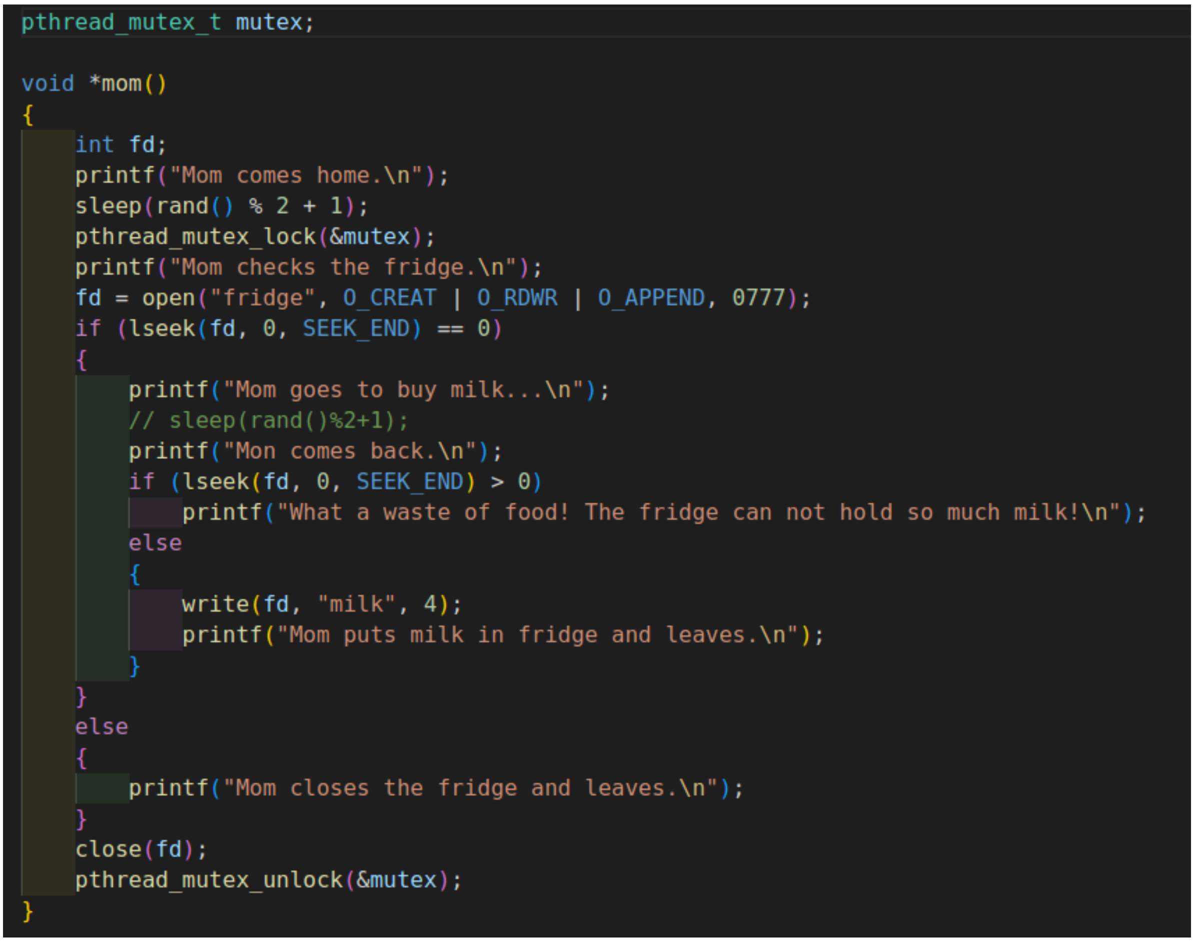
CS302 OS Lab12 - Report

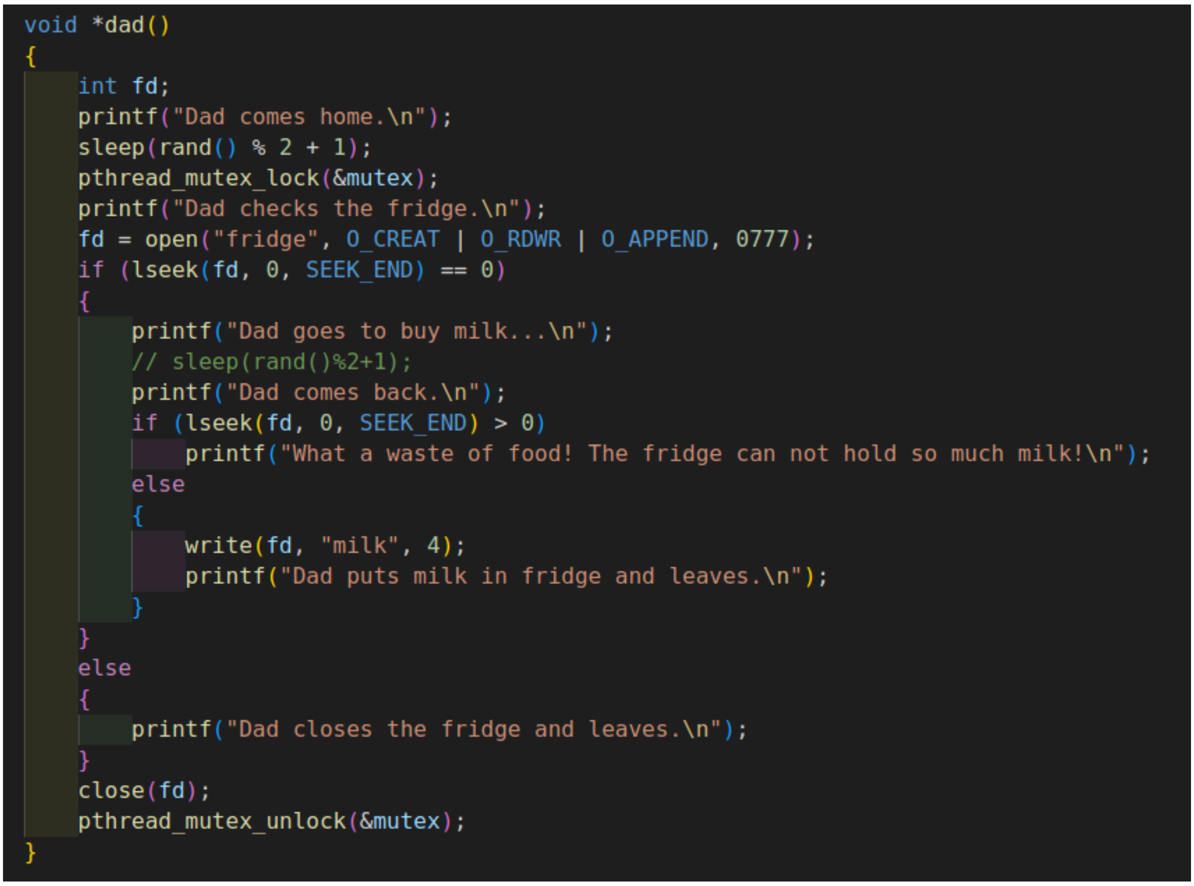
Name: 刘仁杰
SID: 11911808

Answers

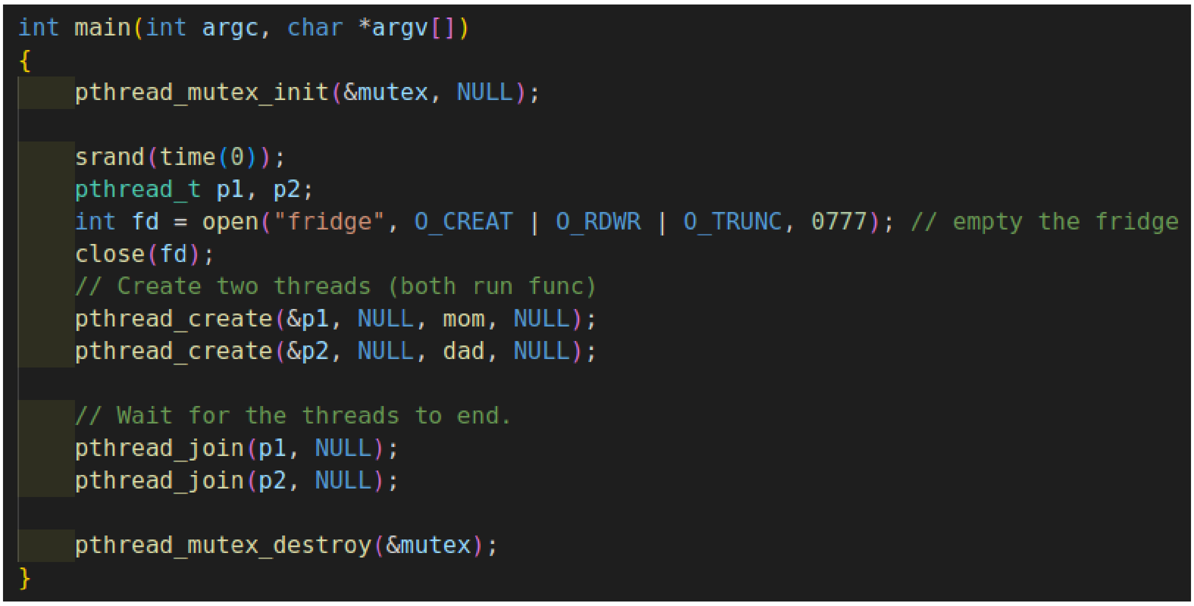
1.使用mutex解决too much milk problem，直接修改milk.c，报告中包含代码截图及运行结果截图



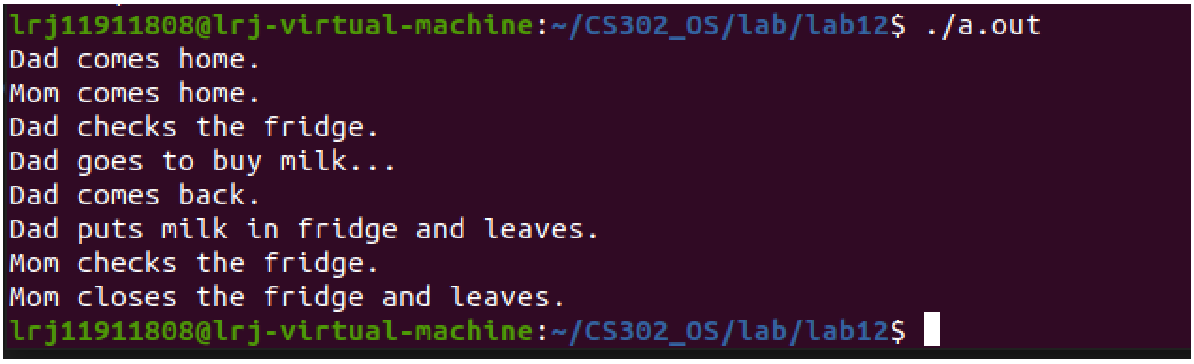
```c
pthread_mutex_t mutex;

void *mom()
{
    int fd;
    printf("Mom comes home.\n");
    sleep(rand() % 2 + 1);
    pthread_mutex_lock(&mutex);
    printf("Mom checks the fridge.\n");
    fd = open("fridge", O_CREAT | O_RDWR | O_APPEND, 0777);
    if (lseek(fd, 0, SEEK_END) == 0)
