

2021 유닉스시스템 프로젝트

NANG Locker

IT정보공학과 202012178 김아은

IT정보공학과 202018392 박나현

목차

1. 서론

1.1 프로젝트 개요	3
1.2 프로젝트 설명	3
1.3 flow chart	4

2. 본론

2.1 프로젝트 장점	5
2.2 프로젝트 내용	8

3. 결론

3.1 맡은 일	9
3.2 프로젝트 수행 후기	9

1. 서론

1.1 프로젝트 개요

이번 프로젝트는 유닉스시스템 수업시간에 배운 내용을 응용하여 사물함 관리 시스템을 구현하는 것이다. Server에서는 사물함을 관리하는 역할을 수행하고, Client에서는 사물함 관리 시스템의 고객 역할을 수행한다.

server와 client의 연결은 소켓과 스레드를 이용하였다. 이를 통해서 server에는 여러 명의 client가 동시에 접근할 수 있도록 만들었다. 또한 하나의 사물함에는 한 명만 접근이 가능하도록 만들어서, 다른 client가 들어와도 해당 사물함에 접근할 수 없도록 하였다.

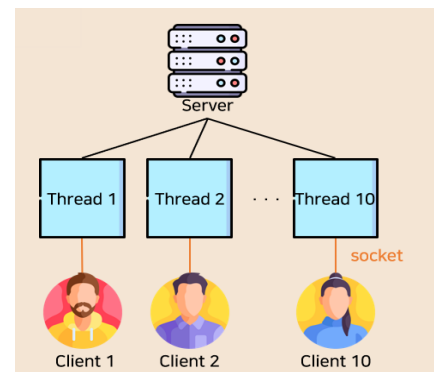
사물함의 정보는 locker 구조체를 이용해서 저장하였다. 사물함이 사용 중 인지, client가 해당 사물함에 접속 중인지, 비밀번호는 무엇인지 등의 정보를 기록한다. 이러한 사물함의 정보는 server에만 저장되고, client에서는 server에서 받은 정보들로 프로그램이 동작한다. 또한 client에서 입력된 정보들은 실시간으로 server에 저장된다.

server에서는 실시간으로 현재 사물함의 상태를 한 눈에 볼 수 있다. client는 접속 시에 현재 사물함의 상태를 보고 어떤 동작을 수행할 지 직접 선택할 수 있다. 또한 client가 사물함에 비밀번호를 설정하고, 다시 해제하는 과정에 여러 기능을 추가하였다.

