|  |
| --- |
| HW9. File IO |

1. **구현 내용에 대한 설명 (60점)**
2. 주요 변수 설명

next: 자기 자신과 같은 구조체를 가리키는 포인터

List : 글로벌 자기참조 구조체

argc: 처음에 주어진 인자의 개수

argv: 인자들이 저장되는 배열

fp: 파일의 포인터( 이용할 파일을 가리키게 한다. )

phone.c

1. 주요 자료 구조 설명

* 자기 참조 구조체를 이용한 연결 리스트를 활용한다.
* 모든 함수가 그 리스트를 간접 참조한다. ↓
* 프로그램이 phone.dat파일을 참조한다. ↓

phone.h

hw8Main

→→→→→→→→

↑ ↓ ↓ ↓ ↓

↑ ↓ ↓ ↓ ↓

list.c

sort.c

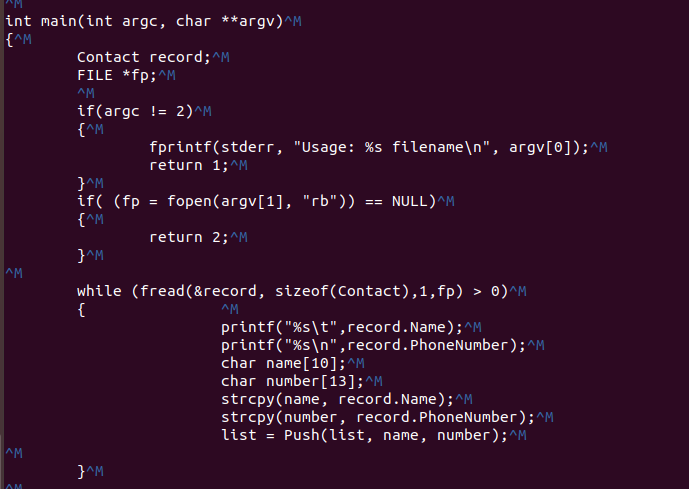
delete.c

search.c

register.c

1. 주요 함수 구현 방법 설명

파일 출력



우선 argc의 개수로 실행 명령어와 데이터 파일 둘다 입력해주었는지 확인한다.

없다면 오류라고 출력한다.

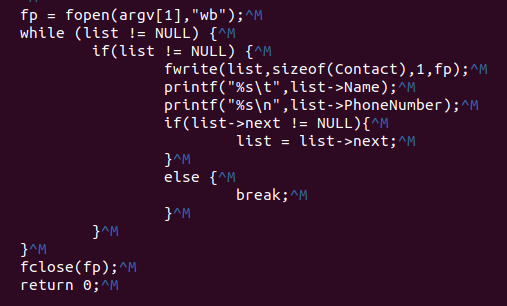
만약 둘다 입력했는데, 파일이 없을 경우엔 그냥 끝내버린다.

그리고 제대로 된 경우에는 파일을 바이너리 읽기모드로 연다음, 임시로 record구조체에

Contact구조체만큼 읽어온다. 그리고 그것을 register의 Push함수를 이용하여 메모리의 list에

추가 시킨다.

