666666

Week12 Report

姓名：Yitong WANG(王奕童) 11910104@mail.sustech.edu.cn

学号：11910104

实验课时段：周五5-6节

实验课教师：Yun SHEN(沈昀) sheny@mail.sustech.edu.cn

实验课SA:

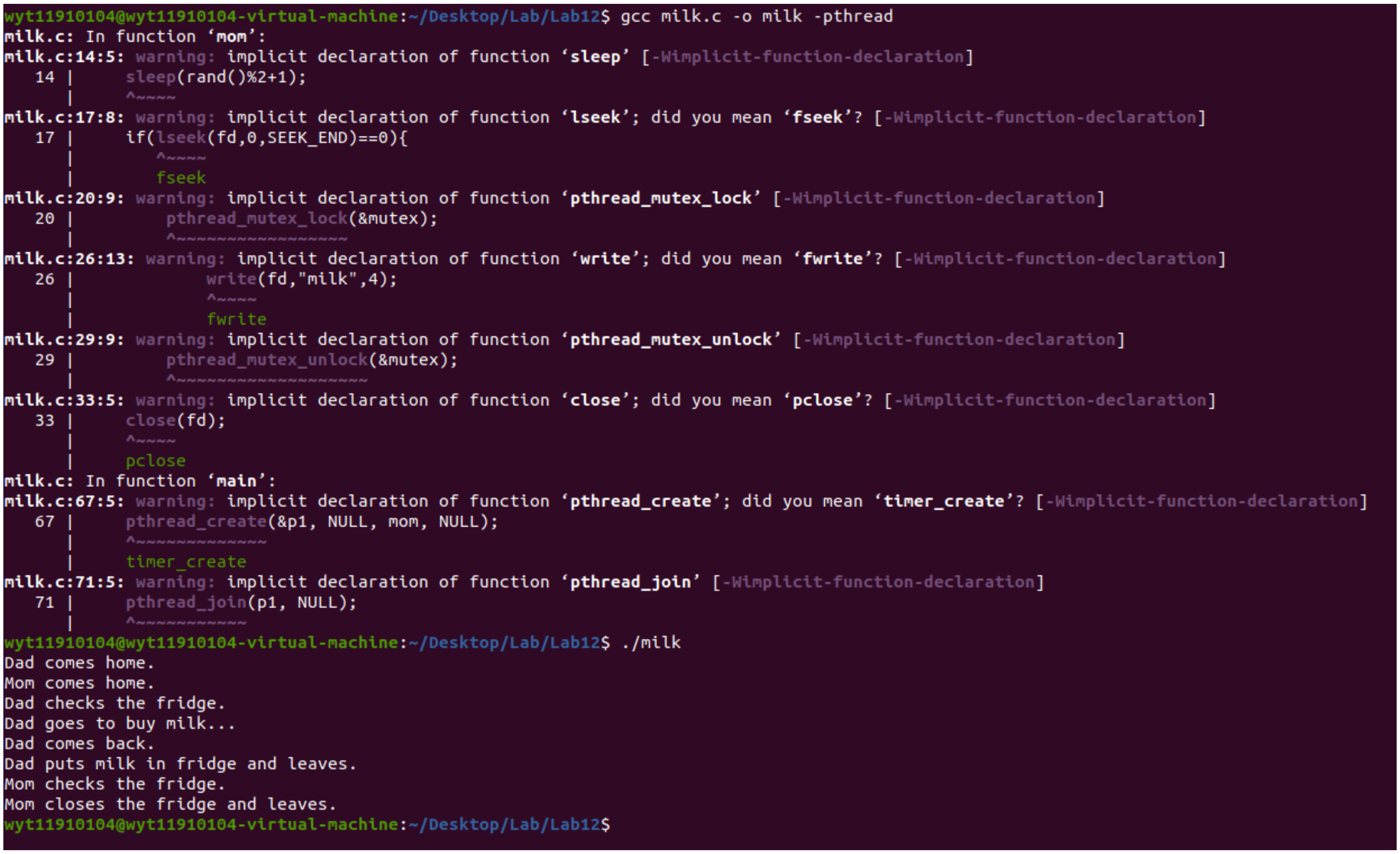
- Yining TANG(汤怡宁) 11811237@mail.sustech.edu.cn
- Yushan WANG(王宇杉) 11813002@mail.sustech.edu.cn