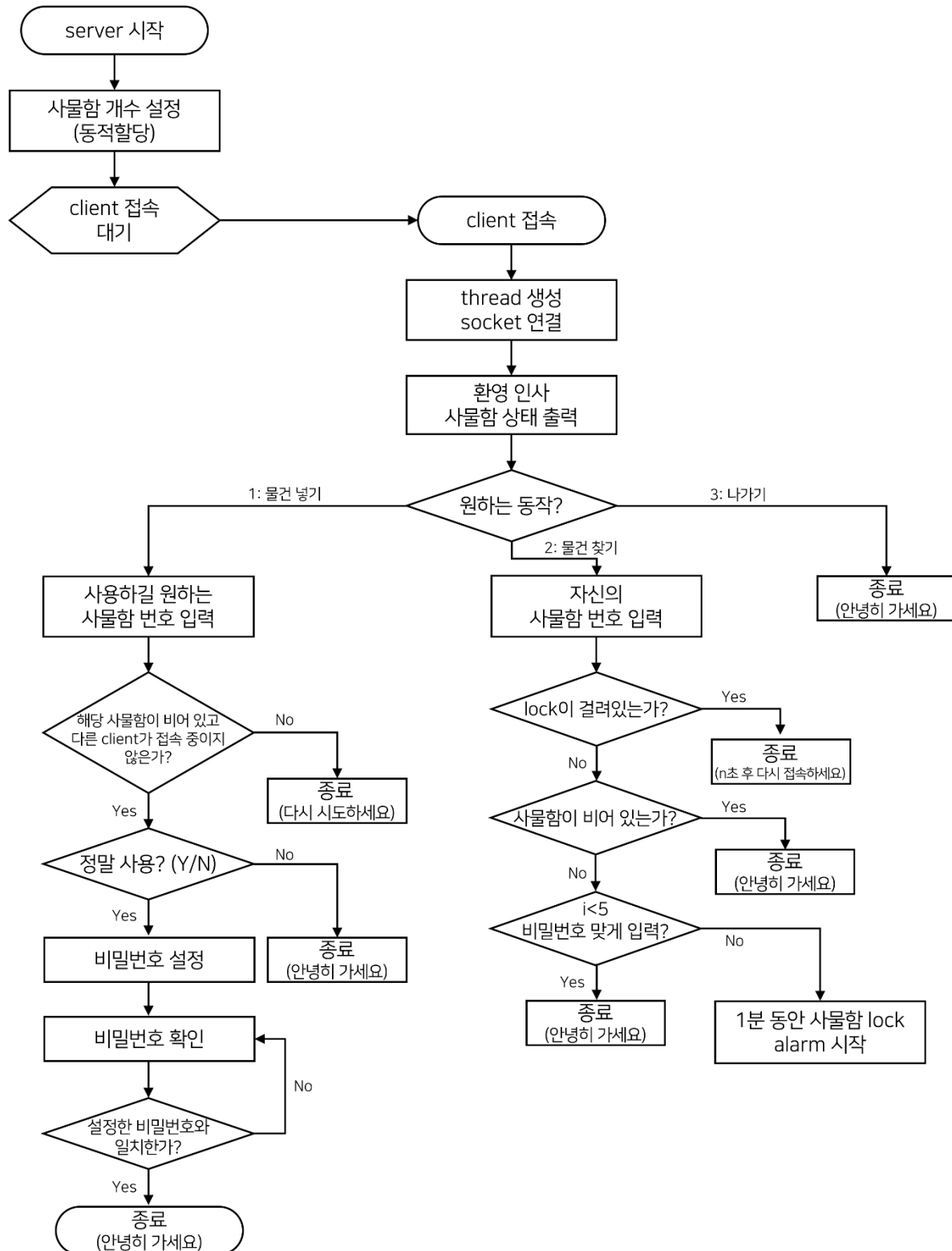
1.2 프로젝트 설명

수업시간에 배운 시스템 호출 (system call)들을 사용하여 사물함 관리 시스템을 구현하는 것이다.

Server와 Client가 통신을 할 수 있도록 소켓을 사용하였고 여러 개(명)의 Client를 처리하기 위해 스레드를 사용하였다.

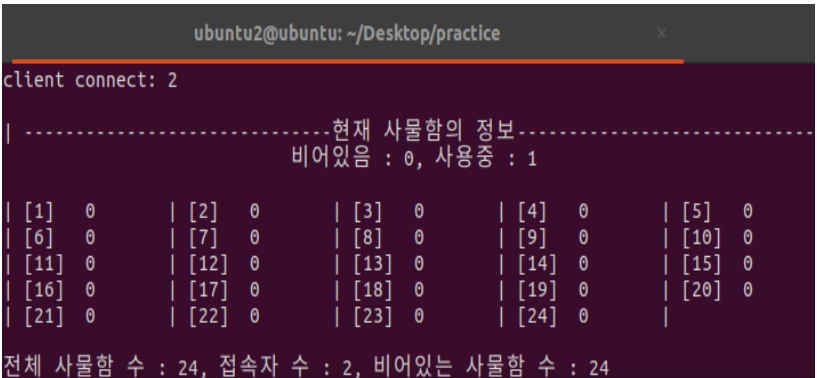
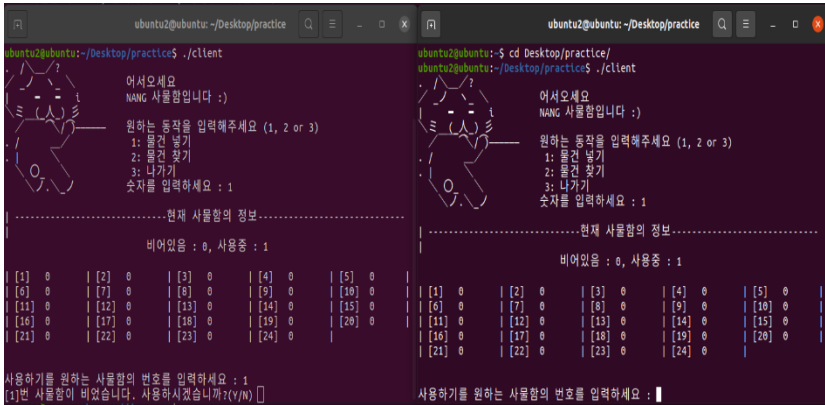


