

HW5. 구조체 배열

부산대학교 정보컴퓨터공학부

201824477

박은영

060분반

Github ID: ey5321@naver.com

제출일: 2020-06-01

보고서에는 다음의 내용을 포함할 것.

1. 구현 내용에 대한 설명 (50점)

- (1) 주요 변수 설명: extern과 static 변수는 반드시 설명함
- (2) 주요 자료 구조 설명
- (3) 주요 함수 구현 방법 설명
- (4) Makefile 설명

위 코드에서 size와 count_service를 사용하는데 size는 phone.h에 정의되어 있고 count_service는 PhoneBookMain에 정의되어 있지만 delete.c와 register.c와 search.c에서는 정의되어 있지 않으므로 외부에 정의되어 있는 것을 선언해서 사용하기 위해 extern 함수를 사용합니다. Static은 사용하지 않았습니다. registerPhoneData()는 register.c에서, printAll()는 print.c에서, searchByName()는 search.c에서, deleteByName()는 delete.c에서 정의해줍니다. make로 이 함수들을 PhoneBookMain.c에서 선언만 해줘도 쓸 수 있도록 make로 link해줍니다. 그러므로 PhoneBookMain.c에서 1이 입력되면 registerPhoneData()를 실행하고 2가 입력되면 printAll()을 실행하고 3이 입력되면 searchByName()을 실행하고 4가 입력되면 deleteByName()가 실행되고 5가 입력되면 함수를 종료합니다.

```
#define MAX 50

struct Contact
{
    char Name[10];
    char PhoneNumber[14];
};

struct Contact PhoneBook[MAX];

int size;
```

```

#include <string.h>
#include "phone.h"
extern int size;

void deleteByName()
{
    char inname[10];
    int i,j;
    printf(">>Enter a name to delete");
    scanf("%s",&inname);
    for(i=0;i<size;i++){
        if(strcmp(PhoneBook[i].Name, inname)==0){
            break;
        }
    }
    if(i<size){
        for(j=i+1;j<size;j++){
            strcpy(PhoneBook[i].Name, PhoneBook[j].Name);
            strcpy(PhoneBook[i].PhoneNumber, PhoneBook[j].PhoneNumber);
            i++;
        }
        size--;
        strcpy(PhoneBook[size].Name, "");
        strcpy(PhoneBook[size].PhoneNumber, "");
        printf("%s id deleted...\n\n", inname);
    }
    else{
        printf("Oops! %s is not in the PhoneBook\n\n");
    }
}

```

```

#include <stdio.h>
#include "phone.h"

void printAll()
{
    int i;
    for(i=0; i<size; i++){
        printf("%s\t%s\n", PhoneBook[i].Name, PhoneBook[i].PhoneNumber);
    }
    printf("\n");
}

```

```

#include <stdio.h>
#include <string.h>
#include "phone.h"
extern int size;
extern int count_service;
void registerPhoneData()
{
    char pwd[]="qwerty1234";
    char inpwd[20];
    printf("Ask to type password\n");
    printf("Password:\n");
    while((scanf("%s",&inpwd)&&(strcmp(inpwd,pwd)==0) || count_service!=3)){
        if(strcmp(inpwd,pwd)==0){
            printf("New User Name:");
            scanf("%s",PhoneBook[size].Name);
            printf("PhoneNumber:");
            scanf("%s",PhoneBook[size].PhoneNumber);
            printf("Registered...\n\n");
            count_service=0;
            size++;
            break;
        }
        else{
            count_service++;
            if(count_service==1){
                printf(">>Not Matched!!!\n");
            }
            if(count_service==2){
                printf(">>Not Matched(twice)!!!\n");
            }
            if(count_service==3){
                printf("success PhoneBook.\n\n");
                printf("success PhoneBook.\n\n");
                count_service=0;
                break;
            }
        }
    }
}

```

```

#include <stdio.h>
#include <string.h>
#include "phone.h"
extern int size;

void searchByName()
{
    int i;
    char inname[10];
    printf(">>Enter a name to search:");
    scanf("%s",&inname);
    for(i=0;i<size;i++){
        if(strcmp(PhoneBook[i].Name,inname)==0){
            break;
        }
    }
    if(i<size){
        printf("%s\t%s\n\n",PhoneBook[i].Name,PhoneBook[i].PhoneNumber);
    }
    else{
        printf("Oops! %s is not in the PhoneBook\n\n",inname);
    }
}

```

```

#include <stdio.h>
#include "phone.h"

int count_service=0;

void registerPhoneData();
void printAll();
void searchByName();
void deleteByName();

int main()
{
    int service;
    do
    {
        printf("=====Telephone Book Management=====");
        printf("\n <<<1.Register\t 2.print All\t 3.Search by ID\t 4.Delete\t 5.Exit >>>\n");
        scanf("%d",&service);

        switch(service)
        {
            case 1: registerPhoneData(); break;
            case 2: printAll(); break;
            case 3: searchByName(); break;
            case 4: deleteByName(); break;
        }
    } while(service != 5);
    return 0;
}