파일 입력



입력할 때 주어졌던 파일을 바이너리 쓰기모드로 연다. 그리고 list가 Null이 될 때까지

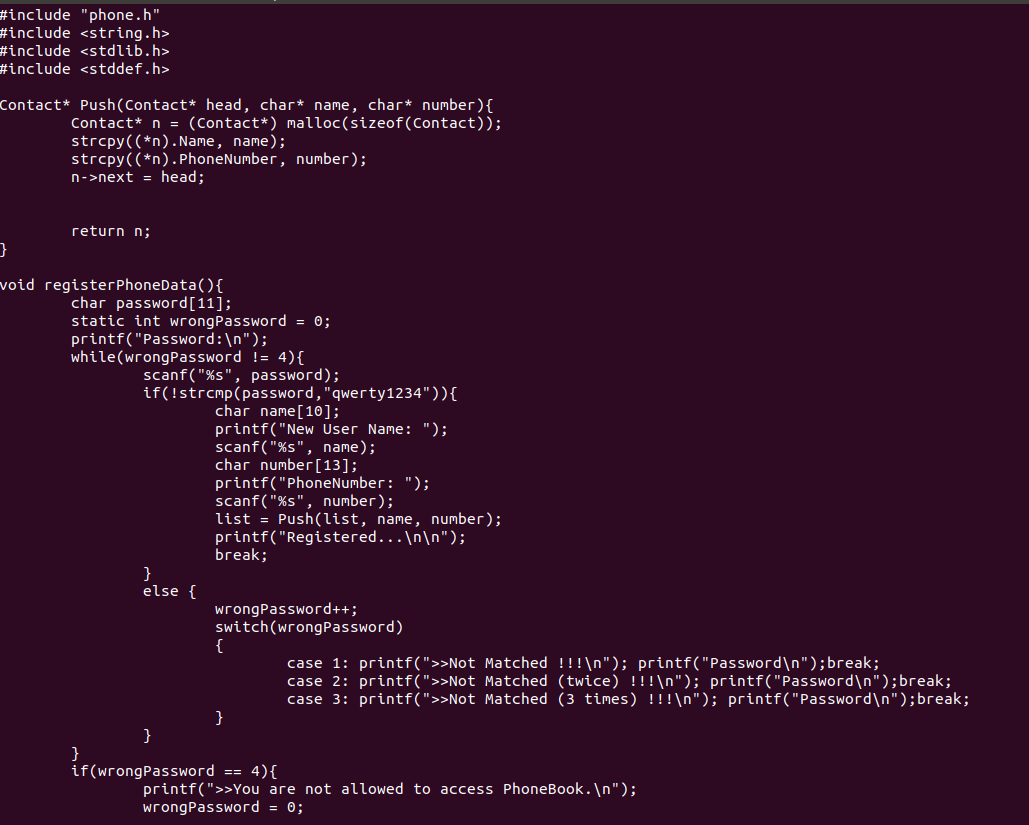
list의 내용을 파일에 Contact구조체 사이즈만큼 하나씩 입력시키고 list는 list의 next를

가리키게 한다.

list의 next가 null이 된다면 while문을 빠져나온다.

그리고 파일을 닫아 저장한다.

register



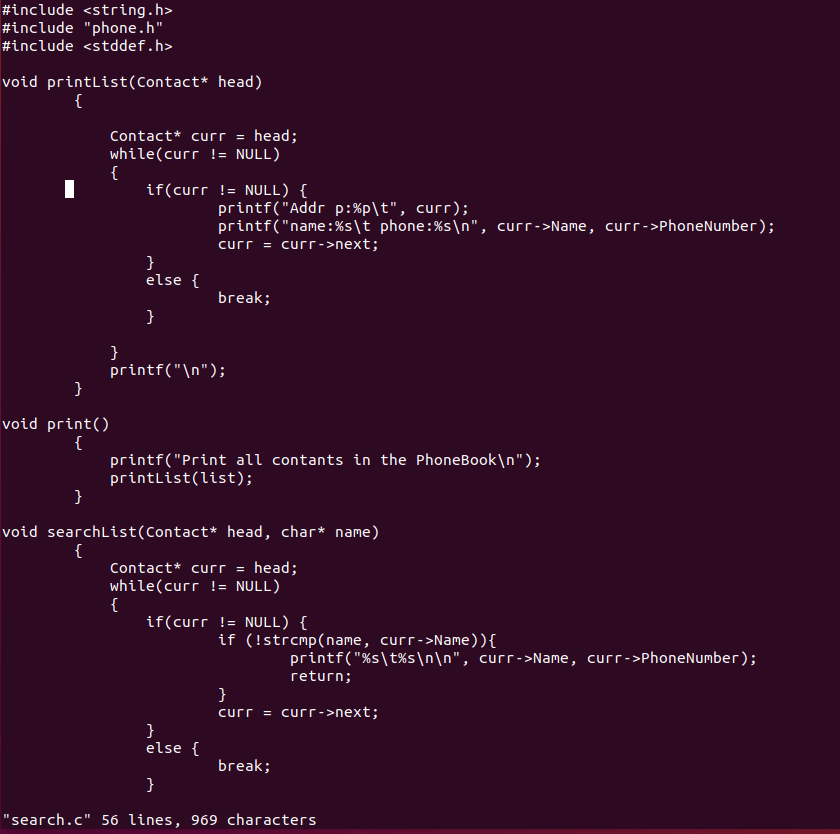
register할때 구조체를 새로 할당하여 그곳에 값을 집어넣는다.

