21500630
전여훈

Lab5

1. Header 파일 생성방법, 소스, 기타설명
   A. 헤더파일은 touch menu.h 혹은 vi menu.h 로 생성한다.
   B. 헤더파일 안에는 함수의 원형을 작성하여 넣는다.

```
1 //menu.h
2
3 #include <stdio.h>
4
5 void displayMenu();
~
~
~
```

```
1 // guest.h
2
3 #include <stdio.h>
4
5 int addGuest();
6 void displayGuest(int menu);
~
~
```

2. C파일 생성, 소스, 기타 설명
   A. 편하게 C파일을 생성하기 위해서 cp menu.h menu.c 로 헤더파일을 확장자만 바꾸어 복사한다.
   B. C파일 내에서는 헤더에 선언한 함수들을 정의한다.

```
s21500630@peace:~/project/shop$ cat main.c
#include "menu.h"
#include "guest.h"

int main(){
        int menu;
        displayMenu();
        menu = addGuest();
        displayGuest(menu);
        return 0;
}
```

```
s21500630@peace:~/project/shop$ cat menu.c
#include "menu.h"

void displayMenu(){
        printf("************************\n");
        printf("1. Pizza : 20000\n");
        printf("2. Chicken : 12000\n");
        printf("************************\n");
}
s21500630@peace:~/project/shop$ cat guest.c
// guest.h

#include "guest.h"

int addGuest(){
        int op;
        printf("원하는 메뉴는? ");
        scanf("%d", &op);
        return op;
}

void displayGuest(int menu){
        if(menu == 1){
                printf("Pizza 선택\n");
        }
        else{
                printf("Chicken 선택\n");
        }
}
```

3. 개별 컴파일
   A. gcc -c -o 실행파일이름 C파일이름 으로 컴파일하여 오브젝트파일을 만든다.
   B. main.c, guest.c, main.c 총 세개의 파일을 컴파일 하고 세개의 오브젝트파일을 만든다.

4. 실행파일 생성
   A. 만들어진 세개의 오브젝트파일을 실행파일로 만든다.

```
s21500630@peace:~/project/shop$ gcc -o shop menu.o guest.o main.o
```

   >> menu.o, guest.o, main.o를 묶어서 shop이라는 이름의 실행파일 생성

5. 실행
   A. 만들어진 실행파일을 실행하기 위해서 ./실행파일이름 을 커맨드 라인에 입력하면 된다.

```
s21500630@peace:~/project/shop$ ./shop
************************
1. Pizza : 20000
2. Chicken : 12000
************************
원하는 메뉴는? 
```

6. Makefile 생성

```
1 shop : menu.o main.o guest.o
2         gcc -o shop menu.o main.o guest.o
3 clean :
4         rm *.o shop
```

   A. vi makefile(파일이름은 아무거나) 로 파일을 하나 만든다.
   B. shop : menu.o main.o guest.o //목적파일 이름을 지정하고 목적파일을 만들기 위한 파일을 나열한다.
   C. 목적파일을 만들기위한 명령어를 아래에 넣는다.

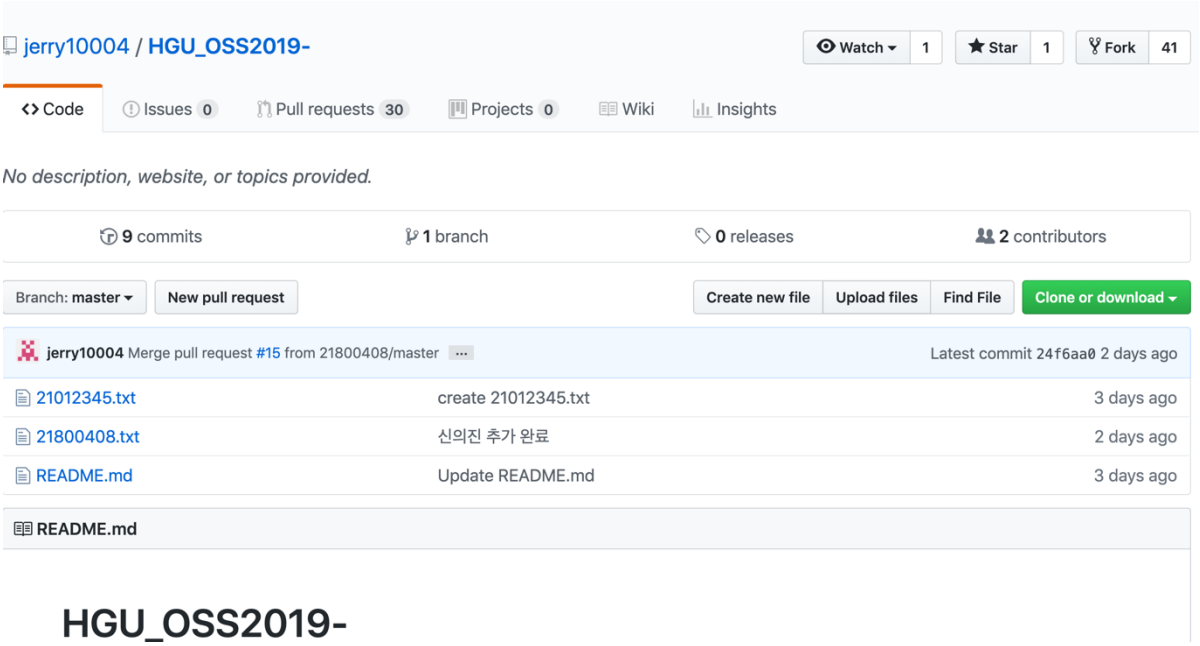
7. Makefile2 생성

```
1 CC = gcc
2 CFlags = -W -Wall // 작은 에러도 모두 출력
3 TARGET = shop
4 OBJECTS = menu.o guest.o main.o
5
6 all : $(TARGET)
7
8 $(TARGET) : $(OBJECTS)
9         $(CC) $(CFLAGS) -o $@ $^
10
11 clean :
12         rm *.o shop
```

   A. 매크로를 사용해서 더 활용성이 좋은 makefile을 만들 수 있다. 코드 상단에서 매크로를 지정하고 $(매크로이름) 을 사용하면 지정된 코드가 그대로 치환된다.

깃허브 이용하기

1. Open source repository 선택
   A. 깃허브에 가입을 한 이후 검색창에 키워드를 넣어 검색하면 연관된 오픈소스 저장소들이 나온다.



2. 하나를 선택하여 들어가면 위와 같이 저장소의 세부 정보들이 나온다.
3. 오른쪽 상단에서 포크를 누르면 이 저장소가 내 저장소로 복제이동한다.
4. 내가 코드를 수정하거나 다른 파일을 추가하고, 나 뿐만아니라 본래 코드를 배포한 사람에게도 적용하길 원한다면 왼쪽에 보이는 New pull request 를 누른다.
5. Pull requst 를 작성하면서 코멘트를 달면 원배포자가 확인할 수 있다.
6. 원 배포자가 나의 pull request 를 accept 하고 본인의 파일에 merge 하면 나의 수정사항이 원 배포자에게도 업데이트 된다.