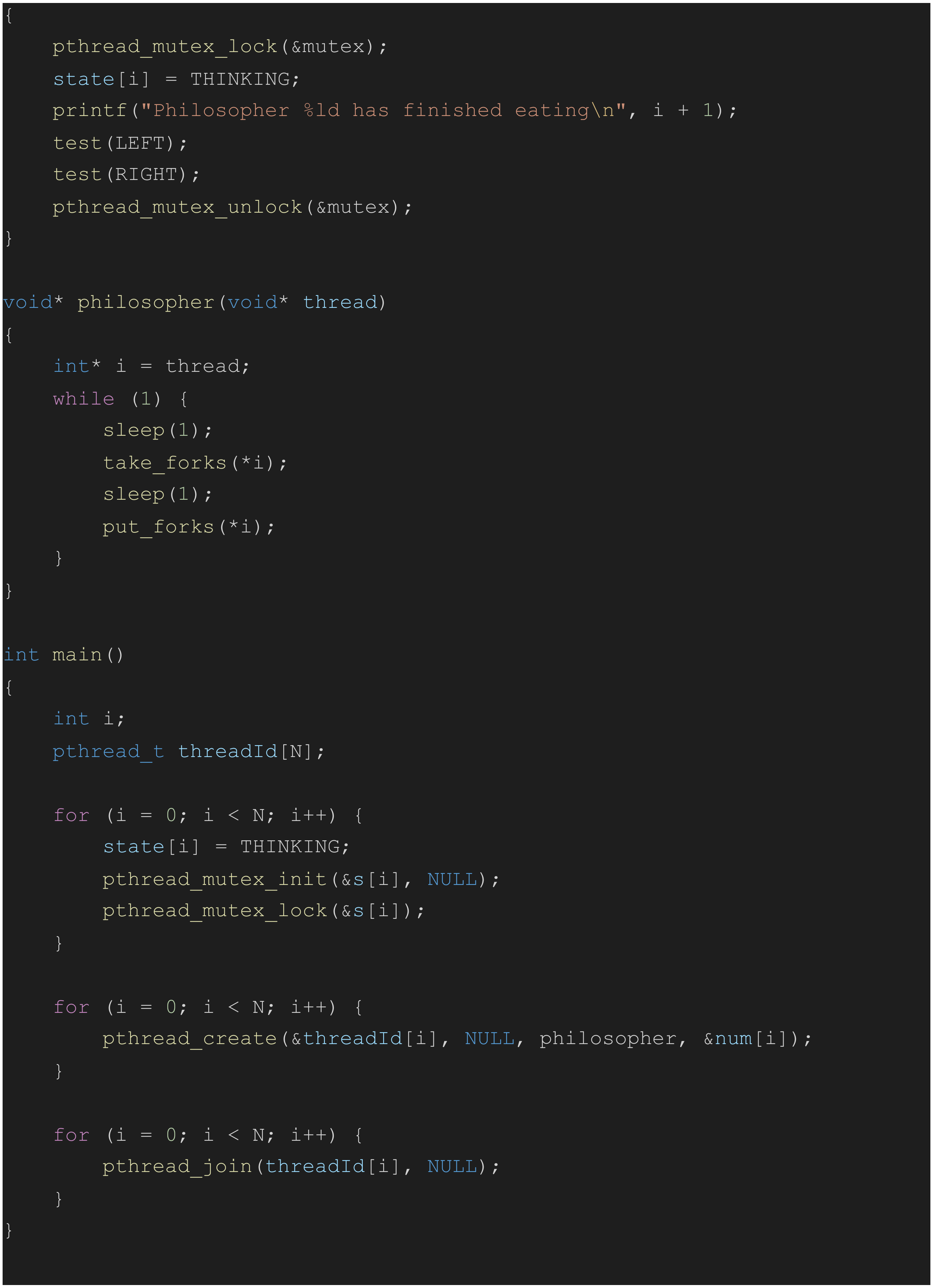
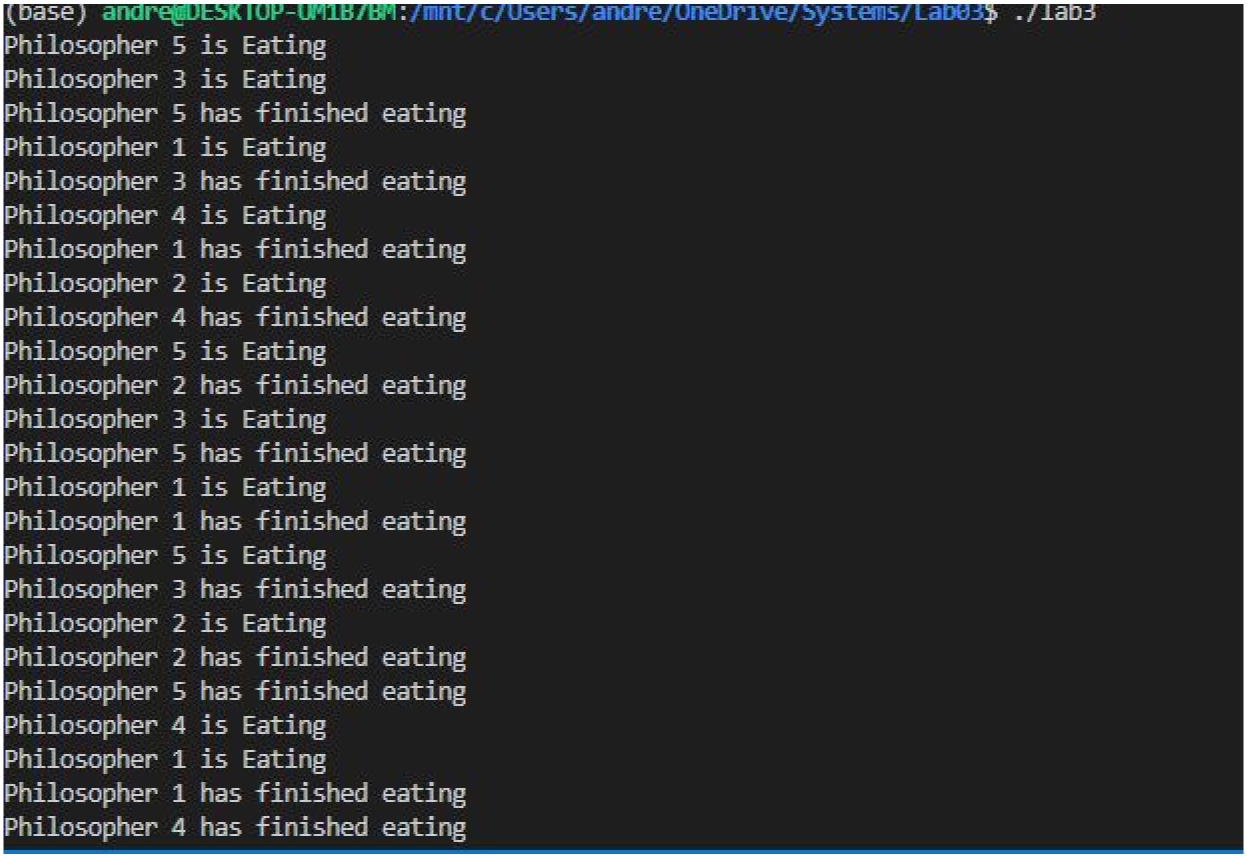
```
#include <pthread.h>
#include <semaphore.h>
#include <stdio.h>
#include <unistd.h>

#define N 5
#define THINKING 2
#define HUNGRY 1
#define EATING 0
#define LEFT (i + 4) % N
#define RIGHT (i + 1) % N

int state[N];
int num[N] = {0, 1, 2, 3, 4};

pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
pthread_mutex_t s[N];

void test(long i)
{
    if (state[i] == HUNGRY && state[LEFT] != EATING && state[RIGHT] !=
EATING) {
        state[i] = EATING;
        printf("Philosopher %ld is Eating\n", i + 1);
        pthread_mutex_unlock(&s[i]);
    }
}


void take_forks(long i)
{
    pthread_mutex_lock(&mutex);
    state[i] = HUNGRY;
    test(i);
    pthread_mutex_unlock(&mutex);
    pthread_mutex_lock(&s[i]);
}


void put_forks(long i)
```



```
{
    pthread_mutex_lock(&mutex);
    state[i] = THINKING;
    printf("Philosopher %ld has finished eating\n", i + 1);
    test(LEFT);
    test(RIGHT);
    pthread_mutex_unlock(&mutex);
}

void* philosopher(void* thread)
{
    int* i = thread;
    while (1) {
        sleep(1);
        take_forks(*i);
        sleep(1);
        put_forks(*i);
