

전북대학교 컴퓨터인공지능학부
리눅스프로그래밍 프로젝트 보고서

사물함 관리 프로그램



16조

202323007 이진선

202312854 허예린



개요

1 . 프로젝트 주제와 목표

2. 프로젝트의 기본 기능

3. 프로젝트의 흐름

4. 추가 기능

5. 추가 기능 분석

6. 프로젝트 내 역할 분담

7. 프로젝트를 마치며

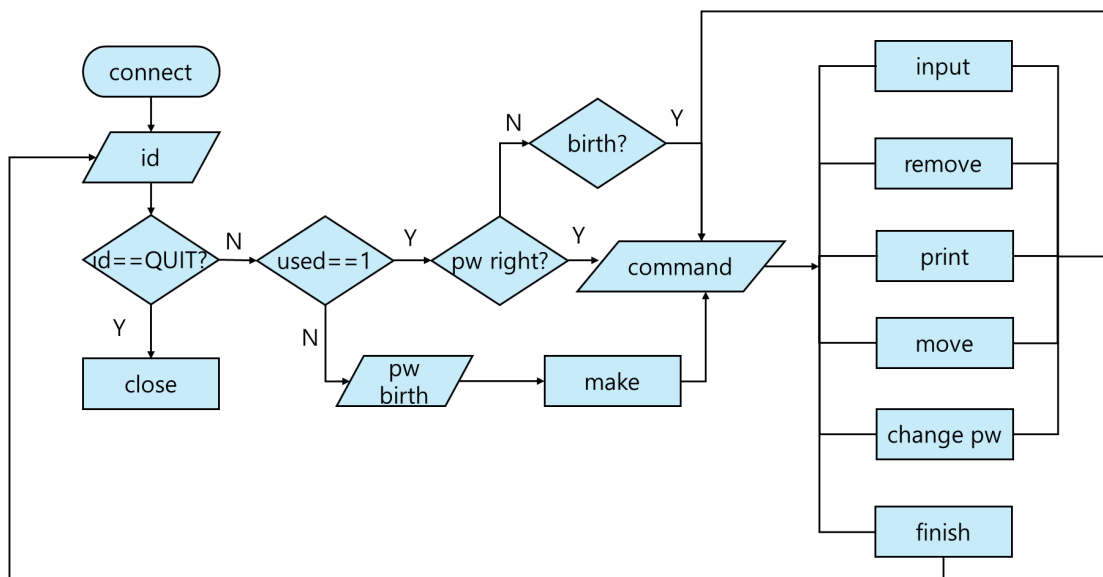
1. 프로젝트의 주제와 목표

- 서버클라이언트 모델로 동작하는 사물함 관리 시스템을 시스템 호출을 사용하여 구현한다.
- 서버는 사물함 관리하는 역할을 수행하고 클라이언트는 사물함 관리 시스템의 고객 역할을 수행한다.
- 서버는 사물함 관리, 클라이언트들에게 사물함 제공 및 비밀번호를 설정하는 기능을 활용하고, 클라이언트는 사물함 사용 및 비밀번호 설정을 수행하도록 만들고 위 기능을 향상시켜 개발한다.

2. 프로젝트의 기본 기능

- 서버
 - * 사용자에게 사물함에 대한 정보를 제공
 - * 사용자에게 사물함 할당 시 패스워드를 지정하도록 함
 - * 사용자가 본인 사물함에 접근 시 입력하는 패스워드가 맞는지 확인하고, 맞 으면 접근 가능하도록 함.
 - * 내용물 저장
- 클라이언트
 - * 원하는 사물함에 접근
 - * 해당 사물함의 비밀번호 설정 및 입력 기능
 - * 내용물 입력

3. 프로젝트의 흐름



흐름도

```
leejinsun@ubuntu:~/Downloads/project$ ./client
-----
locker Management
welcome to the store
-----

You can control your own locker system
You can choose your own locker and the number of locker
And you can put your own item in your locker
-----Enjoy-----

locker number (over: QUIT type): 1
password(XXXXXXX): 12345678
password check(XXXXXXX): 12345678
birth (YYYYMMDD): 20030303
1: put in the locker, 2: get out in the locker, 3: check locker inside, 4: move items, 5: change password, 6: quit
```

실행 화면

4. 추가 기능

- 물건을 추가할 때, 동적 할당을 통해 메모리 할당
- 비밀번호를 3번 틀릴 시, 사물함 잠금(30초)
- 비밀번호를 5번 틀릴 시, 개인정보 확인을 통한 새 비밀번호 설정 가능
- 사물함 생성 시, 비밀번호 생성할 때 2 번 입력
- 사물함 내의 내용물 삭제
- 사물함 내의 내용물 확인
- 사물함 내의 내용물 다른 사물함으로 이동
- 비밀번호 변경 기능