1.3 flowchart



2. 본론

2.1 프로젝트 장점

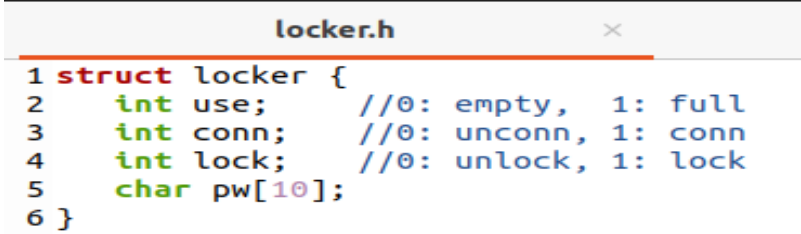
기능	다중 접속 및 사물함 정보 실시간으로 반영
기능설명, 내용	스레드와 소켓 사용
장점	멀티 스레드는 멀티 프로세스보다 적은 메모리 공간을 차지하고 운영체제가 시스템 자원을 효율적으로 관리 할 수 있다. 사물함의 정보가 실시간으로 반영되어 client가 접속하면, server에서는 소켓과 스레드를 이용하여 사물함의 정보를 알려준다.
server 결과 이미지	 <pre> ubuntu2@ubuntu: ~/Desktop/practice client connect: 2 -----현재 사물함의 정보----- 비어있음 : 0, 사용중 : 1 [1] 0 [2] 0 [3] 0 [4] 0 [5] 0 [6] 0 [7] 0 [8] 0 [9] 0 [10] 0 [11] 0 [12] 0 [13] 0 [14] 0 [15] 0 [16] 0 [17] 0 [18] 0 [19] 0 [20] 0 [21] 0 [22] 0 [23] 0 [24] 0 전체 사물함 수 : 24, 접속자 수 : 2, 비어있는 사물함 수 : 24 </pre>
client 결과 이미지	 <pre> ubuntu2@ubuntu: ~/Desktop/practice ubuntu2@ubuntu:~/Desktop/practice\$./client 어서오세요 NANG 사물함입니다 :) 원하는 동작을 입력해주세요 (1, 2 or 3) 1: 물건 넣기 2: 물건 찾기 3: 나가기 숫자를 입력하세요 : 1 -----현재 사물함의 정보----- 비어있음 : 0, 사용중 : 1 [1] 0 [2] 0 [3] 0 [4] 0 [5] 0 [6] 0 [7] 0 [8] 0 [9] 0 [10] 0 [11] 0 [12] 0 [13] 0 [14] 0 [15] 0 [16] 0 [17] 0 [18] 0 [19] 0 [20] 0 [21] 0 [22] 0 [23] 0 [24] 0 사용하기를 원하는 사물함의 번호를 입력하세요 : 1 [1]번 사물함이 비었습니다. 사용하시겠습니까?(Y/N) ubuntu2@ubuntu:~/Desktop/practice\$ cd Desktop/practice/ ubuntu2@ubuntu:~/Desktop/practice\$./client 어서오세요 NANG 사물함입니다 :) 원하는 동작을 입력해주세요 (1, 2 or 3) 1: 물건 넣기 2: 물건 찾기 3: 나가기 숫자를 입력하세요 : 1 -----현재 사물함의 정보----- 비어있음 : 0, 사용중 : 1 [1] 0 [2] 0 [3] 0 [4] 0 [5] 0 [6] 0 [7] 0 [8] 0 [9] 0 [10] 0 [11] 0 [12] 0 [13] 0 [14] 0 [15] 0 [16] 0 [17] 0 [18] 0 [19] 0 [20] 0 [21] 0 [22] 0 [23] 0 [24] 0 사용하기를 원하는 사물함의 번호를 입력하세요 : </pre>