그 후 그 구조체의 next가 list의 초기화 값인 NULL을 가리키게 한다.

이후에 새로 register되는 구조체들의 next는 전에 register했던 구조체를 가리키게 된다.

이런식으로 linked list가 만들어진다.

search.c의 print와 search



print

함수 내에서 list와 똑 같은 구조체를 만들어준다.

구조체의 주소, 이름과 번호를 출력한 뒤, list에 list의 next가 가리키는 구조체를 대입

while문으로 null을 가리키는 list가 나올때까지 출력한다.

search

함수 내에서 list와 똑 같은 구조체를 만들어준다.

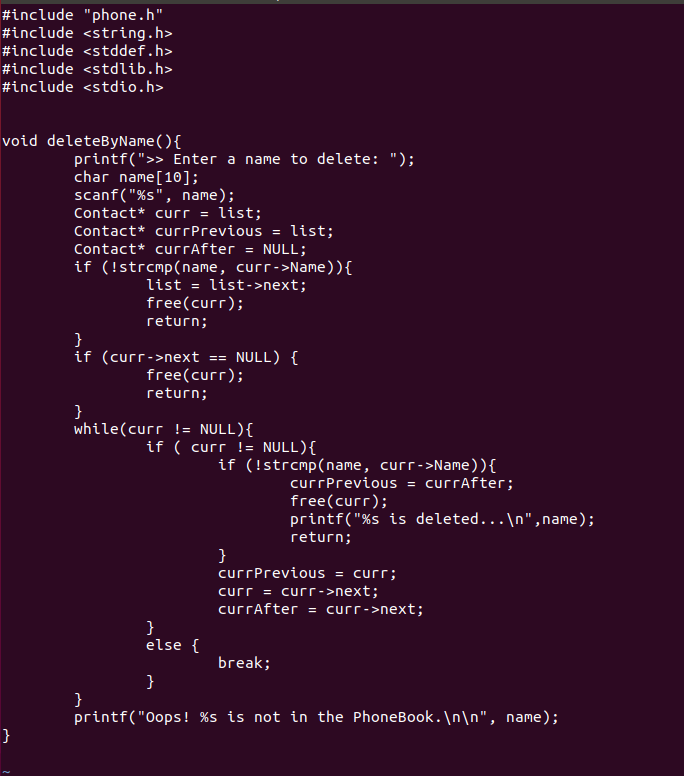
구조체의 이름과 입력받은 이름을 비교한 뒤, list에 list의 next가 가리키는 구조체를 대입

while문으로 두 이름이 같을때까지 비교한다.

만약 두 이름이 같을 경우엔 그 이름과 해당되는 번호를 출력한다.

입력받은 이름이 없을 경우엔 없다는 메시지를 출력한다.

delete



함수 내에서 list와 똑 같은 구조체를 만들어준다.

구조체의 이름과 입력받은 이름을 비교한 뒤, list에 list의 next가 가리키는 구조체를 대입

while문으로 두 이름이 같을때까지 비교한다.

