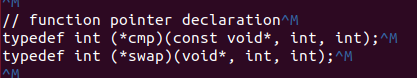
|  |
| --- |
| HW7. 함수 포인터와 void 포인터 |

# 구현 내용에 대한 설명 (60점)

1. 주요 변수 설명: 함수 포인터 설명



함수 이름은 주소이다. 즉 포인터가 함수를 가리킬 수 있다.

여기서 cmp는 int를 리턴하고 const void형 포인터와 int 2개를 매개변수로 갖는 함수를 가리킬 수 있는 포인터 이고, swap은 int를 리턴하고 void형 포인터와 int 2개를 매개변수로 갖는 함수를 가리킬 수 있는 포인터이다.

여기서 cmp는 contactCmpr(const void\* arr, int i, int j)와 contactSwap(void\* arr, int i, int j)를 가리키게 된다.

1. 주요 자료 구조 설명

hw7Main.c

phone.h

register.c

sort.c

delete.c

search.c

hw7Main에서 register.c, search.c, delete.c, sort.c의 함수들을 이용하고 register.c, search.c, delete.c, sort.c는 phone.h을 include하여 그곳의 변수들을 이용한다. 또한 sort.c는 hw7Main의 함수를 이용한다.

1. 주요 함수 구현 방법 설명

RegisterPhoneData()

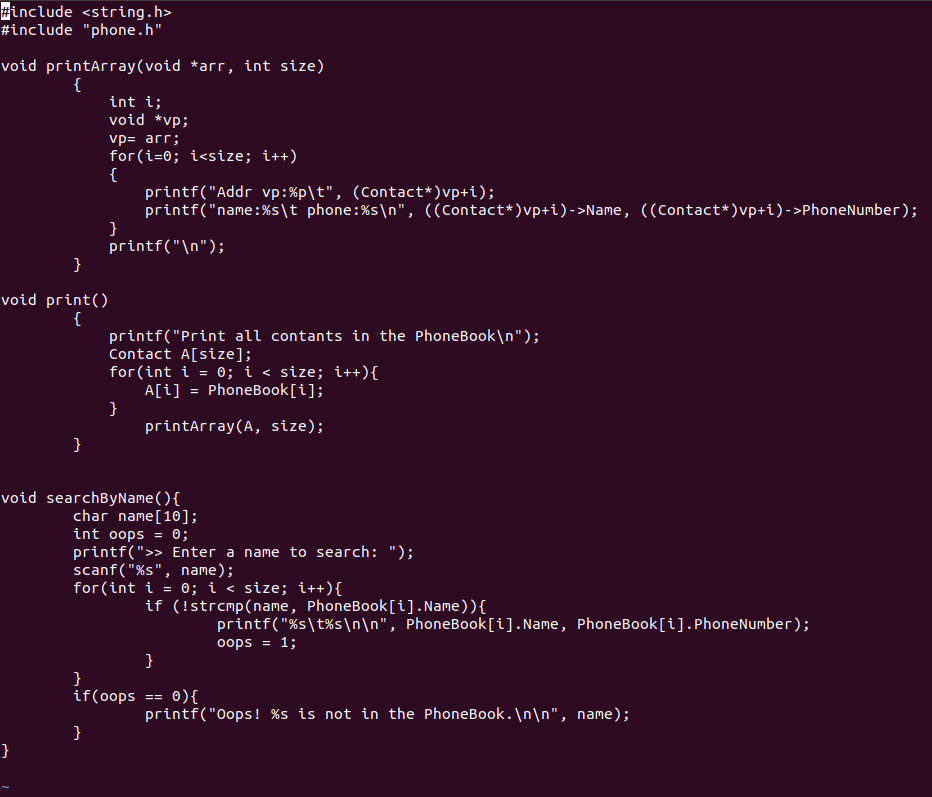


static변수인 bookSize변수로 값을 입력받을때마다 1씩 더하고 그 값을 size에 대입한다. (값은 유지된다.)

wrongPassword변수로 잘못된 비밀번호를 입력받을 때마다 1씩 더하고 switch로 그 값에 해당하는 출력값을 출력한다.