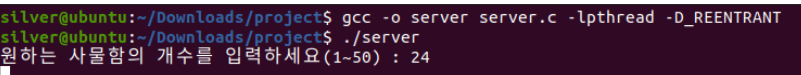
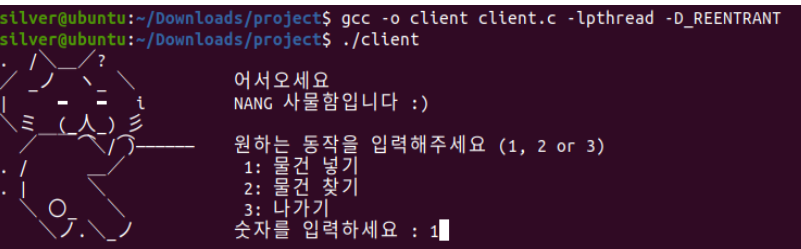
기능	비밀번호 설정 시 확인
기능설명, 내용	물건을 넣기 위해 비밀번호를 설정할 때에는 처음 입력한 비밀번호와 일치할 때까지 입력을 받는다. 두 비밀번호가 일치할 때 사물함에 물건을 넣는 것이 완료된 것이다.
장점	비밀번호를 설정할 때 제대로 입력되었는지 확인할 수 있다.
server 결과 이미지	server는 clear 상태로 현재는 보이지 않는다.

client 결과 이미지	<pre> 사용하기를 원하는 사물함의 번호를 입력하세요 : 2 [2]번 사물함이 비었습니다. 사용하시겠습니까?(Y/N) Y 설정할 비밀번호를 입력해주세요 (영어, 숫자 모두 입력 가능): silver222 다시 입력해주세요: silver222 감사합니다. 안녕히가세요 :) </pre>
---------------	---

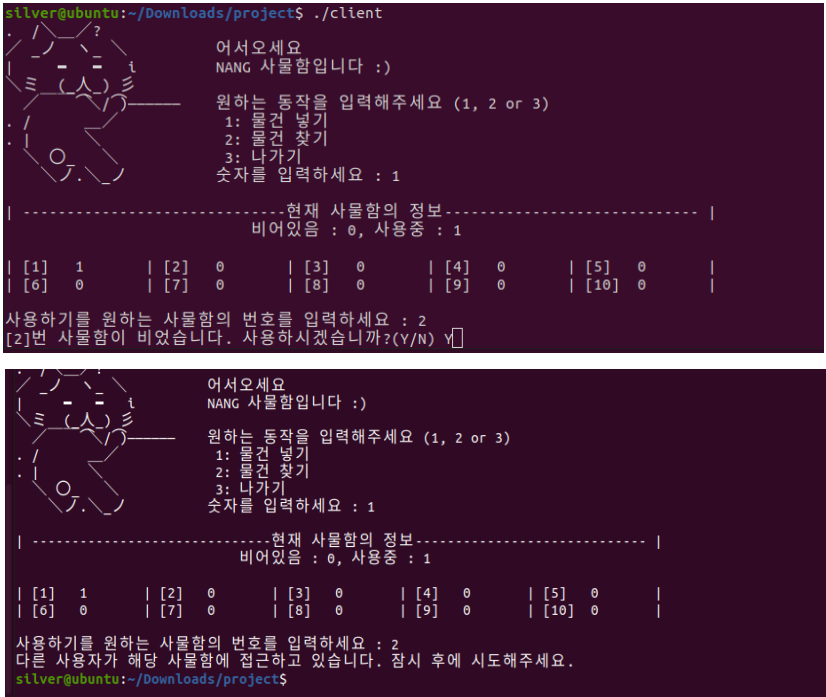
기능	비밀번호 입력 오류 잠금
기능설명, 내용	<p>물건을 찾을 때에는 client가 비밀번호는 최대 5번 입력할 수 있는데 비밀번호가 마지막까지 일치하지 않는다면 1분 동안 해당 사물함이 잠기고 프로그램이 종료된다. client들이 해당 사물함에 다시 접근하려고하면, 남은 시간이 출력된다.</p>
장점	<p>사물함이 1분간 잠겨도 client가 직접 시간을 쟁 필요 없이 알아서 시간을 계산하여 주고 남은 시간을 확인도 가능하다. 해당 사물함의 주인이 아닌 사람이 와서 접근하는 것을 막을 수 있다.</p>
server 결과 이미지	<pre> -----현재 사물함의 정보----- 비어있음 : 0, 사용중 : 1 [1] 0 [2] 1 [3] 0 [4] 0 [5] 0 [6] 0 [7] 0 [8] 1 [9] 1 [10] 0 [11] 1 [12] 0 [13] 0 [14] 0 [15] 0 [16] 0 [17] 0 [18] 0 [19] 1 [20] 0 [21] 0 [22] 1 [23] 0 [24] 0 </pre>
client 결과 이미지	<pre> 사물함 번호를 입력하세요 : 2 [1번째 시도] 비밀번호를 입력하세요 : 1 [2번째 시도] 비밀번호를 다시 입력하세요 : 12 [3번째 시도] 비밀번호를 다시 입력하세요 : 123 [4번째 시도] 비밀번호를 다시 입력하세요 : 1234 [5번째 시도] 비밀번호를 다시 입력하세요 : 12345 잘못된 비밀번호입니다. 1분동안 프로그램이 잠깁니다. -----현재 사물함의 정보----- 비어있음 : 0, 사용중 : 1 [1] 0 [2] 1 [3] 0 [4] 0 [5] 0 [6] 0 [7] 0 [8] 1 [9] 1 [10] 0 [11] 1 [12] 0 [13] 0 [14] 0 [15] 0 [16] 0 [17] 0 [18] 0 [19] 1 [20] 0 [21] 0 [22] 1 [23] 0 [24] 0 </pre> <p>사물함 번호를 입력하세요 : 2</p> <p>사물함이 잠겼습니다. 18초 뒤에 다시 시도해주세요.</p>