    {
        printf("Mom goes to buy milk...\n");
        // sleep(rand()%2+1);
        printf("Mon comes back.\n");
        if (lseek(fd, 0, SEEK_END) > 0)
            printf("What a waste of food! The fridge can not hold so much milk!\n");
        else
        {
            write(fd, "milk", 4);
            printf("Mom puts milk in fridge and leaves.\n");
        }
    }
    else
    {
        printf("Mom closes the fridge and leaves.\n");
    }
    close(fd);
    pthread_mutex_unlock(&mutex);
}
```



```c
void *dad()
{
    int fd;
    printf("Dad comes home.\n");
    sleep(rand() % 2 + 1);
    pthread_mutex_lock(&mutex);
    printf("Dad checks the fridge.\n");
    fd = open("fridge", O_CREAT | O_RDWR | O_APPEND, 0777);
    if (lseek(fd, 0, SEEK_END) == 0)
    {
        printf("Dad goes to buy milk...\n");
        // sleep(rand()%2+1);
        printf("Dad comes back.\n");
        if (lseek(fd, 0, SEEK_END) > 0)
            printf("What a waste of food! The fridge can not hold so much milk!\n");
        else
        {
            write(fd, "milk", 4);
            printf("Dad puts milk in fridge and leaves.\n");
        }
    }
    else
    {
        printf("Dad closes the fridge and leaves.\n");
    }
    close(fd);
    pthread_mutex_unlock(&mutex);
}
```

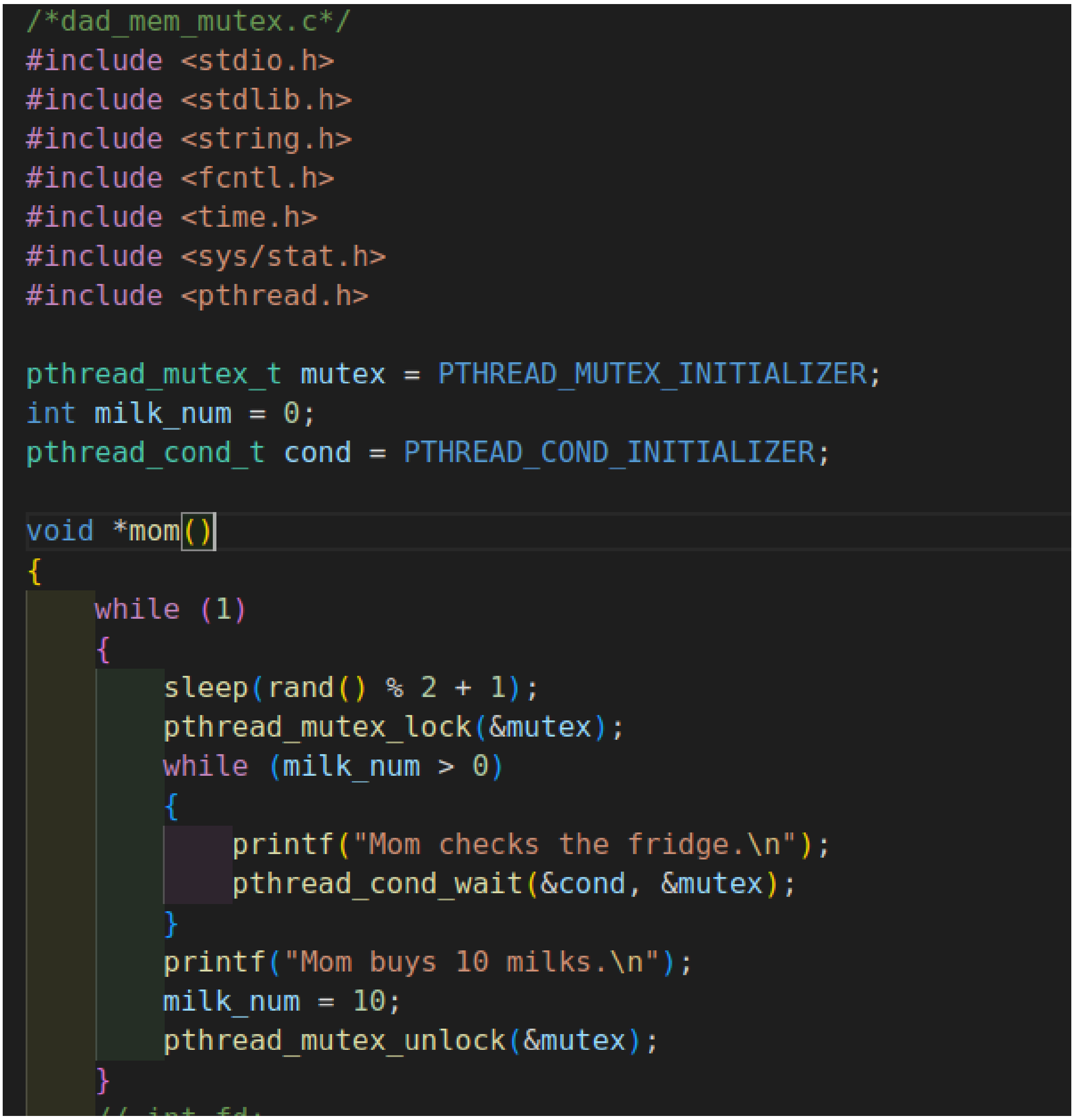


```c
int main(int argc, char *argv[])
{
    pthread_mutex_init(&mutex, NULL);

    srand(time(0));
    pthread_t p1, p2;
    int fd = open("fridge", O_CREAT | O_RDWR | O_TRUNC, 0777); // empty the fridge
    close(fd);
    // Create two threads (both run func)
    pthread_create(&p1, NULL, mom, NULL);
    pthread_create(&p2, NULL, dad, NULL);

    // Wait for the threads to end.