올바른 비밀번호를 입력받았을시에 phoneBook의 Name과 PhoneNumber에 이름과 번호를 입력받는다. 입력받을 때 bookSize변수를 이용하여 PhoneBook에 순서대로 차곡차곡 입력받는다.

search.c 의 printArray, print, searchByName



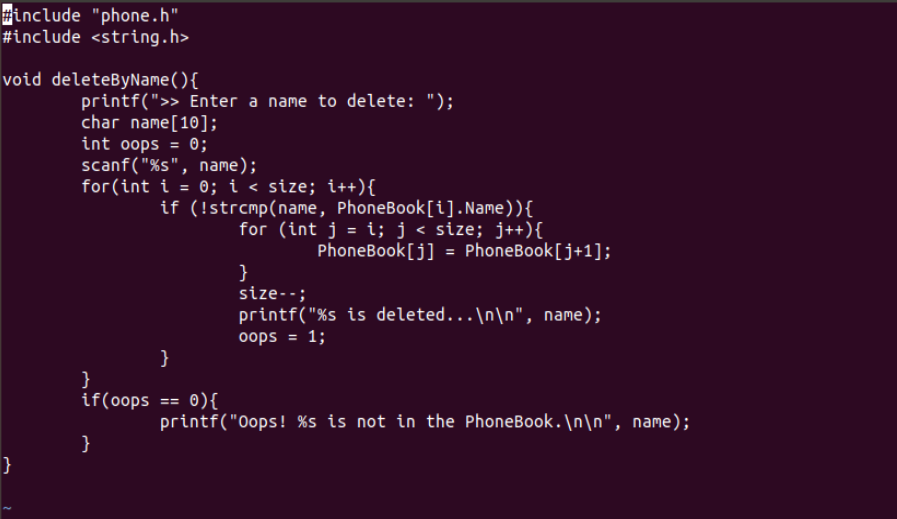
Print()는 PhoneBook의 내용을 그대로 복사하여 printArray()를 실행시킨다.

printArray()는 size와 void형 포인터 arr을 매개변수로 받은 뒤에, vp void 포인터변수를 이용하여 (Contact\*)로 지정해준 다음 포인터가 가리키는 구조체의 이름과 번호를 출력한다.

SearchByName()함수는 특정 이름을 입력받았을시에 PhoneBook을 뒤져서 같은 이름이 있을시에는 그 자리의 이름과 번호를 출력시킨다.

없을때에는 oops변수를 이용하여서 없다는 말을 출력시킨다.