5. 추가 기능 분석

1) 물건을 추가할 때, 동적 할당을 통해 메모리 할당

물건을 넣을 때, 동적 할당으로 메모리를 할당해 물건을 저장하지 않은 경우 메모리 절약
물건이 있을 시에만 메모리 사용 가능

```
if (count < 40) {
    locker[locker_id].item[count] = (char *)malloc(BUFSIZE);
    if (locker[locker_id].item[count] != NULL) {
        strcpy(locker[locker_id].item[count], buffer);
        count++;
    } else {
```

1)server.c

```
Command number: 1
input item (Q to stop): I CAN TYPE ANY WORDS. IT CAN WRITE WHATEVER I WANT TO WRITE. THIS IS THE WORD WE CAN CHECK ABOUT.
input item (Q to stop): can
input item (Q to stop): LOCKER
input item (Q to stop): PENCIL
input item (Q to stop): PENCIL CASE
input item (Q to stop): IPHONE
input item (Q to stop): NOTEBOOK
input item (Q to stop): LINUX
input item (Q to stop): PROGRAMMING
input item (Q to stop): PROJECT
input item (Q to stop): Q
++++items stored++++
```

1)실행 화면

2) 비밀번호를 3번 틀릴 시, 사물함 잠금(30초)

비밀번호를 틀리면 서버에서 pwTime 전송, 클라이언트에서 받아 틀린 횟수 출력

3번 틀릴 시, sleep으로 30초 클라이언트 잠금

5초마다 남은 시간 출력

```
if (password == locker[locker_id].pw) {
    write(client_sock, "login", BUFSIZE);
    break;
}
write(client_sock, "pwTime", BUFSIZE);
pwTime++;
```

2)server.c

```
if (strcmp(buffer, "pwTime") == 0) {
    printf("####%d count wrong####\n", ++pwTime);
    if (pwTime == 3) {
        printf("wrong password Three times\n");
        printf("locker system is locked for 30sec\n");
        printf("please wait for the System\n");
        for (int i=1;i<=30;i++){if (i%5==0) printf("wait %d sec\n",30-i);
        sleep(1);}}
```

2)client.c

```
-----Enjoy-----
locker number (over: QUIT type): 1

password(XXXXXXXX): 00000000
####1 count wrong###

password(XXXXXXXX): 00000000
####2 count wrong###

password(XXXXXXXX): 00000000
####3 count wrong###
wrong password Three times
locker system is locked for 30sec
please wait for the System
wait 25 sec
wait 20 sec
wait 15 sec
wait 10 sec
wait 5 sec
wait 0 sec

password(XXXXXXXX): 00000000
```

2)실행 화면

3) 비밀번호를 5번 틀릴 시, 개인정보 확인을 통한 새 비밀번호 설정 가능

시도 횟수가 5가 되면 생성시 기입한 생일 정보 확인, 맞으면 새 비밀번호 설정, 로그인

```
if (pwTime == 5) {
    read(client_sock, buffer, BUFSIZE);
    locker_birth = atoi(buffer);
    if (locker[locker_id].birth == locker_birth) {
        write(client_sock, "access", BUFSIZE);
        read(client_sock, buffer, BUFSIZE);
        locker[locker_id].pw = atoi(buffer);
        write(client_sock, "login", BUFSIZE);
        pwTime = 0;
        break;
    }
}
```

3)server.c

```

if (pwTime == 5) {
    printf("birthday (YYYYMMDD): ");
    fgets(buffer, BUFSIZE, stdin);
    clear_newline(buffer);
    write(sock, buffer, BUFSIZE);
    read(sock, buffer, BUFSIZE);
    if (strcmp(buffer, "access") == 0) {
        printf("new password(XXXXXXXX): ");
        fgets(buffer, BUFSIZE, stdin);
        clear_newline(buffer);
        write(sock, buffer, BUFSIZE);
        read(sock, buffer, BUFSIZE);
        printf("++++new password record success++++\n");
    }
}

```

3)client.c

```

password(XXXXXXXX): 00000000
####5 count wrong####

birthday (YYYYMMDD): 20030303

new password(XXXXXXXX): 11111111

++++new password record success++++

