**유닉스 시스템 프로젝트 보고서**

**<사물함 관리 시스템>**

**15조**

**IT정보공학과 202012252 지수영**

**IT정보공학과 202012182 김채영**

**목 차**

**1. 프로젝트 주제**

**2. 기본 기능**

**2.1 서버**

**2.2 클라이언트**

**3. 추가 기능**

**3.1 동적 할당**

**3.2 접근 오류 시 사물함 잠금 기능**

**3.3 메모 저장 기능**

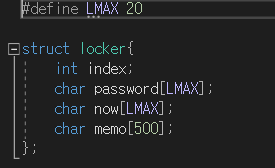
**3.4 비밀번호 길이 자유 설정 기능**

**4. 역할 분담**

**1. 프로젝트 주제**

시스템 호출(system call)을 사용하여 서버-클라이언트(server-client) 모델로 동작하는 사물함 관리 시스템을 구현하였다. 서버는 사물함을 관리하는 역할을, 클라이언트는 사물함 관리 시스템의 고객 역할을 수행한다.

**2. 기본 기능**

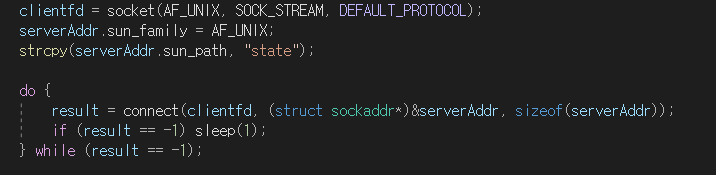
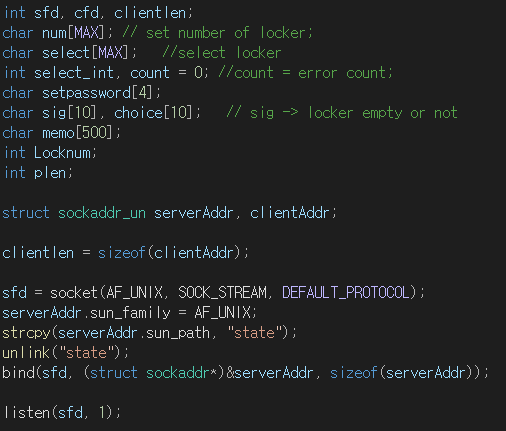
 **<헤더파일<clocker.h>**

index => 사물함 번호 / password[LMAX] => 비밀번호

/ now[LMAX] => 현재 상태 / memo[500] => 메모

struct 키워드를 사용하여 locker라는 이름의 구조체를 정의하고 index, password, now, memo 4개의 멤버 변수를 선언해주었다. index 변수는 사물함의 번호를 나타내고 password 변수는 비밀번호를 나타낸다. now는 현재 사물함이 비어있는 상태인지 사용 중인 상태인지를 나타내기 위해 사용하고 memo는 사용자가 사물함 이용 시 저장할 메모 내용을 받기 위해 사용한다.

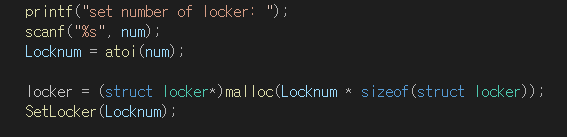
<소켓 연결>



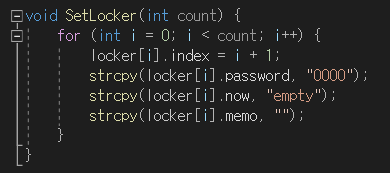
서버가 먼저 소켓을 만든 후 클라이언트가 소켓을 만들고 서버에 연결 요청을 한다. 서버가 클라이언트의 연결 요청을 수락(accept)하면 소켓 연결이 이루어진다.

**2.1 서버**

사용자에게 사물함에 대한 정보를 제공한다.

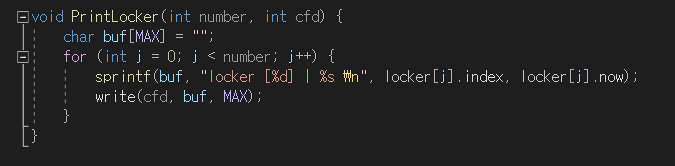


* 프로그램을 실행시키면 사물함의 개수를 입력받는다..