delete.c의 deleteByName()함수

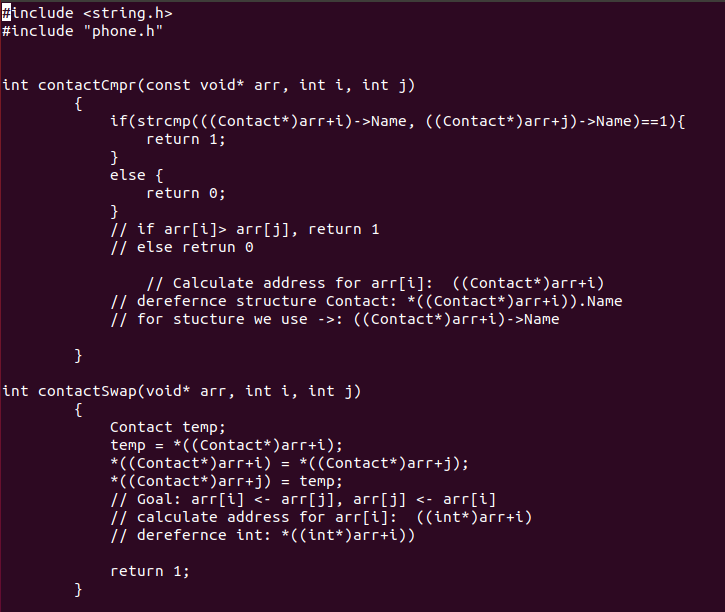


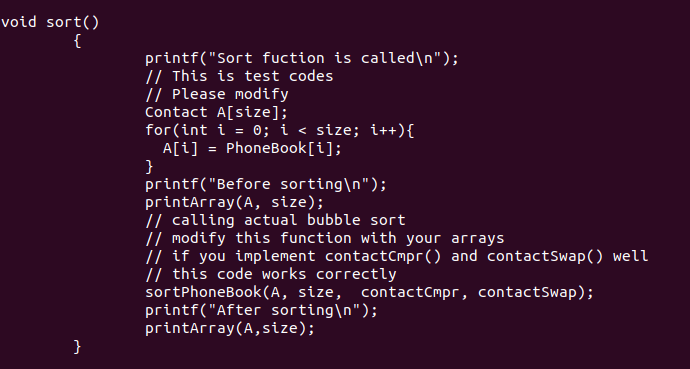
특정이름을 입력받으면 PhoneBook에서 같은 이름을 가진 자리를 찾아서 그곳의 바로 뒤에서부터 하나씩 값을 앞으로 이동시킨다. (그곳의 값은 사라진다.)

그 다음 size의 값을 1씩 줄인다.

같은 이름을 가진 자리가 없다면 oops변수를 이용하여 없다는 말을 출력시킨다.

sort.c의





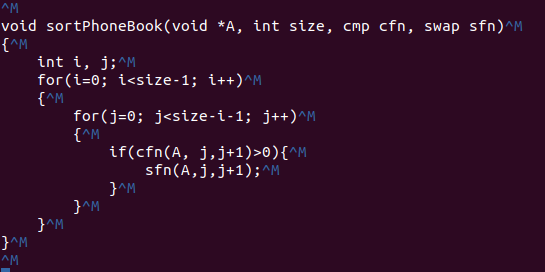
sort함수는 PhoneBook의 배열을 복사하여 printArray함수로 원래값을 출력해준 다음,

sortPhoneBook함수를 이용하여 배열과 size, contactCmpr, contactSwap함수를 매개변수로 받아 sort를 실행시킨 다음 값들을 출력시킨다.

contactCmpr는 void형 포인터와 정수, i와 j를 매개변수로 받아, 포인터를 (\*Contact)로 설정해준 다음, PhoneBook배열의 i번째 name과 j번째 name을 비교하여 i가 더 클시에 1을 리턴하고 아니면 0을 리턴한다.

contactSwap는 void형 포인터와 정수, i와 j를 매개변수로 받아, 포인터를 (\*Contact)로 설정해준 다음, PhoneBook배열의 i번째 구조체와 j번째 구조체의 내용을 서로 바꿔준다.

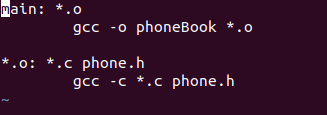
hw7Main.c의 sortPhoneBook함수



Void 형 포인터와 size, 그리고 함수 포인터로 만들어둔 cmp와 swap을 매개변수로 받아서

0부터 size까지 두 번에 걸쳐 cfn(contactCmpr)으로 검사한 다음 sfn(contactSwap)으로 서로 swap해준다.

1. Makefile 설명



main으로 phoneBook을 명령어로 만들고, 해당되는 object파일들이 gcc로 컴파일 된다.