만약 두 이름이 같을 경우엔 그 이름이 있는 구조체의 이전 구조체의 next가

이후 구조체를 가리키게 한뒤에, 해당 구조체는 free로 없애준다.

입력받은 이름이 없을 경우엔 없다는 메시지를 출력한다.

sort



함수 내에서 list와 똑 같은 구조체를 만들어준다.

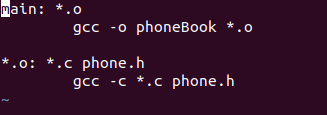
현재 구조체와 구조체의 next가 가리키는 구조체의 이름을 비교한다.

순서가 맞다면, list에 list의 next가 가리키는 구조체를 대입

순서가 반대면 해당 구조체와 이후 구조체의 내용을 뒤바꾼다.

while문으로 끝까지 총 2번 검사한다.

1. Makefile 설명



main으로 phoneBook을 명령어로 만들고, 해당되는 object파일들이 gcc로 컴파일 된다.

object파일들은 해당 .c 파일들과 phone.h로 구성되게끔 gcc -c로 실행된다.

1. **실행 방법 설명 (10점)**
2. 사용한 운영체제 및 컴파일러의 종류

우분투 리눅스, 터미널

1. 컴파일 방법 및 실행 방법

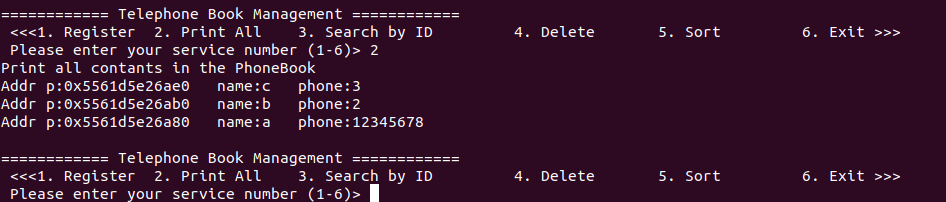
* make로 컴파일
* touch phone.dat으로 데이터 저장 파일을 만들어준다.
* ./phoneBook phone.dat으로 실행