    pthread_join(p1, NULL);
    pthread_join(p2, NULL);

    pthread_mutex_destroy(&mutex);
}
```

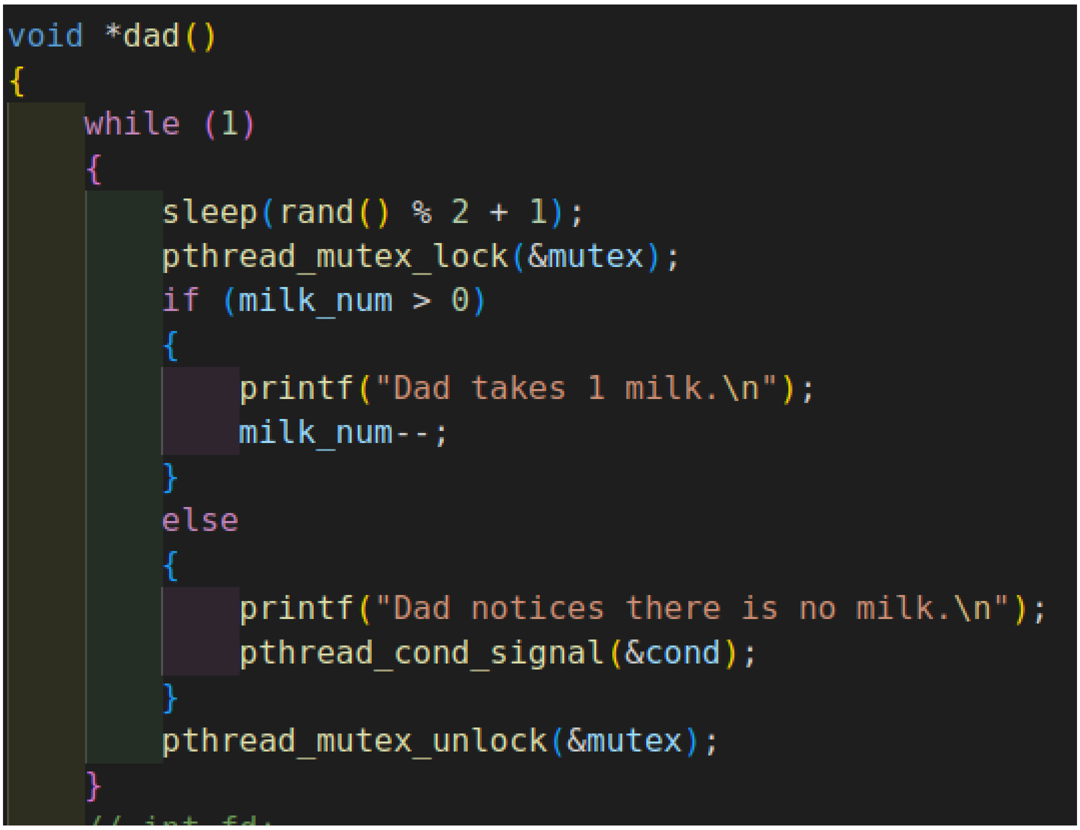


```
lrj11911808@lrj-virtual-machine:~/CS302_OS/lab/lab12$ ./a.out
Dad comes home.
Mom comes home.
Dad checks the fridge.
Dad goes to buy milk...
Dad comes back.
Dad puts milk in fridge and leaves.
Mom checks the fridge.
Mom closes the fridge and leaves.
lrj11911808@lrj-virtual-machine:~/CS302_OS/lab/lab12$ 
```

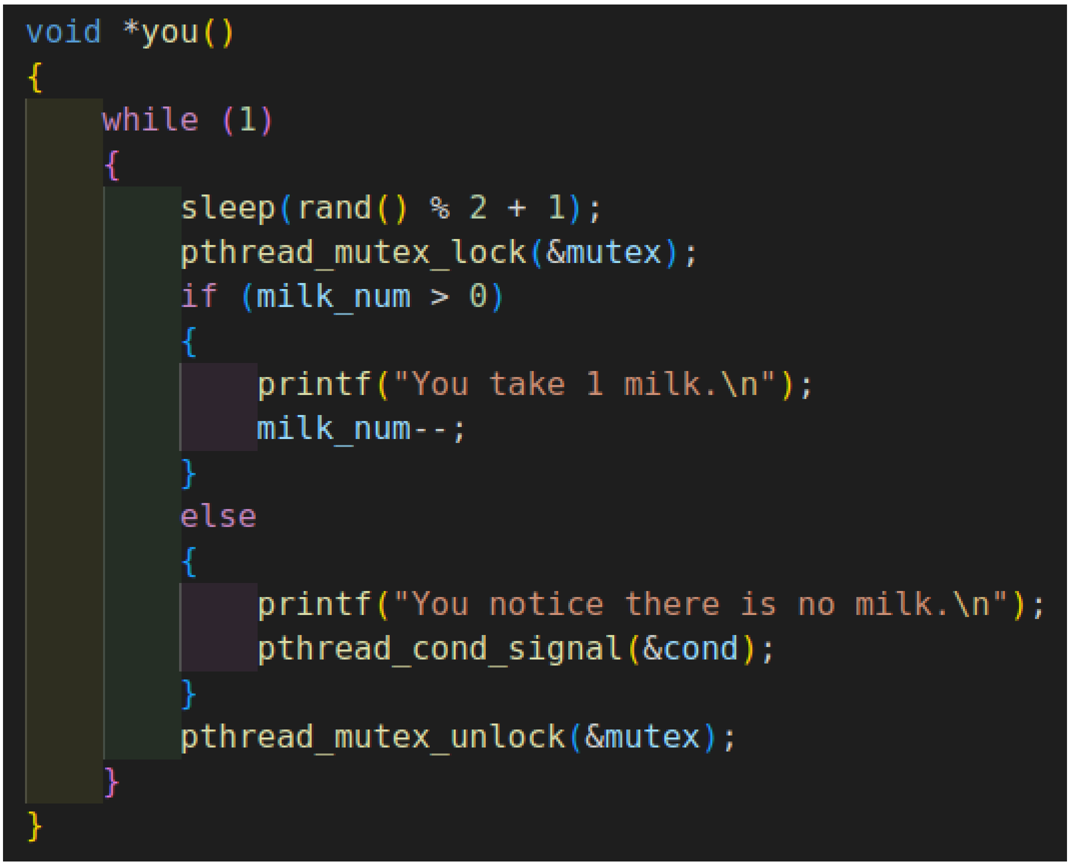
2.基于milk.c使用condition variable实现第四步中new problem的solution，报告中需要包含main()，mom()，dad()，you()，sister()的实现代码截图，以及运行结果截图



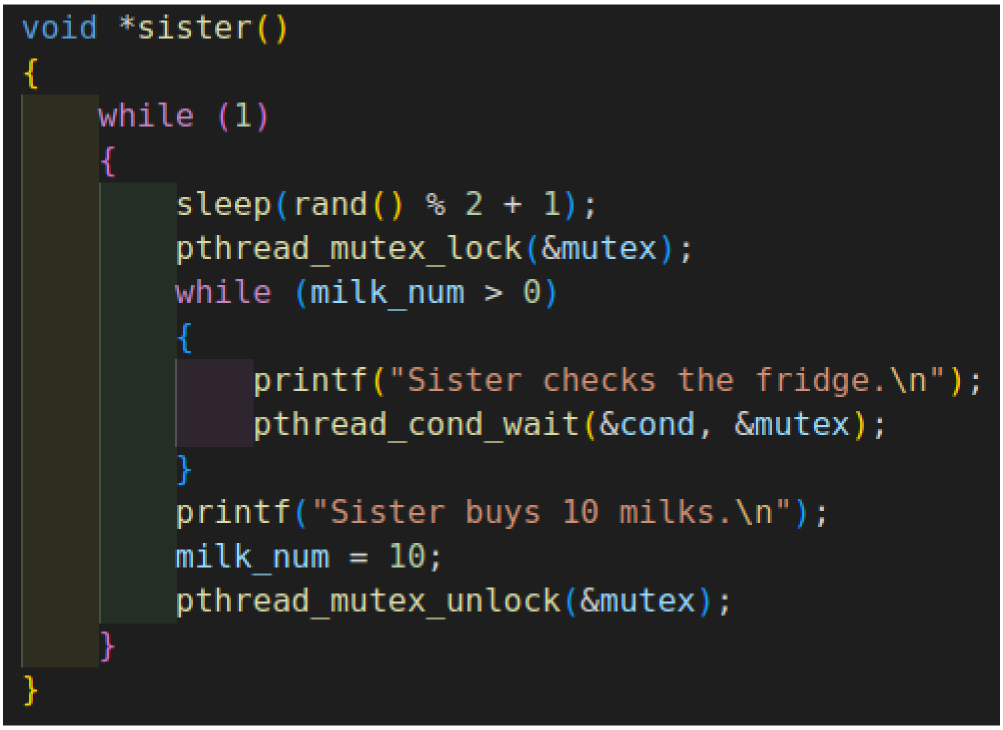
```
/*dad_mem_mutex.c*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <time.h>
#include <sys/stat.h>
#include <pthread.h>

pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
int milk_num = 0;
pthread_cond_t cond = PTHREAD_COND_INITIALIZER;

void *mom()
