666666 暗号

1

```
wangxiequn11910405@ubuntu-linux-20-04-desktop:~/Desktop/Parallels Shared Folders
/Home/CLionProjects/week12$ ./milk
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.
```

```
Mom closes the fridge and leaves.
wangxiequn11910405@ubuntu-linux-20-04-desktop:~/Desktop/Parallels Shared Folders/Home/CLionProjects/week12$ ca
t milk.c
/*dad_mem_mutex.c*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <time.h>
#include <sys/stat.h>
pthread_mutex_t mutex;
void *mom(){
    int fd;
    printf("Mom comes home.\n");
    pthread_mutex_lock(&mutex);
    sleep(rand()%2+1);
    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);
}

void *dad(){
    int fd;
    printf("Dad comes home.\n");
    pthread_mutex_lock(&mutex);
    sleep(rand()%2+1);
    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);
}

int main(int argc, char * argv[]) {
    srand(time(0));
    pthread_t p1, p2;
    pthread_mutex_init(&mutex, NULL);
    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);
```

2

```
wangxiequn11910405@ubuntu-linux-20-04-desktop:~/Desktop/Parallels Shared Folders/Home/CLionProjects/week12$ ./
milk
you comes home.
you add lock
you drink milk
Mom comes home.
Sister comes home.
dad comes home.
milks remain 4
you release lock
you add lock
you drink milk
milks remain 3
you release lock
you add lock
you drink milk
milks remain 2
you release lock
you add lock
you drink milk
milks remain 1
you release lock
you add lock
you drink milk
milks remain 0
you release lock
you add lock
begin wait
mom add lock
mom buy milk
milks remain 5
mom release lock
mom add lock
mom release lock
mom add lock
mom release lock
mom add lock
mom release lock
mom add lock
mom release lock
mom add lock
mom release lock
mom add lock
mom release lock
mom add lock
mom release lock
mom add lock
mom release lock
mom add lock
mom release lock
mom add lock
mom release lock
end wait
you drink milk
milks remain 4
you release lock
you add lock
you drink milk
milks remain 3
you release lock
you add lock
you drink milk
milks remain 2
you release lock
you add lock
you drink milk
milks remain 1
you release lock
you add lock
you drink milk
milks remain 0
you release lock
```

```
wangxiequn11910405@ubuntu-linux-20-04-desktop:~/Desktop/Parallels Shared Folders/Home/CLionProjects/week12$ ca
t milk.c
/*dad_mem_mutex.c*/
#include <pthread.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
pthread_cond_t milk = PTHREAD_COND_INITIALIZER;
int milks = 5;
void *mom(void *arg){

    printf("Mom comes home.\n");
    int t = 10;
    while (t--){
        pthread_mutex_lock(&mutex);
        printf("mom add lock\n");
        if (milks > 0)
        {

            pthread_cond_signal(&milk);
        } else{
            printf("mom buy milk\n");
            milks = 5;
            printf("milks remain %d\n",milks);
        }
        printf("mom release lock\n");
        pthread_mutex_unlock(&mutex);
    }


}
void *sister(void *arg){

    printf("Sister comes home.\n");
    int t = 10;

    while (t--){
        pthread_mutex_lock(&mutex);
        printf("sister add lock\n");
        if (milks > 0)
        {

            pthread_cond_signal(&milk);
        } else{
            printf("mom buy milk\n");
            milks = 5;
            printf("milks remain %d\n",milks);
        }
        pthread_mutex_unlock(&mutex);
    }


}

void *dad(void *arg){

    printf("dad comes home.\n");
    int t = 10;

    while (t--){
        pthread_mutex_lock(&mutex);
        printf("dad add lock\n");
        if (milks <= 0)
        {
            printf("begin wait\n");
            pthread_cond_wait(&milk, &mutex);
            printf("end wait\n");
        }
        printf("dad drink milk\n");
        milks--;
        printf("milks remain %d\n",milks);
        printf("dad release lock\n");
        pthread_mutex_unlock(&mutex);
```

```
void *sister(void *arg){

    printf("Sister comes home.\n");
    int t = 10;
```

```
    while (t--){
        pthread_mutex_lock(&mutex);
        printf("sister add lock\n");
        if (milks > 0)
        {

            pthread_cond_signal(&milk);
        } else{
            printf("mom buy milk\n");
            milks = 5;
            printf("milks remain %d\n",milks);
        }
        pthread_mutex_unlock(&mutex);
    }


}

void *dad(void *arg){

    printf("dad comes home.\n");
    int t = 10;

    while (t--){
        pthread_mutex_lock(&mutex);
        printf("dad add lock\n");
        if (milks <= 0)
        {
            printf("begin wait\n");
            pthread_cond_wait(&milk, &mutex);
            printf("end wait\n");
        }
        printf("dad drink milk\n");
        milks--;
        printf("milks remain %d\n",milks);
        printf("dad release lock\n");
        pthread_mutex_unlock(&mutex);
    }
    exit(0);
}
void *you(void *arg){

    printf("you comes home.\n");

    int t = 10;
    while (t--){
        pthread_mutex_lock(&mutex);
        printf("you add lock\n");
        if (milks <= 0)
        {
            printf("begin wait\n");
            pthread_cond_wait(&milk, &mutex);
            printf("end wait\n");
        }
        printf("you drink milk\n");
        milks--;
        printf("milks remain %d\n",milks);
        printf("you release lock\n");
        pthread_mutex_unlock(&mutex);
    }
    exit(0);
}




int main(int argc, char * argv[]) {
    pthread_t m, d, y, s;
    pthread_create(&m, NULL, mom, NULL);
    pthread_create(&d, NULL, dad, NULL);
    pthread_create(&y, NULL, you, NULL);
    pthread_create(&s, NULL, sister, NULL);
    pthread_join(m, NULL);
    pthread_join(d, NULL);
    pthread_join(y, NULL);
    pthread_join(s, NULL);
    return 0;
```