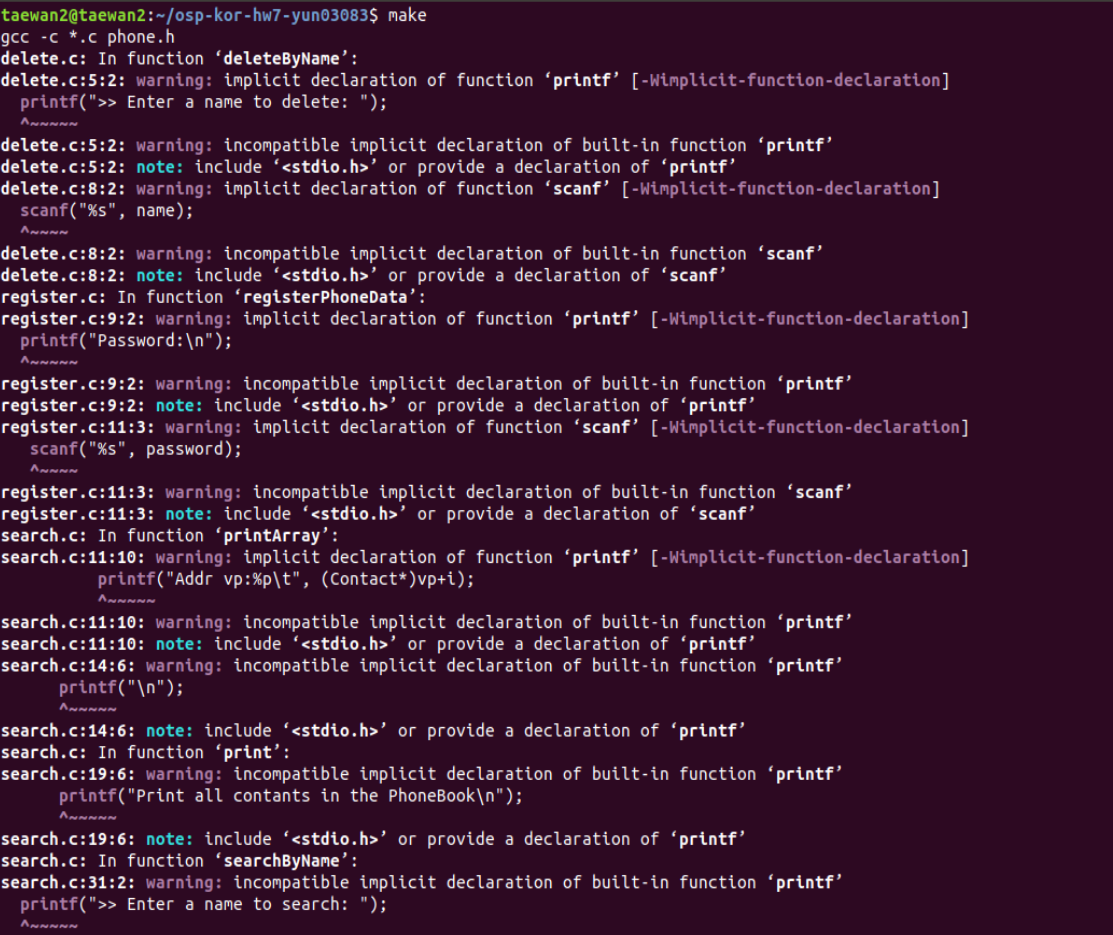
object파일들은 해당 .c 파일들과 phone.h로 구성되게끔 gcc -c로 실행된다.

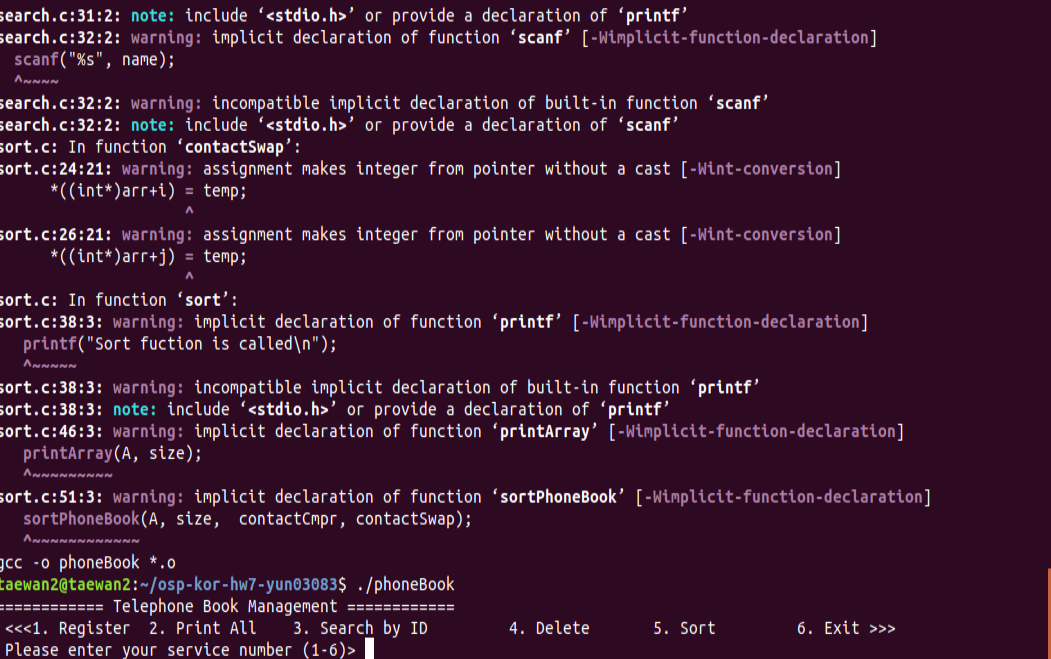
# 실행 방법 설명 (10점)

