lab12-report

11911609 葛兆宁

- 暗号：654321

Q1

- code:

```
/*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;
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");
    }
    pthread_mutex_unlock(&mutex);
    close(fd);
}
```

```
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");
    }
    pthread_mutex_unlock(&mutex);
    close(fd);
}

int main(int argc, char *argv[])
{
    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);
}
```

- result:

```
11911609JohnnyGe@johnny-Ge-WXX9:~/OS/labs/lab12$ ./a.out
Mom comes home.
Dad comes home.
Mom checks the fridge.
```

```
Mom goes to buy milk...
Mon comes back.
Mom puts milk in fridge and leaves.
Dad checks the fridge.
Dad closes the fridge and leaves.
```

Q2

*code:

```
// Producers and Consumers.
// Two producers vs two consumers
// At any time, only one person can access count
#include <pthread.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
int count = 0;
pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
pthread_cond_t cond = PTHREAD_COND_INITIALIZER;
void *mom(void *arg)
{
    int i = 5;
    if (count == 5)
    {
        pthread_cond_signal(&cond);
        printf("MOM leaves\n");
        return NULL;
    }
    while (i--)
    {
        pthread_mutex_lock(&mutex);
        printf("MOM go out\n");
        count++;
        printf("MOM buy %d Bottles of milk\n", count);
        if (count > 0)
        {
            pthread_cond_signal(&cond);
        }
        printf("MOM leaves\n");
        pthread_mutex_unlock(&mutex);
    }
    return NULL;
}
void *sister(void *arg)
{
    int i = 5;
    if (count == 5)
    {
```

```
            pthread_cond_signal(&cond);
            printf("SISTER leaves\n");
            return NULL;
        }
        while (i--)
        {
            pthread_mutex_lock(&mutex);
            printf("SISTER go out\n");
            count++;
            printf("SISTER buy %d bottles of milk\n", count);
            if (count > 0)
            {
                pthread_cond_signal(&cond);
            }
            printf("SISTER leaves\n");
            pthread_mutex_unlock(&mutex);
        }
        return NULL;
    }
    void *dad(void *arg)
    {
        int i = 5;
        while (i--)
        {
            pthread_mutex_lock(&mutex);
            printf("DAD check the fridge\n");
            if (count <= 0)
            {
                printf("DAD call MOM and SISTER\n");
                pthread_cond_wait(&cond, &mutex);
                printf(" DAD end wait\n");
            }
            count--;
            printf("The milk is %d bottles\n", count);
            pthread_mutex_unlock(&mutex);
            printf("DAD leaves\n");
        }
        return NULL;
    }
    void *son(void *arg)
    {
        int i = 5;
        while (i--)
        {
            pthread_mutex_lock(&mutex);
            printf("Son check fridge\n");
            if (count <= 0)
            {
                printf("call mom and sister\n");
                pthread_cond_wait(&cond, &mutex);
                printf(" SON end wait\n");
            }
            count--;
            printf("The milk is %d bottlwes\n", count);
```

```
        pthread_mutex_unlock(&mutex);
        printf("SON leaves\n");
    }
    return NULL;
}
int main()
{
    pthread_t producethread1, producethread2, consumethread1,
consumethread2;
    pthread_create(&consumethread1, NULL, dad, NULL);
    pthread_create(&consumethread2, NULL, son, NULL);
    pthread_create(&producethread1, NULL, mom, NULL);
    pthread_create(&producethread2, NULL, sister, NULL);
    pthread_join(producethread1, NULL);
    pthread_join(consumethread1, NULL);
    pthread_join(producethread2, NULL);
    pthread_join(consumethread2, NULL);
    return 0;
}
```

- result:

```
11911609JohnnyGe@johnny-Ge-WXX9:~/OS/labs/lab12$ gcc milk.c -pthread
11911609JohnnyGe@johnny-Ge-WXX9:~/OS/labs/lab12$ ./a.out
DAD check the fridge
DAD call MOM and SISTER
Son check fridge
call mom and sister
MOM go out
MOM buy 1 Bottles of milk
MOM leaves
MOM go out
MOM buy 2 Bottles of milk
MOM leaves
MOM go out
MOM buy 3 Bottles of milk
MOM leaves
MOM go out
MOM buy 4 Bottles of milk
MOM leaves
MOM go out
MOM buy 5 Bottles of milk
MOM leaves
 SON end wait
The milk is 4 bottlwes
SON leaves
Son check fridge
The milk is 3 bottlwes
SON leaves
Son check fridge
The milk is 2 bottlwes
SON leaves
```

```
Son check fridge
The milk is 1 bottlwes
SON leaves
Son check fridge
The milk is 0 bottlwes
SON leaves
SISTER go out
SISTER buy 1 bottles of milk
SISTER leaves
SISTER go out
SISTER buy 2 bottles of milk
SISTER leaves
SISTER go out
SISTER buy 3 bottles of milk
SISTER leaves
SISTER go out
SISTER buy 4 bottles of milk
SISTER leaves
SISTER go out
SISTER buy 5 bottles of milk
SISTER leaves
 DAD end wait
The milk is 4 bottles
DAD leaves
DAD check the fridge
The milk is 3 bottles
DAD leaves
DAD check the fridge
The milk is 2 bottles
DAD leaves
DAD check the fridge
The milk is 1 bottles
DAD leaves
DAD check the fridge
The milk is 0 bottles
DAD leaves
```