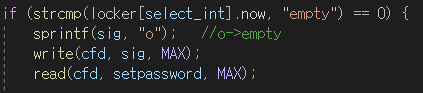


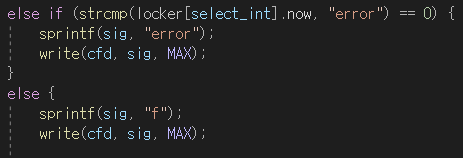
* 초기 사물함의 정보를 비밀번호는 0000, 사물함 상태는 empty, 사물함의 memo는 비어있는 상태로 설정한다.





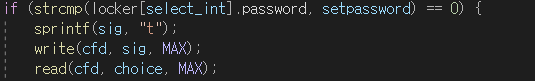
* write 함수를 통해 사물함의 현재 상태를 사물함 한 개씩 클라이언트에게 제공한다.
* 사용자에게 사물함 할당 시 사용자가 패스워드를 지정하도록 한다.





클라이언트로부터 사용자가 사용할 사물함 번호를 입력 받으면 사물함의 상태를 확인하고 상태에 따라 비어있는 경우에는 “o”, 추가기능에서 설명할 “error” 그리고 사물함을 사용 중일 경우 “f” 라는 시그널을 클라이언트에 전달한다. 신호를 전달받은 클라이언트는 비밀번호를 입력해서 다시 서버에 전달한다.

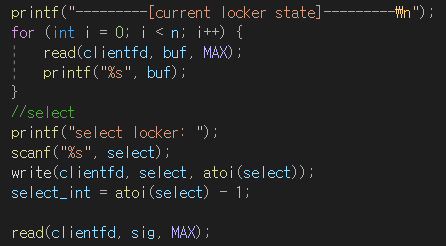
* 사용자가 본인 사물함에 접근 시 입력한 패스워드가 초기 설정한 패스워드와 동일한지 확인 후 맞을 경우 접근 가능하도록 한다.



만약 사용자가 본인의 사물함에 접근할 경우 비밀번호가 일치하는지 여부를 먼저 체크한다. 비밀번호가 일치하는 경우 “t”라는 시그널이 클라이언트로 전달되고 클라이언트가 시그널을 전달받았을 경우 사물함의 잠금을 해제할 것인지에 대한 대답이 choice에 저장 되어 다시 서버로 전달된다. 만약 대답(Y/N)이 Y(Yes)일 경우 사물함의 잠금은 해제되고 초기 상태로 재설정된다. 반대로 N(No)일 경우 사물함의 비밀번호는 유지되고 추가로 메모를 작성하라는 메시지가 나오고 사용자는 메모에 원하는 내용을 적어 저장할 수 있다.

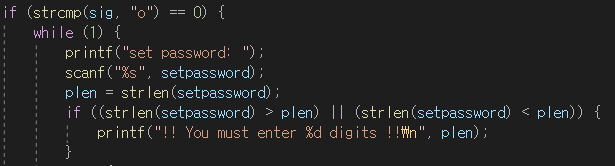
**2.2 클라이언트**

원하는 사물함에 접근하여 사물함을 사용한다.

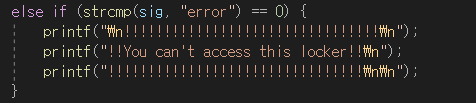


서버로부터 사물함의 현재 상태를 전달받고 출력한 뒤 어떤 사물함에 접근할 것인지에 대해 입력한다. 입력된 값(사용하고자 하는 사물함 번호)은 다시 서버로 전달되고 사물함의 상태에 따라 시그널을 받는다.

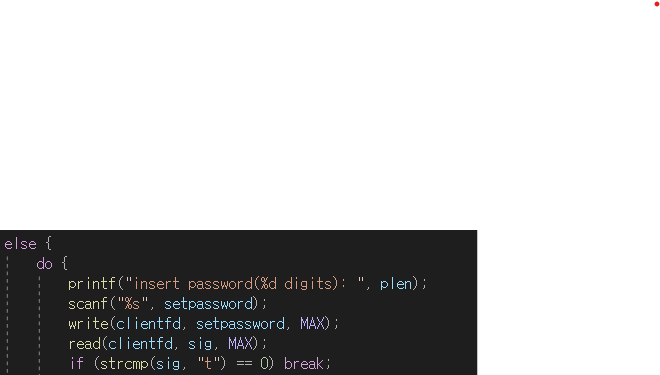
* 해당 사물함의 비밀번호 설정 및 입력 기능



서버의 기능에서 설명했듯이 시그널이 “o”일 경우(사물함이 비어있음) 비밀번호를 입력할 수 있다. 비밀번호의 길이를 저장하기 위해 변수 “plen”을 사용한다.