Q1. Milk problem I

Code Screenshot:

```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");
    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");
        pthread_mutex_lock(&mutex);
        //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");
        }
        pthread_mutex_unlock(&mutex);
    }else{
        printf("Mom closes the fridge and leaves.\n");
    }
    close(fd);
}

void *dad(){
    int fd;
    printf("Dad comes home.\n");
    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");
        printf("Dad goes to buy milk...\n");
        pthread_mutex_lock(&mutex);
        //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");
        }
        pthread_mutex_unlock(&mutex);
    } else{
        printf("Dad closes the fridge and leaves.\n");
    }
    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);
}
```

Execution Screenshot:



```
wyt11910104@wyt11910104-virtual-machine:~/Desktop/Lab/Lab12$ gcc milk.c -o milk -pthread
milk.c: In function ‘mom’:
milk.c:14:5: warning: implicit declaration of function ‘sleep’ [-Wimplicit-function-declaration]
   14 |     sleep(rand()%2+1);
      |     ^~~~~
milk.c:17:8: warning: implicit declaration of function ‘lseek’; did you mean ‘fseek’? [-Wimplicit-function-declaration]
   17 |     if(lseek(fd,0,SEEK_END)==0){
      |        ^~~~~
      |        fseek
milk.c:20:9: warning: implicit declaration of function ‘pthread_mutex_lock’ [-Wimplicit-function-declaration]
   20 |         pthread_mutex_lock(&mutex);
      |         ^~~~~~~~~~~~~~~~~~
milk.c:26:13: warning: implicit declaration of function ‘write’; did you mean ‘fwrite’? [-Wimplicit-function-declaration]
   26 |             write(fd,"milk",4);
      |             ^~~~~
      |             fwrite
milk.c:29:9: warning: implicit declaration of function ‘pthread_mutex_unlock’ [-Wimplicit-function-declaration]
   29 |         pthread_mutex_unlock(&mutex);
      |         ^~~~~~~~~~~~~~~~~~~~
milk.c:33:5: warning: implicit declaration of function ‘close’; did you mean ‘pclose’? [-Wimplicit-function-declaration]
   33 |     close(fd);
      |     ^~~~~
      |     pclose
milk.c: In function ‘main’:
milk.c:67:5: warning: implicit declaration of function ‘pthread_create’; did you mean ‘timer_create’? [-Wimplicit-function-declaration]
   67 |     pthread_create(&p1, NULL, mom, NULL);
      |     ^~~~~~~~~~~~~~
      |     timer_create
milk.c:71:5: warning: implicit declaration of function ‘pthread_join’ [-Wimplicit-function-declaration]
   71 |     pthread_join(p1, NULL);
      |     ^~~~~~~~~~~~
wyt11910104@wyt11910104-virtual-machine:~/Desktop/Lab/Lab12$ ./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.
wyt11910104@wyt11910104-virtual-machine:~/Desktop/Lab/Lab12$
```