1. 사용한 운영체제 및 컴파일러의 종류

리눅스 우분투의 터미널을 썼다.

1. 컴파일 방법 및 실행 방법

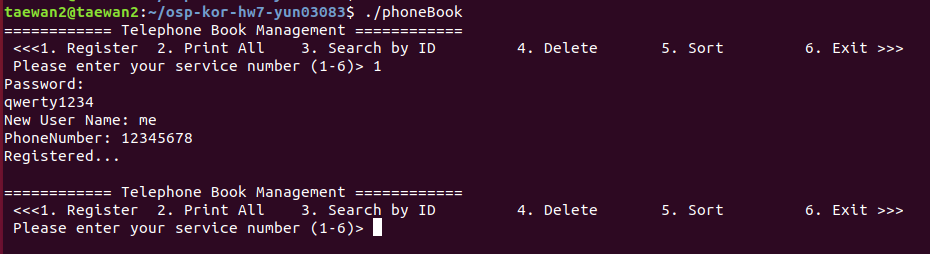




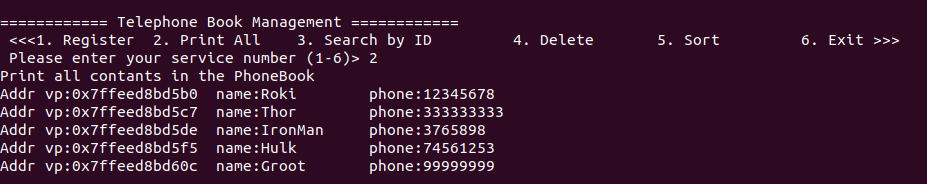
./phoneBook으로 실행시킨다.