사물함의 상태가 error일 경우 접근할 수 없다는 메시지가 뜬다.



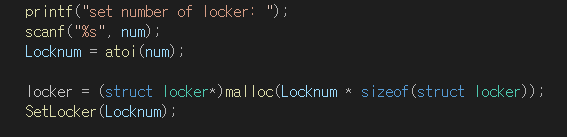
만약 입력한 비밀번호가 저장된 비밀번호와 다르다면 다시 입력하라는 오류 메시지가 출력된다.



비밀번호가 같을 경우 사물함의 잠금을 해제할 것인지에 대한 대답을 입력하고 서버로 전달한다.

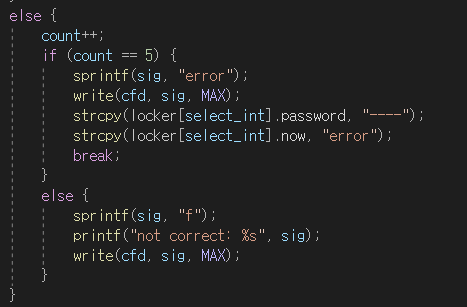
**3. 추가 기능**

**3.1 동적 할당**



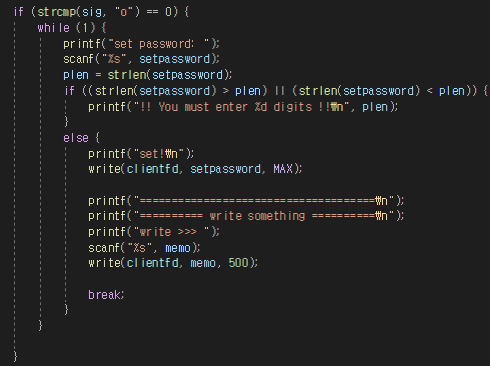
* 서버를 열 때, **동적 할당**을 통하여 사용 가능한 사물함의 수를 매번 다르게 설정할 수 있도록 프로그램을 구현하였다. 따라서 서버 프로그램을 실행시키면 사물함의 개수를 입력받아 서버관리자가 사물함의 수를 설정할 수 있도록 하였다.

**3.2 접근 오류 시 사물함 잠금 기능**



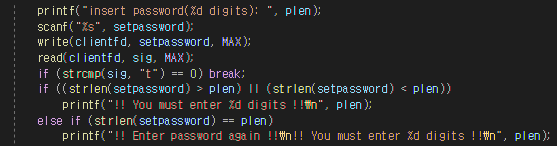
* 사용자가 사물함의 비밀번호 및 사용 권한을 해제할 수 있도록 하였다. 만약 사용자가 잠겨있는 사물함에 접근했을 때 비밀번호를 5회 이상 틀리면 그 사물함의 상태는 “error”로 바뀌고 다시 접근이 불가능하다. 만약 에러가 발생한 사물함에 접근하면 에러코드가 발생한다.

**3.3 메모 저장 기능**



* 구조체 struct locker 설정 시에 memo 변수를 추가하였다. 사용자가 비어있는 사물함을 선택하고 비밀번호까지 설정한 후에 사용자가 사물함에 메모를 저장할 수 있도록 기능을 추가하였다. 사용자가 사용자의 사물함에 재접근 시에 매모를 재입력 받아 메모 내용을 변경할 수 있도록 기능을 구현하였다.

**3.4 비밀번호 길이 설정 기능**



* 사물함 사용자가 비밀번호 설정 시에 숫자/문자 자유롭게 선택할 수 있도록 하기 위해 문자열 char형으로 선언해주었다. 또한 비밀번호 길이도 사용자 뜻대로 설정할 수 있다. 설정한 비밀번호의 길이를 plen 변수에 저장하고, 이 후 사용자가 사물함에 재접근시에 비밀번호 입력을 틀릴 경우, 사용자가 설정했던 비밀번호의 길이를 알려준다.

**4. 역할 분담**

김채영 : 서버, 클라이언트 코드 작성, 추가 기능(비밀번호 길이), 발표

지수영 : 서버, 클라이언트 코드 작성, 추가 기능(메모, 에러코드 출력)