OS Lab Final Exam

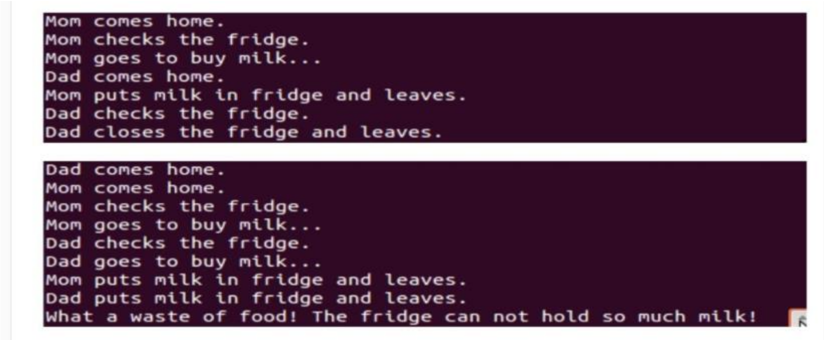
Q2: The too much milk problem Your family drink milk every morning. Mom and Dad are all used to checking the fridge when they arrive home. If milk run out, he or she will leave home to buy milk. The fridge is small in your home that only one bottle of milk can be put in it at a time. Mom and Dad always arrive home at different time. mom.c and dad.c are two programs that simulate the above situation. When process mom or dad buys a bottle of milk, it writes string `"milk"` to a file `fridge` (the file is empty at the beginning). Run these two programs and see what will happen.

Compile and run like this:

- `$ gcc mom.c -o mom`
- `$ gcc dad.c -o dad`
- `$ ./mom & ./dad &`

```
/*mom.c*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <fcntl.h>

#include <sys/stat.h>

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

    int fd;

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

    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(2);
    write(fd,"milk ",5);
        printf("Mom puts milk in fridge and leaves.\n");
        if(lseek(fd,0,SEEK_END)>5)
            printf("What a waste of food! The fridge can not hold so much milk!\n");
    }else{
        printf("Mom closes the fridge and leaves.\n");
    }
    close(fd);
    return 0;
}
```

```c
/*dad.c*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <fcntl.h>

#include <sys/stat.h>

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

    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");

        sleep(rand()%2+1);

             write(fd,"milk ",5);

        printf("Dad puts milk in fridge and leaves.\n");

        if(lseek(fd,0,SEEK_END)>5)

            printf("What a waste of food! The fridge can not hold so much milk!\n");

    }else{

        printf("Dad closes the fridge and leaves.\n");

    }

    close(fd);

    return 0;
}
```



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

```
Dad comes home.
Mom comes home.
Mom checks the fridge.
Mom goes to buy milk...
Dad checks the fridge.
Dad goes to buy milk...
Mom puts milk in fridge and leaves.
Dad puts milk in fridge and leaves.
What a waste of food! The fridge can not hold so much milk!
```

Analysis: If we run `mom.c` and `dad.c` many times, we could see different results as shown above. The result of these two programs depends on the order in which the fridge is accessed.

Hints:

To solve the too much milk problem, mom and dad decide to lock the fridge. There is only one key to open the lock, the one who gets the key can check the fridge. That is to say, if mom gets the key and checks fridge, dad can not open the fridge until mom releases the key. Mom will releases the key until she ensures there is milk in fridge.

Note that programs using the POSIX semaphores API must be compiled with -pthread to link against the real-time library. So you need compile these two programs like this:

```
$ gcc mom.c -pthread -o mom
```

```
$ gcc dad.c -pthread -o dad
```

```
$ ./mom & ./dad &
```

Lab Task Questions:

- What is the problem with these programs?
- Resolve this problem with the solution using your code updates in `mom.c` and `dad.c` program. Use multi-threading and semaphores with proper deadlock avoidance and prevention.