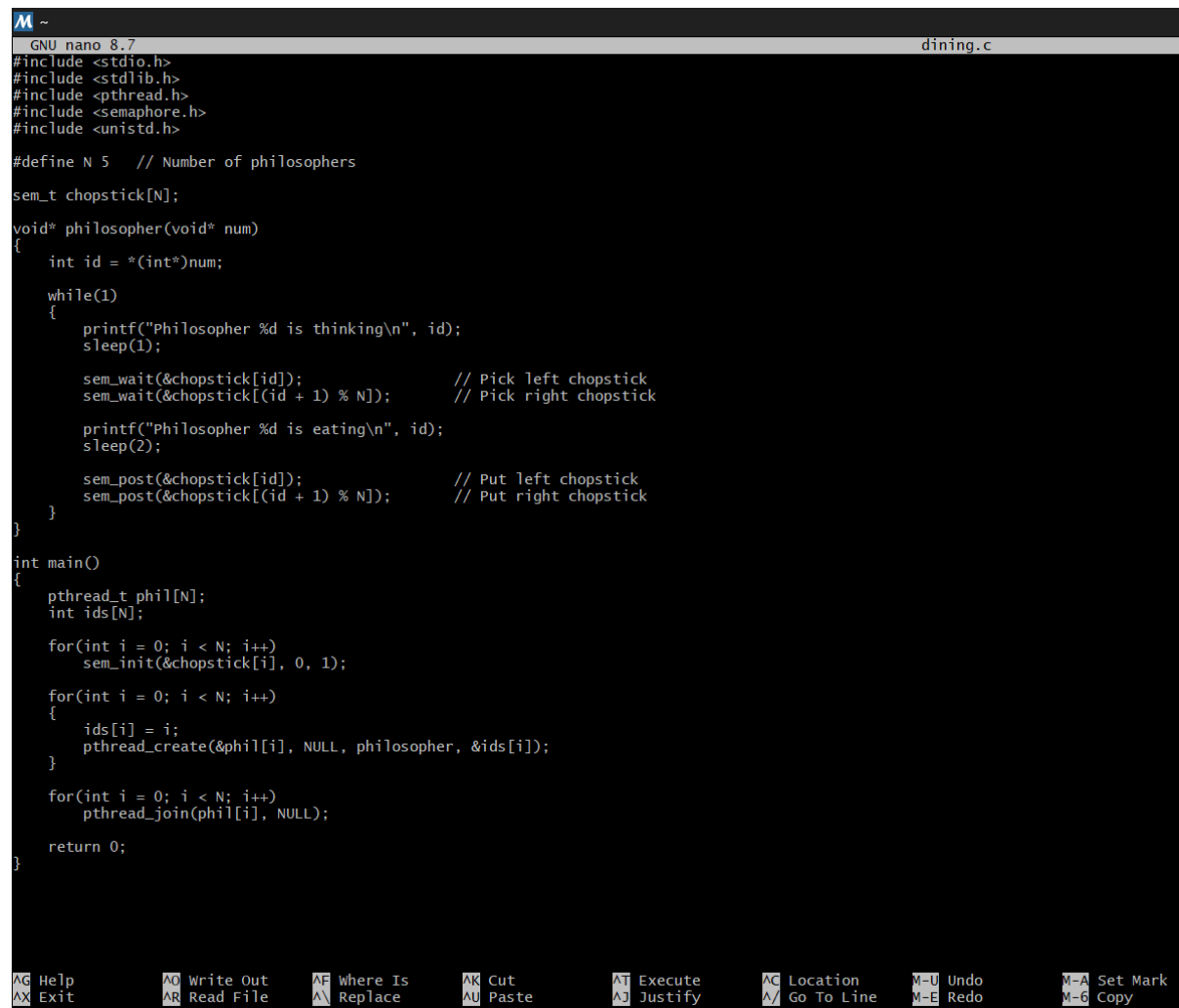


## Practical:- 05

### Aim:

Considered there are N philosophers seated around a circular table with one chopstick between each pair of philosophers. There is one chopstick between each philosopher. A philosopher may eat if he can pick up the two chopsticks adjacent to him. One chopstick may be picked up by any one of its adjacent followers but not both. Write a program to solve the problem using process synchronization technique.

### Code:



```
GNU nano 8.7 dining.c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

#define N 5 // Number of philosophers

sem_t chopstick[N];

void* philosopher(void* num)
{
    int id = *(int*)num;

    while(1)
    {
        printf("Philosopher %d is thinking\n", id);
        sleep(1);

        sem_wait(&chopstick[id]);          // Pick left chopstick
        sem_wait(&chopstick[(id + 1) % N]); // Pick right chopstick

        printf("Philosopher %d is eating\n", id);
        sleep(2);

        sem_post(&chopstick[id]);          // Put left chopstick
        sem_post(&chopstick[(id + 1) % N]); // Put right chopstick
    }
}

int main()
{
    pthread_t phil[N];
    int ids[N];

    for(int i = 0; i < N; i++)
        sem_init(&chopstick[i], 0, 1);

    for(int i = 0; i < N; i++)
    {
        ids[i] = i;
        pthread_create(&phil[i], NULL, philosopher, &ids[i]);
    }

    for(int i = 0; i < N; i++)
        pthread_join(phil[i], NULL);

    return 0;
}
```

⌘ Help ⌘ Write Out ⌘ Where Is ⌘ Cut ⌘ Execute ⌘ Location ⌘ Undo ⌘ Set Mark  
⌘ Exit ⌘ Read File ⌘ Replace ⌘ Paste ⌘ Justify ⌘ Go To Line ⌘ Redo ⌘ Copy

## OUTPUT:

```
M ~  
Nandini Kasare@LAPTOP-NIHKGM4O MINGW64 ~  
$ nano dining.c  
  
Nandini Kasare@LAPTOP-NIHKGM4O MINGW64 ~  
$ gcc dining.c -o dining -lpthread  
  
Nandini Kasare@LAPTOP-NIHKGM4O MINGW64 ~  
$ ./dining  
Philosopher 2 is thinking  
Philosopher 0 is thinking  
Philosopher 1 is thinking  
Philosopher 3 is thinking  
Philosopher 4 is thinking  
Philosopher 1 is eating  
Philosopher 0 is eating  
Philosopher 1 is thinking  
Philosopher 4 is eating  
Philosopher 0 is thinking  
Philosopher 4 is thinking
```