기능	사물함 정보 저장
기능설명, 내용	<p>Struct 사용</p> <p>Struct에 use, conn, lock, pw 라는 변수를 사용하였다.</p> <p>use: 해당 사물함이 사용 중인지 확인</p>

	conn: 해당 사물함에 다른 client가 접근하는지 확인 lock: 해당 사물함이 잠겨있는지 확인 pw: 사물함 비밀번호
장점	conn은 해당 사물함에 다른 client가 접근하는지를 확인하기 위해 1일 때 연결 중으로 반영하였다. client가 사물함 번호를 지정하여 접근하고 있을 경우 다른 client가 접속하여 해당 사물함에 접근하려고 하여도 접근할 수 없게 하였다. lock은 1일 때 잠겼다고 정보를 저장하여 해당 사물함이 비밀번호 입력 오류로 인해 잠겨 있는지를 확인할 때 사용하였다. 이를 통하여 사물함의 정보가 변경될 때 반영에 도움을 주었다.
결과 이미지	 <pre> locker.h 1 struct locker { 2 int use; //0: empty, 1: full 3 int conn; //0: unconn, 1: conn 4 int lock; //0: unlock, 1: lock 5 char pw[10]; 6 } </pre>

기능	사물함 개수 설정
기능설명, 내용	처음 server가 시작되면, 동적할당을 통해 사물함의 개수를 최대 50개까지 매번 다르게 설정 가능하다.
장점	server가 client가 많을 것 같은 날은 사물함을 많이 설정하여 만들 수 있다.
server 결과 이미지	 <pre> silver@ubuntu:~/Downloads/project\$ gcc -o server server.c -lpthread -D_REENTRANT silver@ubuntu:~/Downloads/project\$./server 원하는 사물함의 개수를 입력하세요(1~50) : 24 </pre>
client 결과 이미지	 <pre> silver@ubuntu:~/Downloads/project\$ gcc -o client client.c -lpthread -D_REENTRANT silver@ubuntu:~/Downloads/project\$./client ? (人) 어서오세요 NANG 사물함입니다 :) 원하는 동작을 입력해주세요 (1, 2 or 3) 1: 물건 넣기 2: 물건 찾기 3: 나가기 숫자를 입력하세요 : 1 </pre>

기능	한 사물함에 여러 사용자 접근 불가
기능설명, 내용	한 client가 사물함을 이용하고 있으면, 다른 사용자가 해당 사물함을 이용하지 못한다.
장점	여러 client가 한 사물함에 동시에 접근할 때 정보가 중복되는 것을 막는다.

server 결과 이미지	server는 clear 상태로 현재는 보이지 않는다.
client 결과 이미지	 <pre> silver@ubuntu:~/Downloads/project\$./client 어서오세요 NANG 사물함입니다 :) 원하는 동작을 입력해주세요 (1, 2 or 3) 1: 물건 넣기 2: 물건 찾기 3: 나가기 숫자를 입력하세요 : 1 -----현재 사물함의 정보----- 비어있음 : 0, 사용중 : 1 [1] 1 [2] 0 [3] 0 [4] 0 [5] 0 [6] 0 [7] 0 [8] 0 [9] 0 [10] 0 사용하기를 원하는 사물함의 번호를 입력하세요 : 2 [2]번 사물함이 비었습니다. 사용하시겠습니까?(Y/N) Y 어서오세요 NANG 사물함입니다 :) 원하는 동작을 입력해주세요 (1, 2 or 3) 1: 물건 넣기 2: 물건 찾기 3: 나가기 숫자를 입력하세요 : 1 -----현재 사물함의 정보----- 비어있음 : 0, 사용중 : 1 [1] 1 [2] 0 [3] 0 [4] 0 [5] 0 [6] 0 [7] 0 [8] 0 [9] 0 [10] 0 사용하기를 원하는 사물함의 번호를 입력하세요 : 2 다른 사용자가 해당 사물함에 접근하고 있습니다. 잠시 후에 시도해주세요. silver@ubuntu:~/Downloads/project\$ </pre>