(3) 동작을 확인할 수 있는 실행 화면 캡처

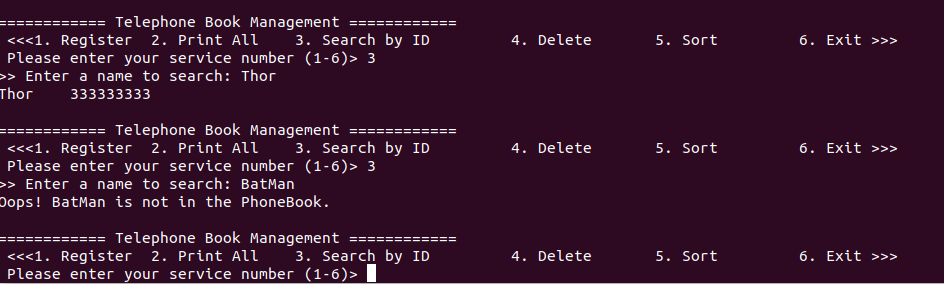
Register



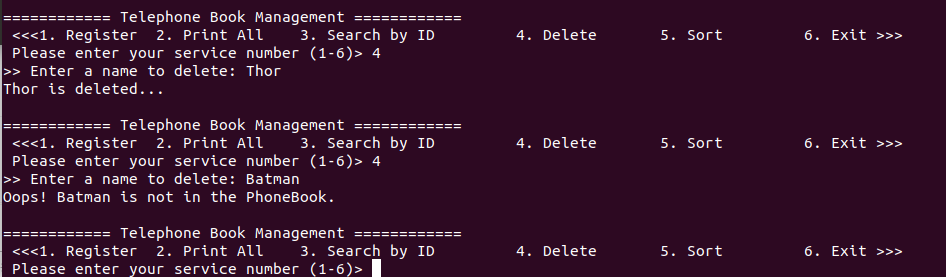
print



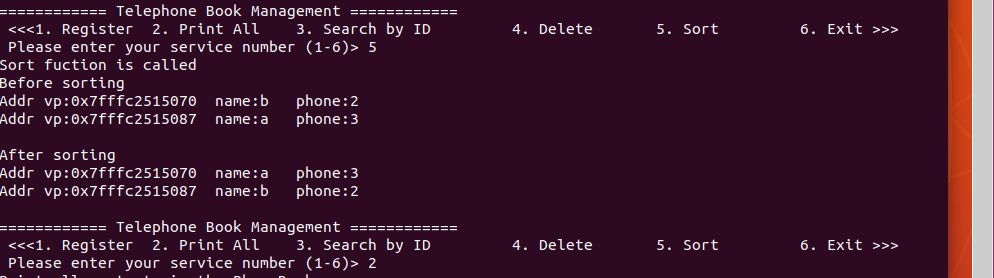
Search



Delete

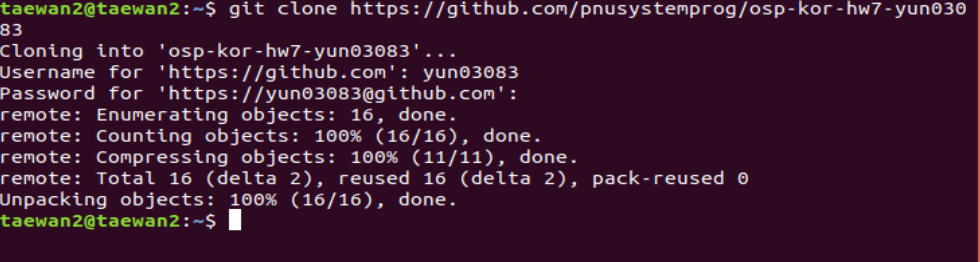


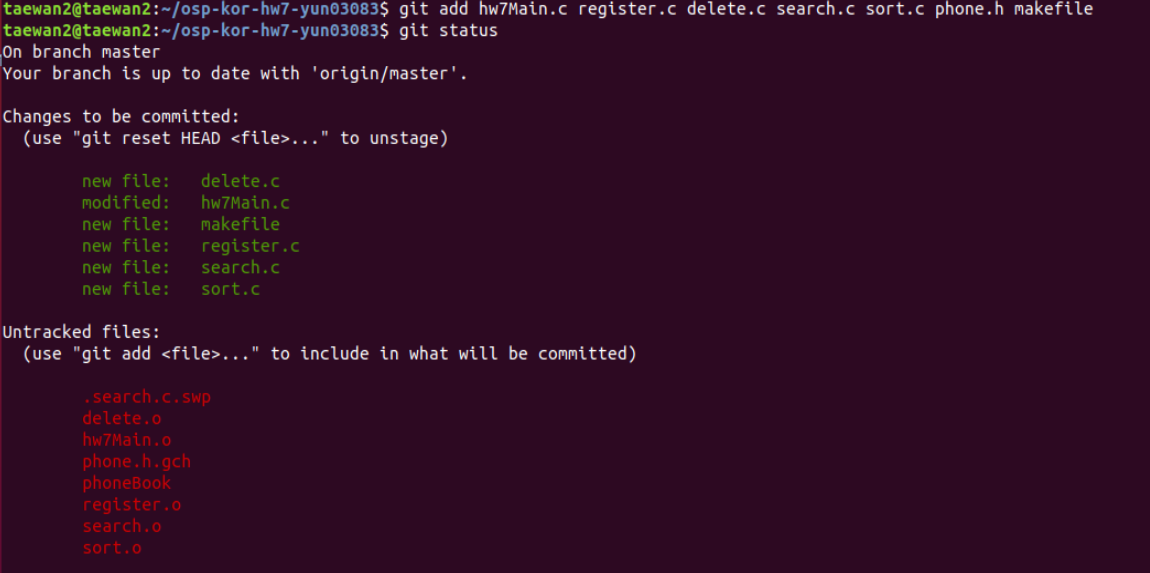
sort

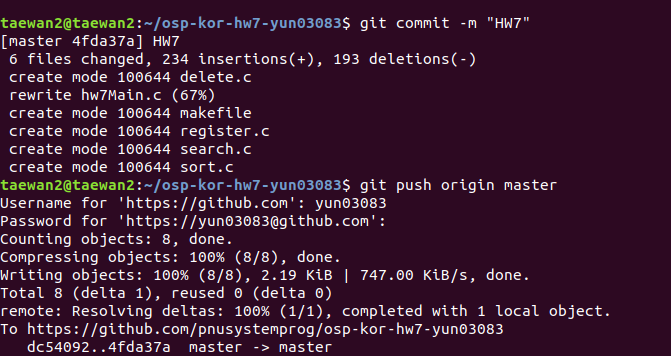