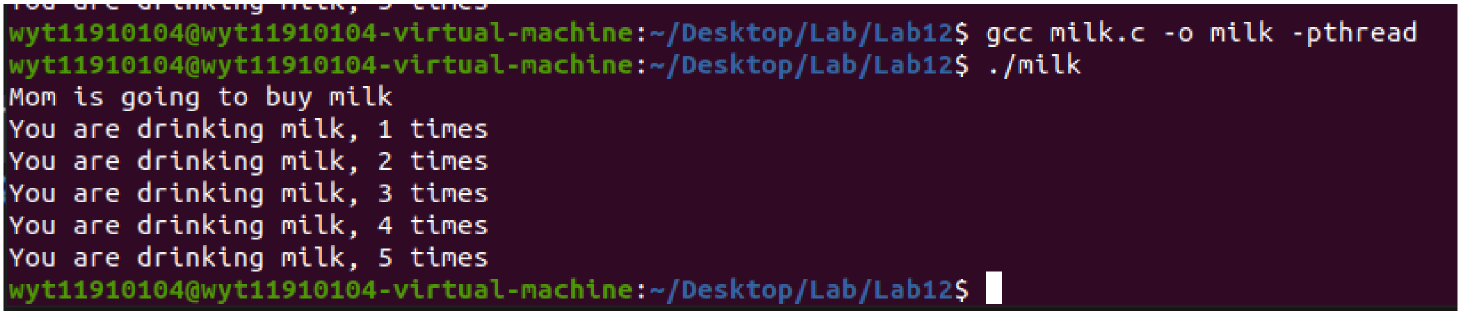
Q2. Milk problem II

Code Screenshot:

```c
/*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>
#include <unistd.h>
#include <sys/types.h>

int youDrink = 0, dadDrink = 0;

int fd;
int milk;
int cond_signal = 0;
pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
pthread_cond_t cond = PTHREAD_COND_INITIALIZER;

int check_fridge(){
    fd=open("fridge", O_CREAT|O_RDWR|O_APPEND, 0777);
    milk = lseek(fd, 0, SEEK_END);
    close(fd);
    return milk;
}

void take_milk(){
    fd=open("fridge", O_CREAT|O_RDWR|O_APPEND, 0777);
    write(fd,"milk",milk-1);
    milk--;
    close(fd);

}

void buy_milk(){
    fd=open("fridge", O_CREAT|O_RDWR|O_APPEND, 0777);
    write(fd,"milk",5);
    milk = 5;
    close(fd);
}
```

```
void *mom(){
    sleep(rand()%2+1);
    while(1) {
        pthread_mutex_lock(&mutex);
        while(check_fridge() > 0){
            pthread_cond_wait(&cond, &mutex);
        }

        printf("Mom is going to buy milk\n");

        buy_milk();
        pthread_mutex_unlock(&mutex);
    }

}

void *sister(){
    sleep(rand()%2+1);
    while(1) {
        pthread_mutex_lock(&mutex);
        while(check_fridge() > 0){
            pthread_cond_wait(&cond, &mutex);
        }

        printf("Sister is going to buy milk\n");

        buy_milk();
        pthread_mutex_unlock(&mutex);
    }

}

void *you(){
    sleep(rand()%2+1);
    while(1){
        pthread_mutex_lock(&mutex);
        int num = check_fridge();
        if(num > 0) {

            take_milk();
            take_milk();
            youDrink++;
            printf("You are drinking milk, %d times\n", youDrink);
            if(youDrink >= 5){
                exit(0);
            }
        }else {
            //printf("You cannot get any milk, sorry...\n");
            pthread_cond_signal(&cond);
        }

        pthread_mutex_unlock(&mutex);
    }
}

void *dad(){
    sleep(rand()%2+1);
    while(1){
        pthread_mutex_lock(&mutex);
        int num = check_fridge();
        if(num > 0) {
            take_milk();
            dadDrink++;
            printf("Dad is drinking milk, %d times\n", dadDrink);
            if(dadDrink >= 5){
                exit(0);
            }
        }else {
            //printf("Dad cannot get any milk, sorry...\n");
            pthread_cond_signal(&cond);
        }

        pthread_mutex_unlock(&mutex);
    }
}

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 four threads (both run func)
    // Create four threads (both run func)
    pthread_create(&p2, NULL, you, NULL);

    pthread_create(&p2, NULL, dad, NULL);

    pthread_create(&p1, NULL, mom, NULL);
    pthread_create(&p2, NULL, sister, NULL);


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

Execution Screenshot:



```
You are drinking milk, 5 times
wyt11910104@wyt11910104-virtual-machine:~/Desktop/Lab/Lab12$ gcc milk.c -o milk -pthread
wyt11910104@wyt11910104-virtual-machine:~/Desktop/Lab/Lab12$ ./milk
Mom is going to buy milk
You are drinking milk, 1 times
You are drinking milk, 2 times
You are drinking milk, 3 times
You are drinking milk, 4 times
You are drinking milk, 5 times
wyt11910104@wyt11910104-virtual-machine:~/Desktop/Lab/Lab12$ 
```