```

3)실행 화면

4) 사물함 생성 시, 비밀번호 생성할 때 2 번 입력

사물함 생성 시, 비밀번호 2번 입력하고 맞으면 생일 정보 입력 받고 사물함 생성

```

do {
    read(client_sock, buffer, BUFSIZE);
    isPassword = atoi(buffer);
    if (isPassword == password) {
        write(client_sock, "setbirth", BUFSIZE);
        break;
    }
    write(client_sock, "notmatch", BUFSIZE);
} while (isPassword != password);

```

4)server.c

```

do {
    printf("password check(XXXXXXXX): ");
    fgets(buffer, BUFSIZE, stdin);
    clear_newline(buffer);
    write(sock, buffer, BUFSIZE);
    read(sock, buffer, BUFSIZE);
} while (strcmp(buffer, "notmatch") == 0);
if (strcmp(buffer, "setbirth") == 0) {
    printf("birth (YYYYMMDD): ");
    fgets(buffer, BUFSIZE, stdin);
    clear_newline(buffer);
    write(sock, buffer, BUFSIZE);
    read(sock, buffer, BUFSIZE);
}

```

4)client.c

```

locker number (over: QUIT type): 1

password(XXXXXXXX): 12345678
password check(XXXXXXXX): 12345678

birth (YYYYMMDD): 20030303

```

4) 실행 화면

5) 사물함 내의 내용물 삭제

Q를 입력할 때까지 물건 받아 위치 찾고 메모리 해제, 여러개 있으면 가장 먼저 넣은 물건 삭제

```

for (int i = 0; i < count; i++) {
    if (locker[locker_id].item[i] != NULL
        && strcmp(locker[locker_id].item[i], buffer) == 0) {
        free(locker[locker_id].item[i]);
        for (int j = i; j < count - 1; j++) {
            locker[locker_id].item[j]
                = locker[locker_id].item[j + 1];
        }
        locker[locker_id].item[count - 1] = NULL;
        locker[locker_id].itemCount--;
        break;
    }
}

```

5)server.c

```

while (1) {
    printf("delete item (Q to stop): ");
    fgets(buffer, BUFSIZE, stdin);
    clear_newline(buffer);
    write(sock, buffer, BUFSIZE);
    if (strcmp(buffer, "Q") == 0) break;
}

```

5)client.c

```

Command number: 2
--- inside the locker ---
I CAN TYPE ANY WORDS. IT CAN WRITE WHATEVER I WANT TO WRITE. THIS IS THE WORD WE CAN CHECK ABOUT.
CAN
LOCKER
BOOK
PENCIL
PENCIL CASE
IPHONE
NOTEBOOK
BOOK
LINUX
PROGRAMMING
PROJECT
q
I CAN TYPE ANY WORDS. IT CAN WRITE WHATEVER I WANT TO WRITE. THIS IS THE WORD WE CAN CHECK ABOUT.
can
LOCKER
PENCIL
PENCIL CASE
IPHONE
NOTEBOOK
LINUX
PROGRAMMING
PROJECT
--- end ---
delete item (Q to stop): I CAN TYPE ANY WORDS. IT CAN WRITE WHATEVER I WANT TO WRITE. THIS IS THE WORD WE CAN CHECK ABOUT.
delete item (Q to stop): CAN
delete item (Q to stop): LOCKER
delete item (Q to stop): BOOK
delete item (Q to stop): PENCIL
delete item (Q to stop): PENCIL CASE
delete item (Q to stop): IPHONE
delete item (Q to stop): NOTEBOOK
delete item (Q to stop): BOOK
delete item (Q to stop): LINUX
delete item (Q to stop): PROGRAMMING
delete item (Q to stop): PROJECT
delete item (Q to stop): q
delete item (Q to stop): Q
++++items removed++++

```

5)실행 화면1

```

Command number: 2
--- inside the locker ---
I CAN TYPE ANY WORDS. IT CAN WRITE WHATEVER I WANT TO WRITE. THIS IS THE WORD WE CAN CHECK ABOUT.
can
LOCKER
PENCIL
PENCIL CASE
IPHONE
NOTEBOOK
LINUX
PROGRAMMING
PROJECT
--- end ---
delete item (Q to stop): Q
++++items removed++++

```

5) 실행 화면2

6) 사물함 내의 내용물 확인

물건 개수만큼 물건 출력, 끝까지 출력하면 서버에서 Q를 보내 끝났음을 알림

```

int count = locker[locker_id].itemCount;
for (int i = 0; i < count; i++) {
    write(client_sock, locker[locker_id].item[i], BUFSIZE);
}
write(client_sock, "Q", BUFSIZE);