{
    while (1)
    {
        sleep(rand() % 2 + 1);
        pthread_mutex_lock(&mutex);
        while (milk_num > 0)
        {
            printf("Mom checks the fridge.\n");
            pthread_cond_wait(&cond, &mutex);
        }
        printf("Mom buys 10 milks.\n");
        milk_num = 10;
        pthread_mutex_unlock(&mutex);
    }
    // int fd;
```



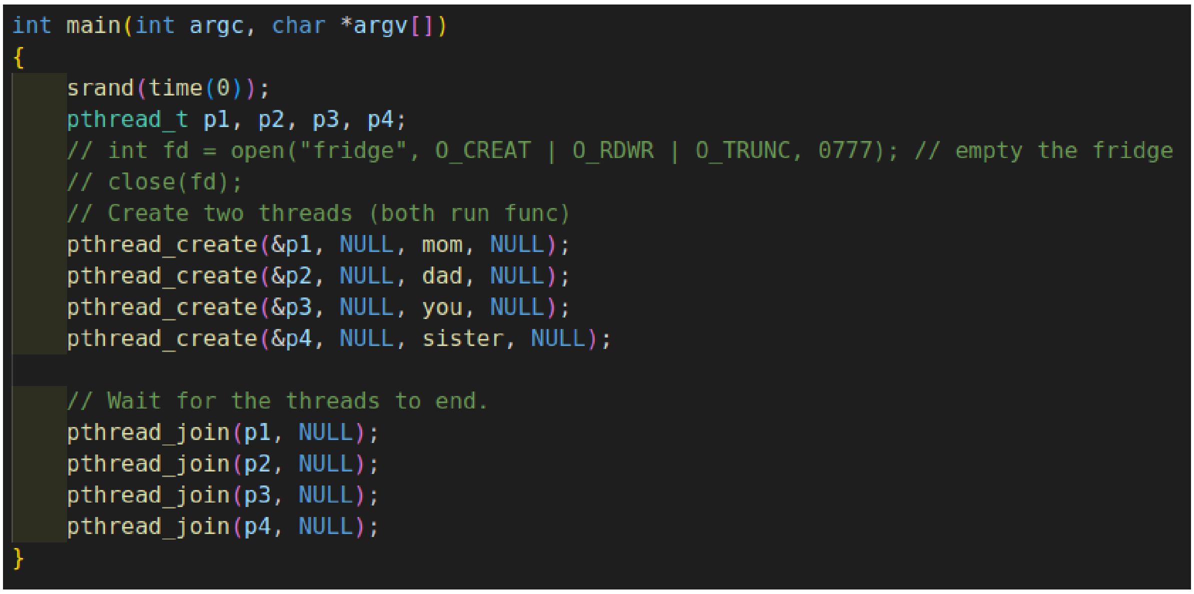
```
void *dad()
{
    while (1)
    {
        sleep(rand() % 2 + 1);
        pthread_mutex_lock(&mutex);
        if (milk_num > 0)
        {
            printf("Dad takes 1 milk.\n");
            milk_num--;
        }
        else
        {
            printf("Dad notices there is no milk.\n");
            pthread_cond_signal(&cond);
        }
        pthread_mutex_unlock(&mutex);
    }
    // int fd;
```



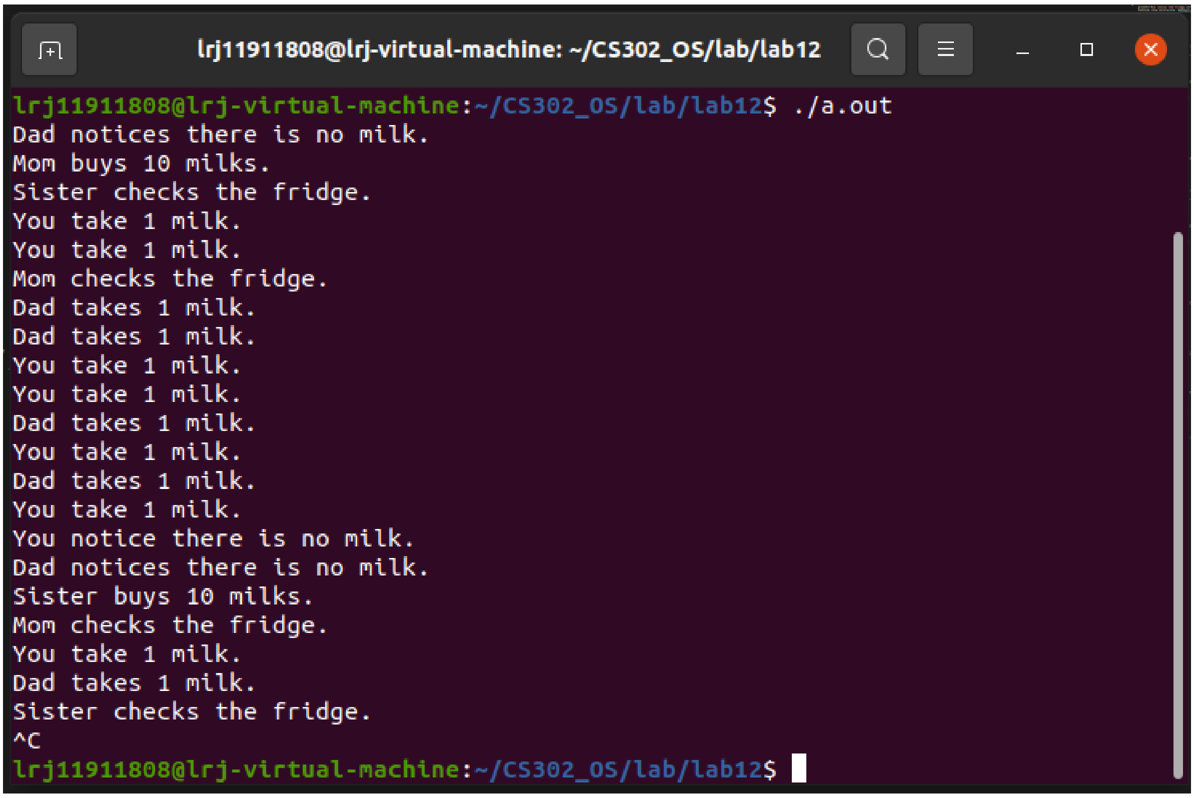
```
void *you()
{
    while (1)
    {
        sleep(rand() % 2 + 1);
