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

pthread_t philosophers[N];
pthread_mutex_t mutex ;
pthread_mutex_t S[N];
int state[N] = {THINKING, THINKING, THINKING, THINKING,

void test(int i){
    if (state[i] == HUNGRY &&
        state[LEFT] != EATING &&
        state[RIGHT] != EATING){
           state[i] = EATING;
           pthread_mutex_unlock(&S[i]);
    }
}

void put_fork(int i){
        pthread_mutex_lock(&mutex);
        state[i] = THINKING;
        test(LEFT);
        test(RIGHT);
        pthread_mutex_unlock(&mutex);
}

void take_fork(int i){
        pthread_mutex_lock(&mutex);
        state[i] = HUNGRY;
        test(i);
        pthread_mutex_unlock(&mutex);
        pthread_mutex_lock(&S[i]);
}

void * philosophers_fn(void * args) {
        int i= *((int*)args);

        for (;;) {
            take_fork(i);
            put_fork(i);
        }
}
```