```

6)server.c

```

while (1) {
    read(sock, buffer, BUFSIZE);
    if (strcmp(buffer, "Q") == 0) break;
    printf("%s\n", buffer);
}

```

6)client.c

```

Command number: 3
--- inside the locker ---
I CAN TYPE ANY WORDS. IT CAN WRITE WHATEVER I WANT TO WRITE. THIS IS THE WORD WE CAN CHECK ABOUT.
can
LOCKER
PENCIL
PENCIL CASE
IPHONE
NOTEBOOK
LINUX
PROGRAMMING
PROJECT
--- end ---

```

6) 실행 화면

7) 사물함 내의 내용물 다른 사물함으로 이동

옳길 물건 받아 삭제하고 옳길 사물함이 사용중인 사물함이면 물건 이동

```

if (locker[move_id].used == 1) {
    locker[move_id].item[locker[move_id].itemCount]
    = (char *)malloc(BUFSIZE);
    strcpy(locker[move_id].item[locker[move_id].itemCount], moveItem);
    locker[move_id].itemCount++;
    write(client_sock, "success", BUFSIZE);
} else {
    write(client_sock, "fail", BUFSIZE);
}
free(moveItem);

```

7)server.c


```

printf("move the item: ");
fgets(buffer, BUFSIZE, stdin);
clear_newline(buffer);
write(sock, buffer, BUFSIZE);
printf("move to locker: ");
fgets(buffer, BUFSIZE, stdin);
clear_newline(buffer);
write(sock, buffer, BUFSIZE);
read(sock, buffer, BUFSIZE);
if (strcmp(buffer, "fail") == 0) {
    printf("####unable####\n");
} else {
    printf("++++%s++++\n", buffer);
}

```

7)client.c

```

Command number: 4
--- inside the locker ---
I CAN TYPE ANY WORDS. IT CAN WRITE WHATEVER I WANT TO WRITE. THIS IS THE WORD WE CAN CHECK ABOUT.
LOCKER
PENCIL
PENCIL CASE
IPHONE
NOTEBOOK
LINUX
PROGRAMMING
PROJECT
--- end ---
move the item: PENCIL
move to locker: 2
++++success++++

```

7)실행화면

8) 비밀번호 변경 기능

현재 비밀번호 확인, 5번 이내로 틀리면 새 비밀번호 설정

```

if(trycount<5){
    read(client_sock,buffer,BUFSIZE);
    try_pw=atoi(buffer);
    while(1){
        read(client_sock,buffer,BUFSIZE);
        int check_pw=atoi(buffer);
        if(try_pw==check_pw) {
            write(client_sock,"M",BUFSIZE);
            locker[locker_id].pw=try_pw;
            break;
        }
    }
}

```

8)server.c

```

if (trycount < 5) {
    printf("new password input: ");
    fgets(buffer, BUFSIZE, stdin);
    clear_newline(buffer);
    write(sock, buffer, BUFSIZE);

    while (1) {
        printf("confirm new password: ");
        fgets(buffer, BUFSIZE, stdin);
        clear_newline(buffer);
        write(sock, buffer, BUFSIZE);
        read(sock, buffer, BUFSIZE);
        if (strcmp(buffer, "M") == 0) {
            printf("++++passwords match++++\n");
            break;
        }
        printf("####passwords do not match. try again####\n");
    }
}

```

8)client.c

```
Command number: 5
current password input: 11111111

new password input: 12345678
confirm new password: 12345678

++++passwords match++++
```

8)실행 화면

6. 프로젝트 내 역할 분담

이진선 : 추가기능 1,3,4 주로 제작, 코드 수정, 대본 제작, 발표, 보고서 제작

허예린 : 네트워크 통신 작성, 추가기능 2,5,6 주로 제작, 코드 수정, ppt 제작, 발표, 보고서 제작

7. 프로젝트를 마치며

한학기동안 공부했던 자료들을 활용해 유저가 직접 활용할 수 있는 코드를 작성하고 이를 통신할 수 있도록 만들었다는 것에 있어서 새로운 경험을 한 기분이었습니다. 자료를 보며 프로그램 동작 방법을 공부하고 제대로 작동 안되는 부분들은 다양한 방법을 찾아 코드를 수정해가는 과정을 통해 더 많은 것들을 알게 된 계기가 되었습니다. 방학동안 학기 중에 작성했던 코드를 토대로 수정해나아가며 새로운 기능을 추가해가고 코드를 정리하면서 더 좋은 프로그램을 만들어보고 싶다고 생각했습니다.