# Github 화면 (20점)

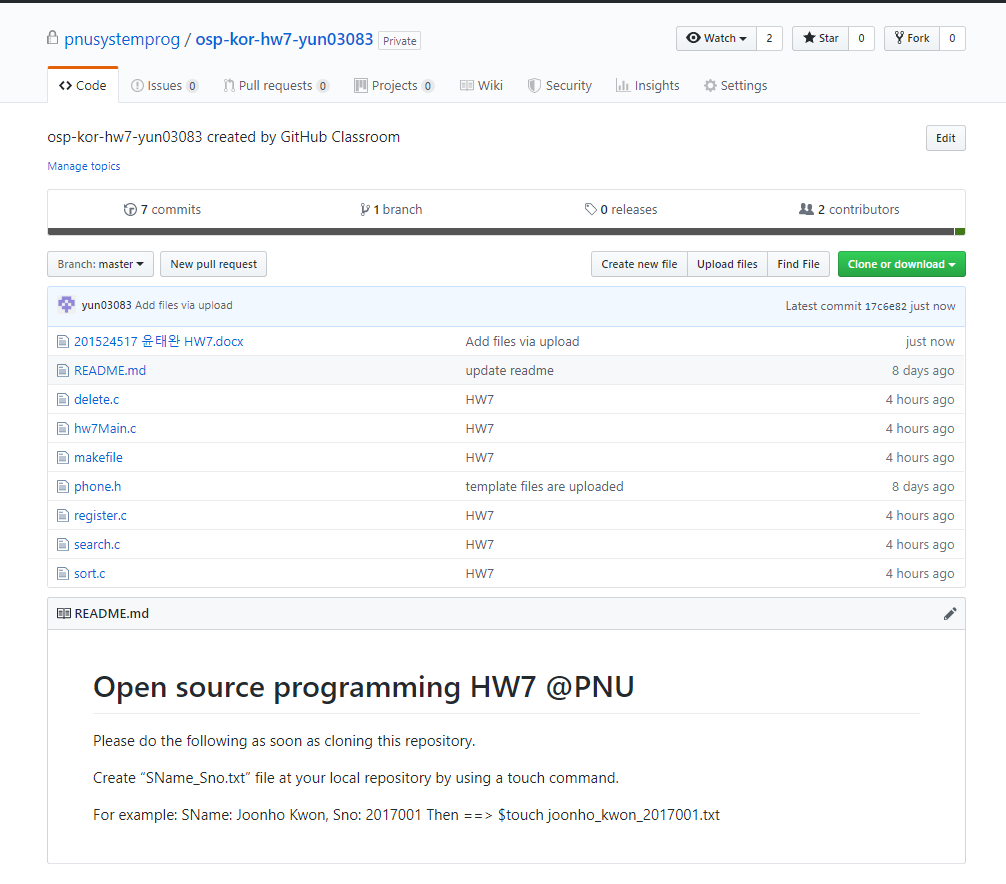
(1) cloning, adding, committing, push을 위한 github 명령들을 포함







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



# 논의 사항 (10점)

- 숙제를 하면서 새로이 알게 된 내용(수업 시간 이외의 내용),

void형 포인터와 함수포인터의 활용

- 숙제를 하는 중에 어려웠던 점 등을 기술

contactSwap을 구현할 때 오류가 많아 힘들었다. 결국 ==1로 해결하였다.

괄호를 많이 붙여서 많이 헷갈렸다.