2.2 프로젝트 내용

처음 server가 시작되면, 동적 할당을 통해 사물함의 개수를 매번 다르게 최대 50개까지 설정이 가능하다. client가 시작되면 server에서 스레드를 만들어 소켓으로 연결한다. 멀티 스레드를 이용하여 다중 접속이 가능하게 만들었다. 사물함의 정보가 실시간으로 반영되어 client가 접속하면, server에서는 사물함의 정보를 알려준다.

client에서 동작 번호를 입력 받으면, server에서도 현재 사물함의 정보와 전체 사물함의 수, 접속자 수, 비어있는 사물함의 수가 출력된다. client에서 동작 번호로 '1'을 입력하면 사물함에 물건을 넣는 과정을 거친다. 이때 현재 사물함의 정보를 보여주고 사용하기를 원하는 사물함의 번호를 입력 받는다.

client에서 사물함의 번호를 입력 받는다. 이때 해당 사물함이 비어 있다면, 정말 사용할 지 여부를 묻는다. client가 'Y'라고 입력하면 원하는 비밀번호를 설정할 수 있도록 한다. 비밀번호를 처음 입력 후 다시 확인하는 과정을 거치는 데 처음 입력했던 비밀번호와 일치할 때까지 다시 입력을 받아 처음 입력한 비밀번호와 일치했을 때 사물함을 사용할 수 있도록 하고 프로그램이 종료된다. client가 'N'이라고 입력하면 프로그램이 바로 종료된다.

client에서 동작번호 '2'를 입력하면 사물함에 물건을 찾는 과정이 일어난다. client가 물건을 찾다고 하면 현재 사물함의 정보를 보여주고 원하는 사물함의 번호를 입력한다. server에서는 client가 동

작번호를 입력하면 현재 사물함의 정보와 전체 사물함의 수, 접속자 수, 비어있는 사물함의 수가 출력된다.

client가 물건을 찾을 때 지정 사물함에 대한 비밀번호를 입력받는다.

client가 물건을 찾을 때는 비밀번호는 최대 5번 입력할 수 있다. 비밀번호를 제대로 입력하면, 감사 인사와 함께 해당 사물함은 비워지면서 프로그램이 종료된다. 만약 비밀번호를 5번 틀리면 해당 사물함이 1분동안 잠겨 접근할 수 없다.

server에서 정보를 한 눈에 확인할 수 있게 하기 위하여 **clear()**함수를 통해 화면을 지워주었다.

3. 결론

3.1 말은 일

이름	말은 일
김아은 (202012178)	<ul style="list-style-type: none">- locker 구조체 만들기, main 작성하기- client 동작 1(물건 넣기) 구현하기- 매 작업물 백업, 합치기- PPT 제작 및 발표하기, 보고서 검토
박나현 (202018392)	<ul style="list-style-type: none">- locker 구조체 만들기, main 작성하기- client 동작 2(물건 찾기) 구현하기- 디버깅 및 수정하기- 보고서 작성하기, PPT 검토

3.2 프로젝트 수행 후기

- 수업 때 배운 signal, socket 등을 가지고 사용해보니 실습 과제때보다 조금 더 큰 프로젝트를 만들어 볼 수 있어서 더욱 잘 이해할 수 있었다.
- 수업시간에 배운 내용 이외에도 thread를 공부한 뒤 적용하며 응용력을 기를 수 있었다.
- 시각화를 위해 nucurse 라이브러리를 사용해보고 싶었는데, 적용하지 못한 점이 아쉬웠다.
- 추가적으로 구현해보고 싶은 기능도 있었으나 시간이 부족해서 다 구현하지 못한 점이 아쉬웠다.