Task 5

Task 5: Simulate Thread and Multi Thread using a C program

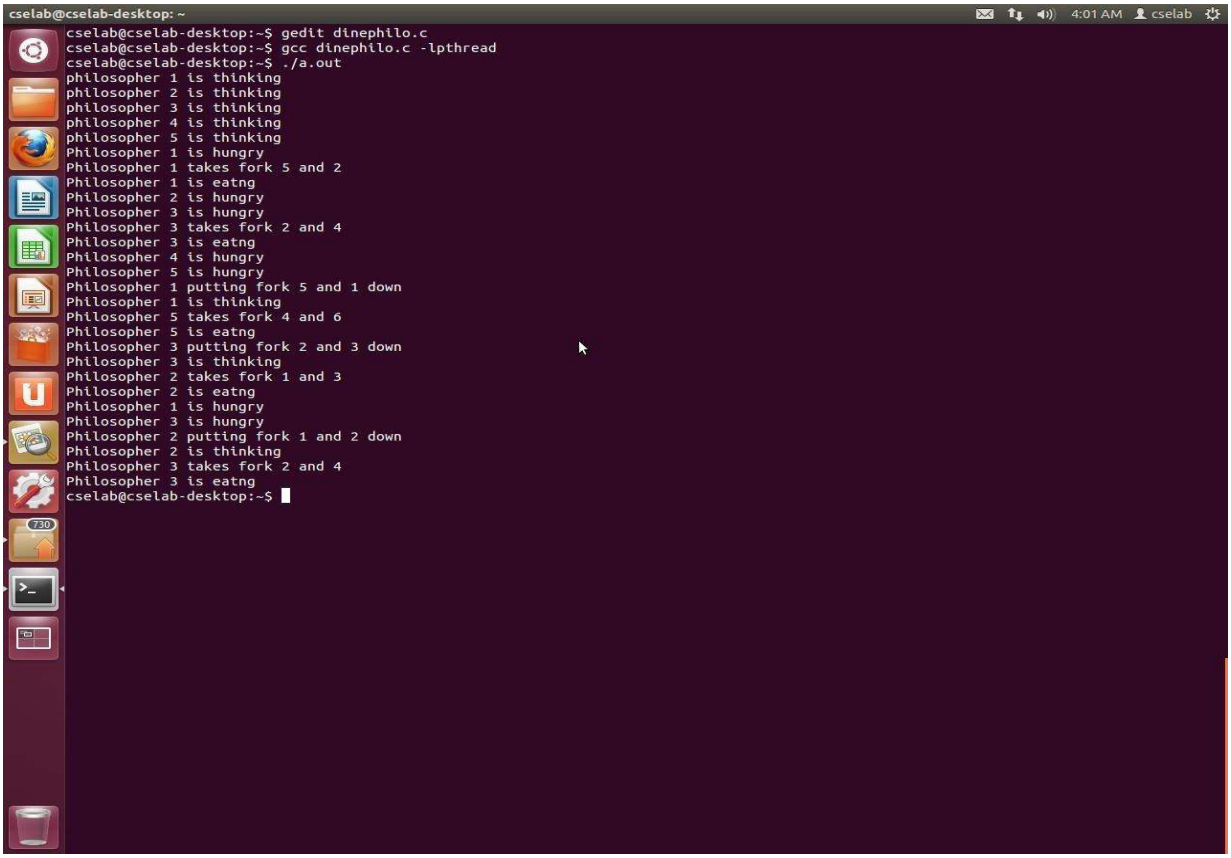
Program:

```
// Program to implement Dining Philosopher problem using semaphores.

#include<stdio.h>
#include<stdlib.h>
#include<semaphore.h>
#define N 5
#define thinking 0
#define hungry 1
#define eating 2
#define left (ph_num+4)%N
#define right (ph_num+1)%N
sem_t mutex;
sem_t s[N];
void *philosopher(void *num);
void take_fork(int);
void put_fork(int);
void test(int);
int state[N]={thinking,thinking,thinking,thinking,thinking};
int phil_num[N]={0,1,2,3,4};
int main()
{
int i;
pthread_t thread_id[N];
sem_init(&mutex,0,1);
for(i=0;i<N;i++)
sem_init(&s[i],0,0);
for(i=0;i<N;i++)
{
pthread_create(&thread_id[i],NULL,philosopher,&phil_num[i]);
printf("philosopher %d is thinking \n",i+1);
}
for(i=0;i<N;i++)
pthread_join(thread_id[i],NULL);
}
void *philosopher(void *num)
{
while(1)
{
int *i=num;
sleep(1);
take_fork(*i);
sleep(1);
put_fork(*i);
}
```

```
}
void take_fork(int ph_num)
{
sem_wait(&mutex);
state[ph_num]=hungry;
printf("Philosopher %d is hungry\n",ph_num+1);
test(ph_num);
sem_post(&mutex);
sem_wait(&s[ph_num]);
sleep(1);
}
void test(int ph_num)
{
static count=0;
if(state[ph_num]==hungry&& state[left]!=eating && state[right]!=eating)
{
state[ph_num]=eating;
printf("Philosopher %d takes fork %d and %d\n",ph_num+1,left+1,ph_num+2);
printf("Philosopher %d is eatng\n",ph_num+1);
sem_post(&s[ph_num]);
count++;
}
if(count==5)
exit(1);
}
void put_fork(int ph_num)
{
sem_wait(&mutex);
state[ph_num]=thinking;
printf("Philosopher %d putting fork %d and %d down \n",ph_num+1,left+1,ph_num+1);
printf("Philosopher %d is thinking\n",ph_num+1);
test(left);
test(right);
sem_post(&mutex);
}
```

Output:


cselab@cselab-desktop: ~
4:01 AM cselab
cselab@cselab-desktop:~$ gedit dinephilo.c
cselab@cselab-desktop:~$ gcc dinephilo.c -lpthread
cselab@cselab-desktop:~$ ./a.out
phlosopher 1 is thinking
philosopher 2 is thinking
philosopher 3 is thinking
philosopher 4 is thinking
philosopher 5 is thinking
Philosopher 1 is hungry
Philosopher 1 takes fork 5 and 2
Philosopher 1 is eatng
Philosopher 2 is hungry
Philosopher 3 is hungry
Philosopher 3 takes fork 2 and 4
Philosopher 3 is eatng
Philosopher 4 is hungry
Philosopher 5 is hungry
Philosopher 1 putting fork 5 and 1 down
Philosopher 1 is thinking
Philosopher 5 takes fork 4 and 6
Philosopher 5 is eatng
Philosopher 3 putting fork 2 and 3 down
Philosopher 3 is thinking
Philosopher 2 takes fork 1 and 3
Philosopher 2 is eatng
Philosopher 1 is hungry
Philosopher 3 is hungry
Philosopher 2 putting fork 1 and 2 down
Philosopher 2 is thinking
Philosopher 3 takes fork 2 and 4
Philosopher 3 is eatng
cselab@cselab-desktop:~$