```

2. 실행 방법 설명 (20점)

(1) 사용한 운영체제 및 컴파일러의 종류

(2) 컴파일 방법 및 실행 방법

(3) 동작을 확인할 수 있는 실행 화면 캡처

linux에서 gcc명령어로 컴파일하였습니다. Makefile을 만들어서 gcc과정을 간단하게 하였습니다. register.o, print.o, search.o, delete.o, PhoneBookMain.o를 매크로로 OBJS로 간략하게 해줍니다. 그리고 suffix rule을 사용해서 OBJS를 link해서 main을 만들어 줍니다. 그리고 .o로 끝나는 것과 main을 rm해줍니다.

1을 입력하면 패스워드를 친 다음 패스워드가 맞으면 이름과 전화번호를 등록할 수 있습니다. 하지만 패스워드가 3번 틀리게 되면 이전으로 돌아갑니다. 2를 입력하면 등록된 사람과 전화번호가 출력됩니다. 3을 입력하고 이름을 입력하면 그 이름과 동일한 사람의 이름과 전화번호가 출력됩니다. 입력한 이름이 등록되어 있지 않은 경우 이전으로 돌아갑니다. 4를 입력하고 이름을 입력하면 그 이름과 동일한 등록된 사람의 이름과 전화번호를 지울 수 있습니다. 배열의 지워진 이름의 위치에 뒷사람들이 채워집니다. 입력된 이름이 등록된 이름에 없으면 이전화면으로 돌아갑니다. 5를 입력하면 종료됩니다.

```
C=gcc
CFLAGS = -Wall -g
OBJS = register.o print.o search.o delete.o phoneBookMain.o
main : $(OBJS)
        $(CC) -o main $(OBJS)
%.o : %.c
        $(CC) -c -o $@ $(CFLAGS) $<
clean :
        rm *.o main
```

```

eunyoung@ubuntu:~/Desktop$ vi phone.h
eunyoung@ubuntu:~/Desktop$ vi delete.c
eunyoung@ubuntu:~/Desktop$ vi print.c
eunyoung@ubuntu:~/Desktop$ vi register.c
eunyoung@ubuntu:~/Desktop$ vi search.c
eunyoung@ubuntu:~/Desktop$ vi PhoneBookMain.c
eunyoung@ubuntu:~/Desktop$ vi Makefile
eunyoung@ubuntu:~/Desktop$ make
gcc -c -o PhoneBookMain.o -Wall -g PhoneBookMain.c
gcc -o main register.o print.o search.o delete.o PhoneBookMain.o
eunyoung@ubuntu:~/Desktop$ ./main
=====Telephone Book Management=====
<<<1.Register 2.print All 3.Search by ID 4.Delete 5.Exit >>>
Please enter your service number (1-5)> 1
Ask to type password
Password:
qwerty1234
New User Name:park
PhoneNumber:010-2869-8033
Registered...

=====Telephone Book Management=====
<<<1.Register 2.print All 3.Search by ID 4.Delete 5.Exit >>>
Please enter your service number (1-5)> 1
Ask to type password
Password:
qwe123
>>Not Matched!!!
qwe1234
>>Not Matched(twice)!!!
qwerty1234
New User Name:kim
PhoneNumber:010-9891-5321
Registered...

=====Telephone Book Management=====
<<<1.Register 2.print All 3.Search by ID 4.Delete 5.Exit >>>
Please enter your service number (1-5)> 2
park 010-2869-8033
kim 010-9891-5321

=====Telephone Book Management=====
<<<1.Register 2.print All 3.Search by ID 4.Delete 5.Exit >>>
Please enter your service number (1-5)> 3
>>Enter a name to search:park
park 010-2869-8033