(3) 동작을 확인할 수 있는 실행 화면 캡처

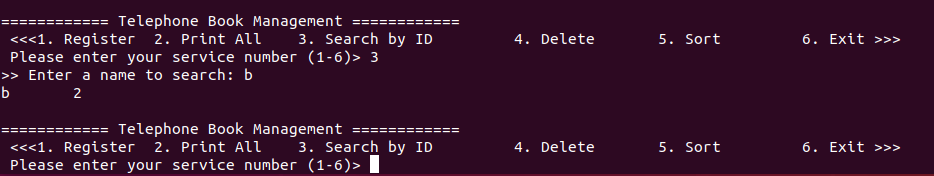
Register



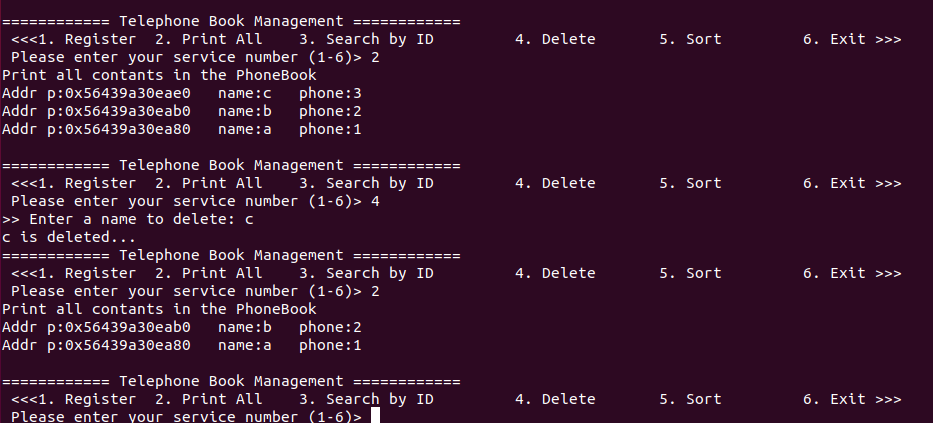
Print All



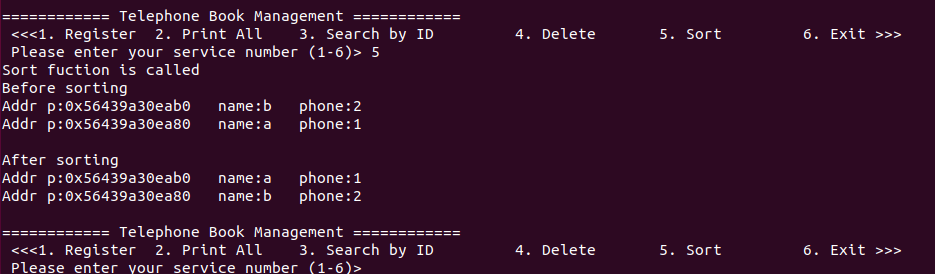
Search by ID



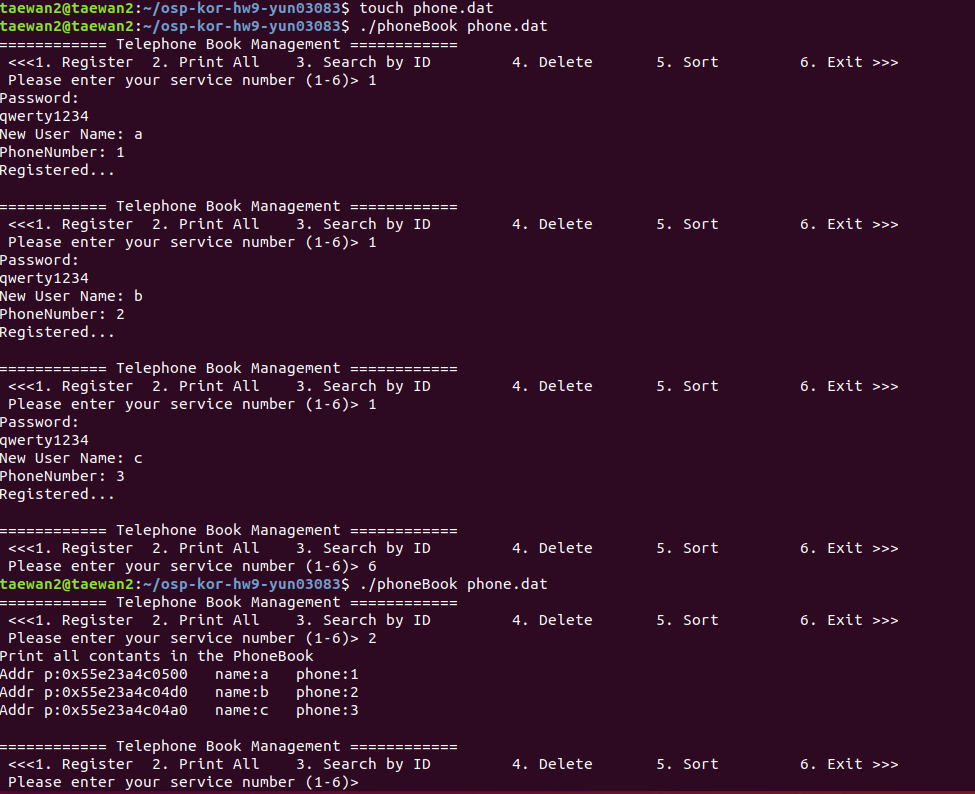
Delete



sort



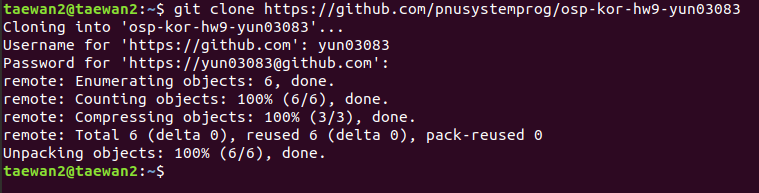
