Week8 Report in Class (Fri56)

11911839 聂雨荷

Q1

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

```
/*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");
    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);
}
```

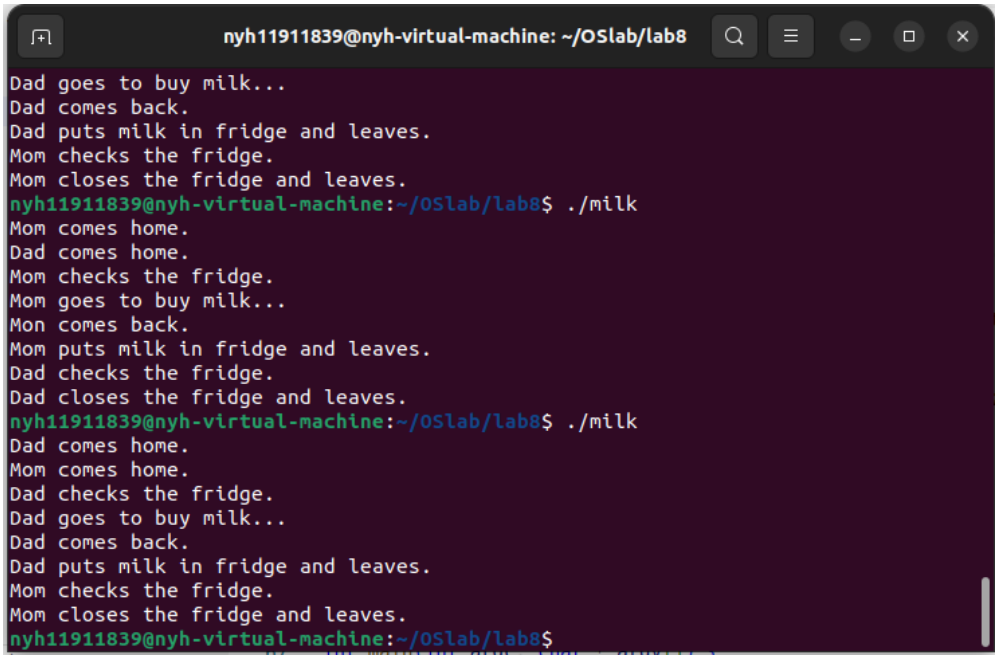
```c
void *dad(){
    int fd;
    printf("Dad comes home.\n");
    sleep(rand()%2+1);

    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);
}

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);
}
```



Q2

> 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

```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;
pthread_cond_t cond = PTHREAD_MUTEX_INITIALIZER;


// a new big fridge can contain a maximum of 100 bottles of milk
int CONTAINER = 5;
// initially, fridge is empty
int fridge = 0;


int check_fridge(){
    return fridge;
}

void take_milk(){
    fridge--;
}

void go_buy_milk(){
    fridge = CONTAINER;
}

void *dad(){
```

```
    while(1){
        sleep(rand()%2+1);
        pthread_mutex_lock(&mutex);
        printf("Dad checks the fridge.\n");
        int num = check_fridge();
        if(num > 0){
            printf("Dad drinks one milk.\n");
            take_milk();
            printf("Now fridge contains %i milk(s).\n", check_fridge());
            sleep(rand()%2+1);
        }else{
            printf("Dad finds that the fridge is empty.\n");
            pthread_cond_signal(&cond);
            sleep(rand()%2+1);
        }
        pthread_mutex_unlock(&mutex);
    }
}


void *you(){
    while(1){
        sleep(rand()%2+1);
        pthread_mutex_lock(&mutex);
        printf("You check the fridge.\n");
        int num = check_fridge();
        if(num > 0){
            printf("You drink one milk.\n");
            take_milk();
            printf("Now fridge contains %i milk(s).\n", check_fridge());
            sleep(rand()%2+1);
        }else{
            printf("You find that the fridge is empty.\n");
            pthread_cond_signal(&cond);
            sleep(rand()%2+1);
        }
        pthread_mutex_unlock(&mutex);
    }
}


void *mom(){
    while(1){
        sleep(rand()%2+1);
        pthread_mutex_lock(&mutex);
        printf("Mom checks the fridge.\n");
        while(check_fridge() > 0){
            printf("Mom finds that the fridge is not empty.\n");
            pthread_cond_wait(&cond, &mutex);
            sleep(rand()%2+1);
        }
        printf("Mom goes to buy some milks.\n");
        go_buy_milk();
        printf("Now fridge contains %i milk(s).\n", check_fridge());
        pthread_mutex_unlock(&mutex);
    }

}
```

```
void *sister(){
    while(1){
        sleep(rand()%2+1);
        pthread_mutex_lock(&mutex);
        printf("Sister checks the fridge.\n");
        while(check_fridge() > 0){
            printf("Sister finds that the fridge is not empty.\n");
            pthread_cond_wait(&cond, &mutex);
            sleep(rand()%2+1);
        }
        printf("Sister goes to buy some milks.\n");
        go_buy_milk();
        printf("Now fridge contains %i milk(s).\n", check_fridge());
        pthread_mutex_unlock(&mutex);
    }
}




int main(int argc, char * argv[]) {
    srand(time(0));
    pthread_t p_dad, p_you, p_mom, p_sister;


    int fd = open("fridge", O_CREAT|O_RDWR|O_TRUNC , 0777);  //empty the fridge
    close(fd);
    // Create two threads (both run func)
    pthread_create(&p_dad, NULL, dad, NULL);
    pthread_create(&p_you, NULL, you, NULL);
    pthread_create(&p_mom, NULL, mom, NULL);
    pthread_create(&p_sister, NULL, sister, NULL);


    // Wait for the threads to end.
    pthread_join(p_dad, NULL);
    pthread_join(p_you, NULL);
    pthread_join(p_mom, NULL);
    pthread_join(p_sister, NULL);
}
```

One of the possible execution result is:

```
nyh11911839@nyh-virtual-machine:~/OSlab/lab8$ ./newmilk
Dad checks the fridge.
Dad finds that the fridge is empty.
You check the fridge.
You find that the fridge is empty.
Sister checks the fridge.
Sister goes to buy some milks.
Now fridge contains 5 milk(s).
Mom checks the fridge.
Mom finds that the fridge is not empty.
Dad checks the fridge.
```

```
Dad drinks one milk.
Now fridge contains 4 milk(s).
You check the fridge.
You drink one milk.
Now fridge contains 3 milk(s).
Dad checks the fridge.
Dad drinks one milk.
Now fridge contains 2 milk(s).
Sister checks the fridge.
Sister finds that the fridge is not empty.
You check the fridge.
You drink one milk.
Now fridge contains 1 milk(s).
Dad checks the fridge.
Dad drinks one milk.
Now fridge contains 0 milk(s).
You check the fridge.
You find that the fridge is empty.
Dad checks the fridge.
Dad finds that the fridge is empty.
You check the fridge.
You find that the fridge is empty.
Sister goes to buy some milks.
Now fridge contains 5 milk(s).
Mom finds that the fridge is not empty.
Dad checks the fridge.
Dad drinks one milk.
Now fridge contains 4 milk(s).
You check the fridge.
You drink one milk.
Now fridge contains 3 milk(s).
Sister checks the fridge.
Sister finds that the fridge is not empty.
```