```

```

=====Telephone Book Management=====
<<<1.Register 2.print All 3.Search by ID 4.Delete 5.Exit >>>
Please enter your service number (1-5)> 3
>>Enter a name to search:lee
Oops! lee is not in the PhoneBook

=====Telephone Book Management=====
<<<1.Register 2.print All 3.Search by ID 4.Delete 5.Exit >>>
Please enter your service number (1-5)> 4
>>Enter a name to deletekim
kim is deleted...

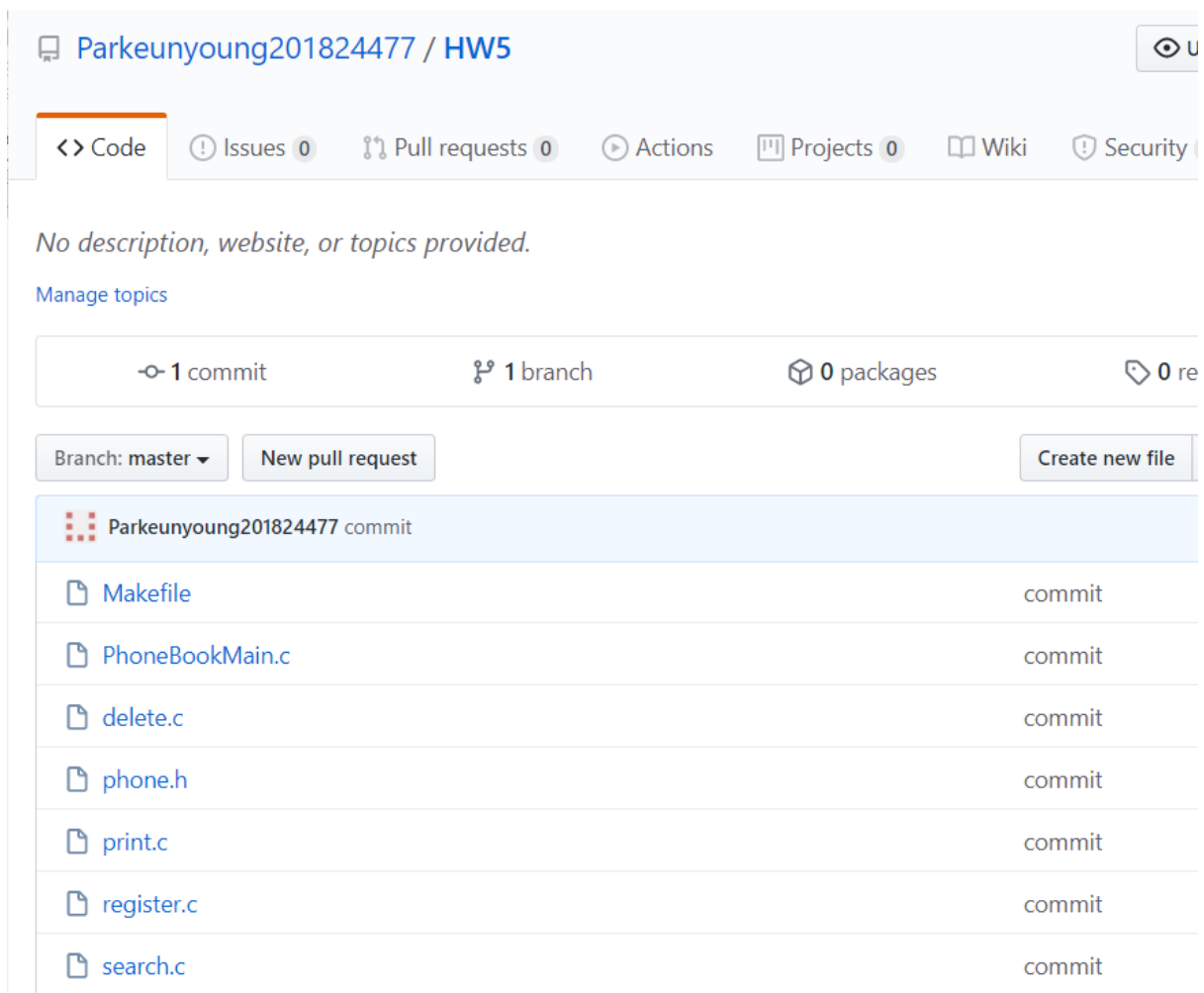
=====Telephone Book Management=====
<<<1.Register 2.print All 3.Search by ID 4.Delete 5.Exit >>>
Please enter your service number (1-5)> 2
park 010-2869-8033

=====Telephone Book Management=====
<<<1.Register 2.print All 3.Search by ID 4.Delete 5.Exit >>>
Please enter your service number (1-5)> 5
eunyoung@ubuntu:~/Desktop$

```

3. Github 화면 (20점)

- (1) cloning, adding, committing, push을 위한 github 명령들을 포함
- (2) 소스 코드와 makefile을 push한 후, 본인의 Github repository를 스크린 캡처하여 포함
 - git init(.git이라는 디렉토리 생성)
 - git add . (모든 파일 추가)
 - git commit -m "commit" (commit에 대한 메시지를 남김)
 - git remote add origin1 <http://github.com/Parkeunyoung201824477/HW5.git> (github의 링크를 입력, origin1이라는 remote생성)
 - git push -u origin1 master (HEAD에 저장된 내용을 origin1을 통해 github에 저장)



4. 논의 사항 (10점)

- 숙제를 하면서 새로이 알게 된 내용(수업 시간 이외의 내용),
 - 숙제를 하는 중에 어려웠던 점 등을 기술
- github을 처음 사용해보아서 다루는게 어려웠습니다. 처음에 git을 설치하고 사용자이름과 이메일

주소를 최초설정 해줘야 되는 것을 몰라 해결방법을 찾는데 애를 먹었지만 나머지는 문제없이 잘 업로드할 수 있었습니다.