파일 입출력



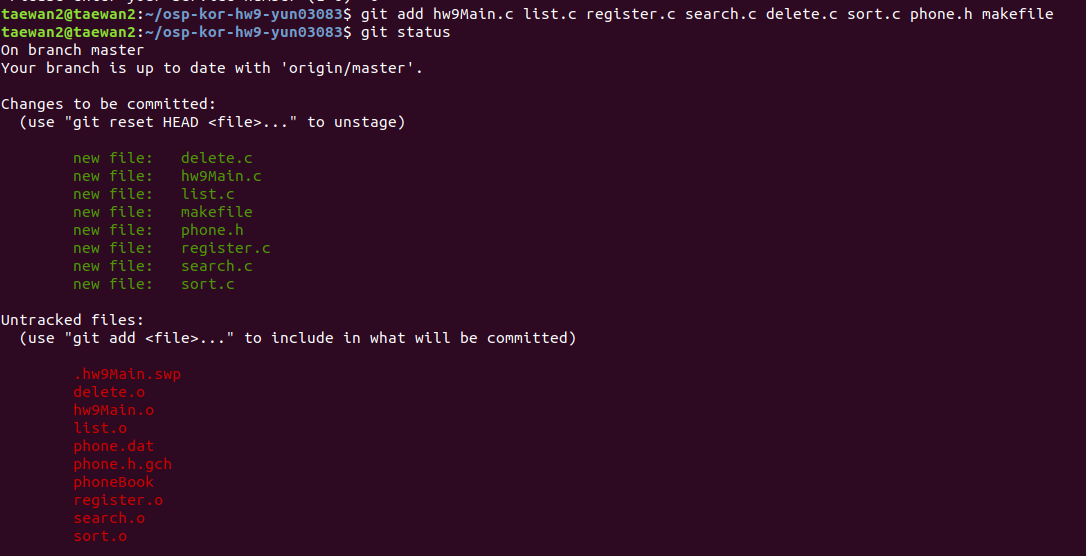
1. **Github 화면 (20점)**

(1) cloning, adding, committing, push을 위한 github 명령들을 포함

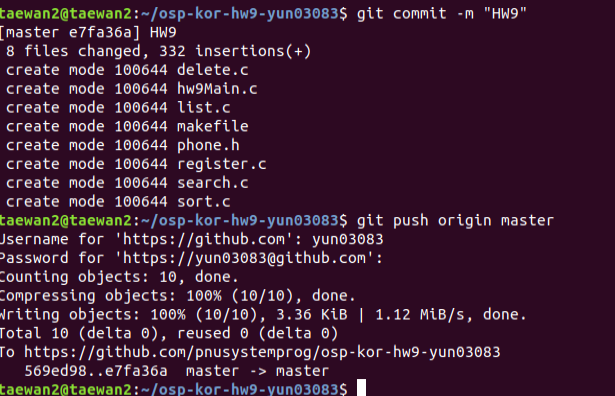
Cloning



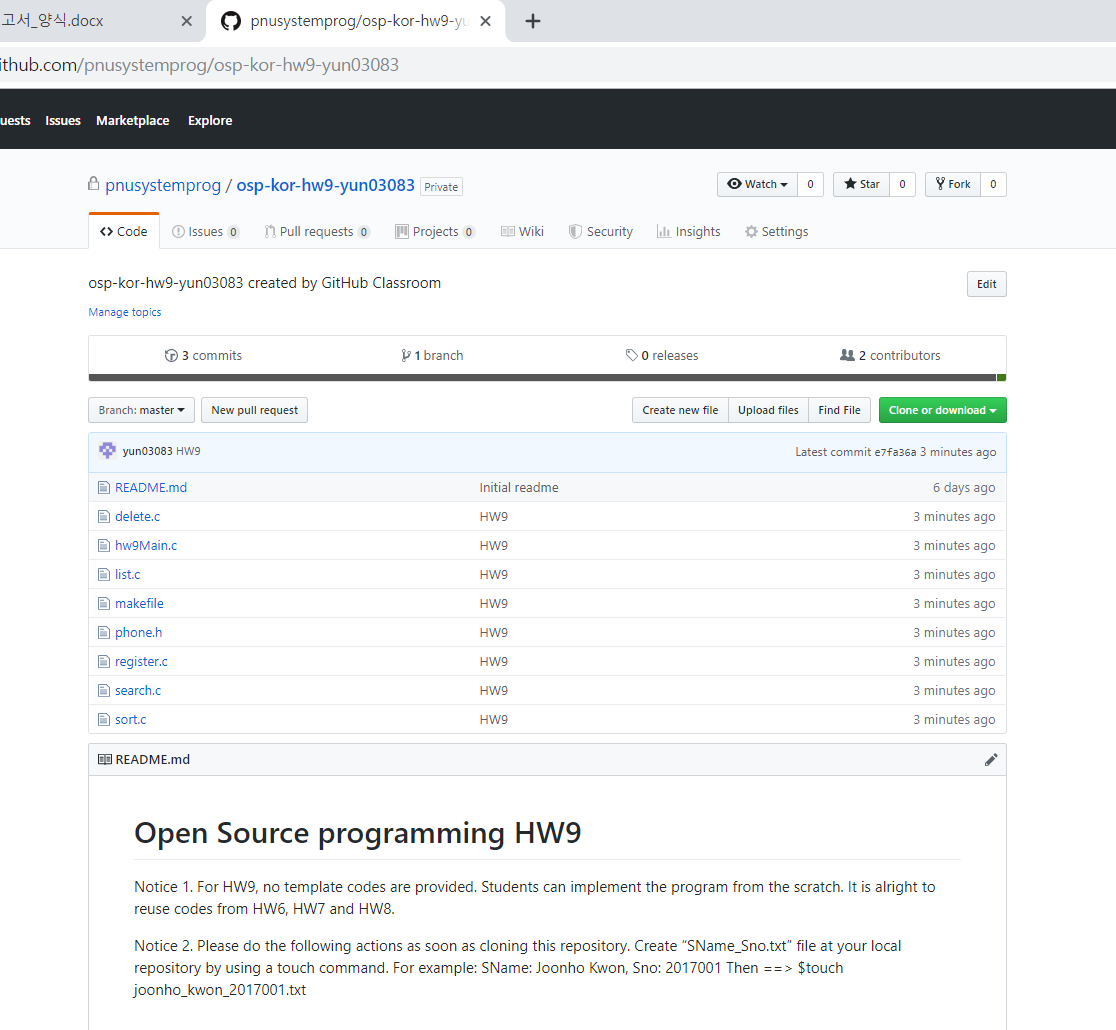
Adding



Committing, Push



(2) 소스 코드와 makefile을 push한 후, 본인의 Github repository를 스크린 캡쳐하여 포함



1. **논의 사항 (10점)**

- 숙제를 하면서 새로이 알게 된 내용(수업 시간 이외의 내용),

파일 포인터, stderr, fread와 fwrite의 매개변수들의 역할

- 숙제를 하는 중에 어려웠던 점 등을 기술

NULL을 참조하지 않게 while과 if로 조절하는 것

phone.dat에 저장이 제대로 되게 하는것과 제대로 읽어들이게

NULL을 참조하지 않게 조절하는것

Implict fuction오류 처리