        pthread_mutex_lock(&mutex);
        if (milk_num > 0)
        {
            printf("You take 1 milk.\n");
            milk_num--;
        }
        else
        {
            printf("You notice there is no milk.\n");
            pthread_cond_signal(&cond);
        }
        pthread_mutex_unlock(&mutex);
    }
}
```



```c
void *sister()
{
    while (1)
    {
        sleep(rand() % 2 + 1);
        pthread_mutex_lock(&mutex);
        while (milk_num > 0)
        {
            printf("Sister checks the fridge.\n");
            pthread_cond_wait(&cond, &mutex);
        }
        printf("Sister buys 10 milks.\n");
        milk_num = 10;
        pthread_mutex_unlock(&mutex);
    }
}
```



```c
int main(int argc, char *argv[])
{
    srand(time(0));
    pthread_t p1, p2, p3, p4;
    // int fd = open("fridge", O_CREAT | O_RDWR | O_TRUNC, 0777); // empty the fridge
    // close(fd);
    // Create two threads (both run func)
    pthread_create(&p1, NULL, mom, NULL);
    pthread_create(&p2, NULL, dad, NULL);
    pthread_create(&p3, NULL, you, NULL);
    pthread_create(&p4, NULL, sister, NULL);

    // Wait for the threads to end.
    pthread_join(p1, NULL);
    pthread_join(p2, NULL);
    pthread_join(p3, NULL);
    pthread_join(p4, NULL);
}
```



```
lrj11911808@lrj-virtual-machine:~/CS302_OS/lab/lab12$ ./a.out
Dad notices there is no milk.
Mom buys 10 milks.
Sister checks the fridge.
You take 1 milk.
You take 1 milk.
Mom checks the fridge.
Dad takes 1 milk.
Dad takes 1 milk.
You take 1 milk.
You take 1 milk.
Dad takes 1 milk.
You take 1 milk.
Dad takes 1 milk.
You take 1 milk.
You notice there is no milk.
Dad notices there is no milk.
Sister buys 10 milks.
Mom checks the fridge.
You take 1 milk.
Dad takes 1 milk.
Sister checks the fridge.
^C
lrj11911808@lrj-virtual-machine:~/CS302_OS/lab/lab12$
```