    }
}

int main()
{
    int i;
    pthread_t threadId[N];

    for (i = 0; i < N; i++) {
        state[i] = THINKING;
        pthread_mutex_init(&s[i], NULL);
        pthread_mutex_lock(&s[i]);
    }

    for (i = 0; i < N; i++) {
        pthread_create(&threadId[i], NULL, philosopher, &num[i]);
    }

    for (i = 0; i < N; i++) {
        pthread_join(threadId[i], NULL);
    }
}
```



```
(base) andre@DESKTOP-OM1B7BM:/mnt/c/Users/andre/OneDrive/Systems/Lab03$ ./lab3
Philosopher 5 is Eating
Philosopher 3 is Eating
Philosopher 5 has finished eating
Philosopher 1 is Eating
Philosopher 3 has finished eating
Philosopher 4 is Eating
Philosopher 1 has finished eating
Philosopher 2 is Eating
Philosopher 4 has finished eating
Philosopher 5 is Eating
Philosopher 2 has finished eating
Philosopher 3 is Eating
Philosopher 5 has finished eating
Philosopher 1 is Eating
Philosopher 1 has finished eating
Philosopher 5 is Eating
Philosopher 3 has finished eating
Philosopher 2 is Eating
Philosopher 2 has finished eating
Philosopher 5 has finished eating
Philosopher 4 is Eating
Philosopher 1 is Eating
Philosopher 1 has finished eating
Philosopher 4 has finished eating
```