

## HW5. 구조체 배열

학번: 정보컴퓨터공학과 2020-55565

이름: 060분반 여지수

Github ID: duwltn1301@naver.com

제출일: 2021-05-19

### 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

등록된 총 개인정보 수를 뜻하는 size 변수는 main 함수가 들어있는 소스코드에서 0으로 선언하고, 다른 소스코드에서는 extern으로 선언하여 다른 소스코드에서도 공유할 수 있도록 한다. register.c 소스코드에 있는 함수가 호출되면 size는 +1이 되도록, delete.c 소스코드에 있는 함수가 호출되면 size는 -1이 되도록 구현했다. 그리고 print.c와 serach.c 소스코드에 있는 함수들은 이 size 수로 번호가 저장된 위치를 찾을 수 있는 것이다. size는 각각의 소스코드에서 모두 공유되어 사용되기에 이처럼 extern으로 선언하게 된 것이다.

서비스가 몇번 호출되었는지 카운팅하는 count\_service 변수는 static으로 선언하여 메인함수가 들어있는 소스파일 외의 다른 소스파일에서는 이 변수에 접근하지 못하도록 한다.

메인 함수가 들어있는 phoneBookMain.c 에서는 stdio.h string.h phone.h 헤더파일을 포함하고 메인에서 쓰일 함수들을 선언하며, 정적 변수 count\_service와 공유될 변수 size를 0으로 선언한다. 그리고 main함수는 1,2,3,4,5 서비스 중 하나를 호출하도록 구현했다. 5(exit)를 입력하지 않는 이상 서비스는 while문을 통해 계속된다. 1을 입력하면 resgisterPhoneData() 함수가, 2를 입력하면 printAll() 함수가, 3을 입력하면 searchByName()함수가, 4를 입력하면 deleteByName() 함수가 호출된다. 마지막으로 서비스가 종료될 때 서비스를 몇번 이용했는지 service count를 출력한다.

phone.h 헤더파일에는 구조체를 선언하고, #ifndef와 #endif 지시자를 통해서 중복 선언을 막았다.

register.c 에는 extern int size를 선언하고 registerPhoneData()함수를 구현했다. 우선 등록할 수 있는 전화번호는 50개까지 이므로 size가 50이상이면 더이상 추가 할 수 없음을 출력한다. 만약 size가 50보다 작다면, 등록할 기회를 주는 데 우선 패스워드를 입력하도록 한다. 맞춰야 할 password를 qwerty1234로 선언하고 strcmp를 이용해 입력 받은 문자열과 맞춰야 할 password가 같은지 비교하고 같으면 등록을 받도록 하고 다르면 에러 메시지를 띄운다. 3번을 초과하여 틀리면 더이상 패스워드를 입력받지 못하도록 한다. 맞췄다면, size번째의 PhoneBook 구조체에 이름과 전화번호를 저장하고 size는 1 늘린다.

print.c에는 extern int size를 선언하고 printAll()함수를 구현했다. PhoneBook 구조체에 접근하여 for문을 이용해 구조체 배열 전체를 출력한다.

search.c에는 extern int size를 선언하고 searchByName()함수를 구현했다. 찾을 이름을 입력받고, for 문을 이용해 구조체에 접근하여 0번째 구조체 배열부터 그 구조체 배열의 name 과 입력받은 문자열을 비교한다. 같으면 0으로 선언했었던 find 변수를 1로 바꾸고 그 번째수 구조체 배열의 전화번호를 미리 선언했던 빈 phone변수에 저장해둔다. for 문을 다 돌고 나서 find가 1이 되었다면 사용자 입력한 이름이 전화번호부에 있었다는 것이고, find가 변하지 않았다면 사용자가 전화번호부에 없었다는 것이므로, 찾았을 때 찾지 못했을 때를 각각 알맞게 출력한다. 찾았을 때는 이름과 저장한 phone 변수를 출력하고 찾지 못했을 때는 없다고 출력한다.

delete.c에는 extern int size를 선언하고 deleteByName()함수를 구현했다. searchByName()함수와 마찬가지로 구현하는데, 차이점을 얘기하자면, 여기서는 for문을 돌려서 i번째수 구조체 배열의 이름과 사용자가 입력한 문자열이 일치하면, i번째부터 마지막 번째 수까지 한칸씩 앞으로 이동한다. 그럼 i번째 구조체 배열은 i+1 구조체 배열로 대치되면서 없어질 것이고, 맨 뒤에 남은 구조체 배열의 원소는 strcpy를 이용해 "" null로 복사한다. 이름이 삭제되었다는 메시지를 출력하고 하나 삭제 되었으니 size는 1 줄인다. 그리고 searchByName()가 마찬가지로 이름을 찾으면 find는 1로 하여 만일 find가 0이면 전화번호부에 그 이름이 없음을 알리는 메시지를 출력한다.

makefile에 대해 설명하자면,

CC=gcc

-> 매크로 CC를 정의한다. 컴파일러를 세팅해서 환경변수로 지정해주는 것이다. 버전을 바꿔서 컴파일할 때 유용하게 쓸 수 있다.

CFLAGS = -Wall -g -c

-> 매크로 CFLAGS를 정의한다. gcc의 옵션을 추가해주는 용도이다. -g는 디버그 정보를 추가하라는 것이다.

INCLUDE = -I.

-> 디렉토리를 추가한다.

LIBS = -L. -lm

-> 링크할 때 필요한 라이브러리를 추가한다.

OBJS = phoneBookMain.o register.o delete.o print.o search.o

-> 오브젝트 파일들을 정의해준다.

all : main

-> make 는 makefile 을 순차적으로 읽어서 가장 처음에 나오는 규칙을 수행하게 된다. all 은 더미타겟(dummy target)이 바로 첫 번째 타겟으로써 작용하게 된다. 결과 파일이 많을 때 all 의 의존 관계(dependency)로써 정의해 두면 편리하다.

%.o: %.c

\$(CC) \$(INCLUDE) \$(CFLAGS) \$<

-> 내부적으로 확장자 규칙이 적용되었다. \$(...)은 매크로의 사용을 의미한다. 매크로를 통해 복잡한 것을 간단하게 나타낼 수 있다. \$<는 입력파일로 콜론의 오른쪽에 오는 패턴을 치환한다는 의미를 담고있다. %는 일치하는 확장자를 제외한 파일명을 의미한다. 그래서 %.c를 컴파일하여 같은 이름의 오브젝트 파일을 만들겠다는 것이다. 즉 어떤 .o 파일 이 필요할 경우 이 규칙에 의해서 해당 .c 파일을 찾고 아랫줄의 명령을 실행시켜서 .o 파일을 생성해내는 것이다.

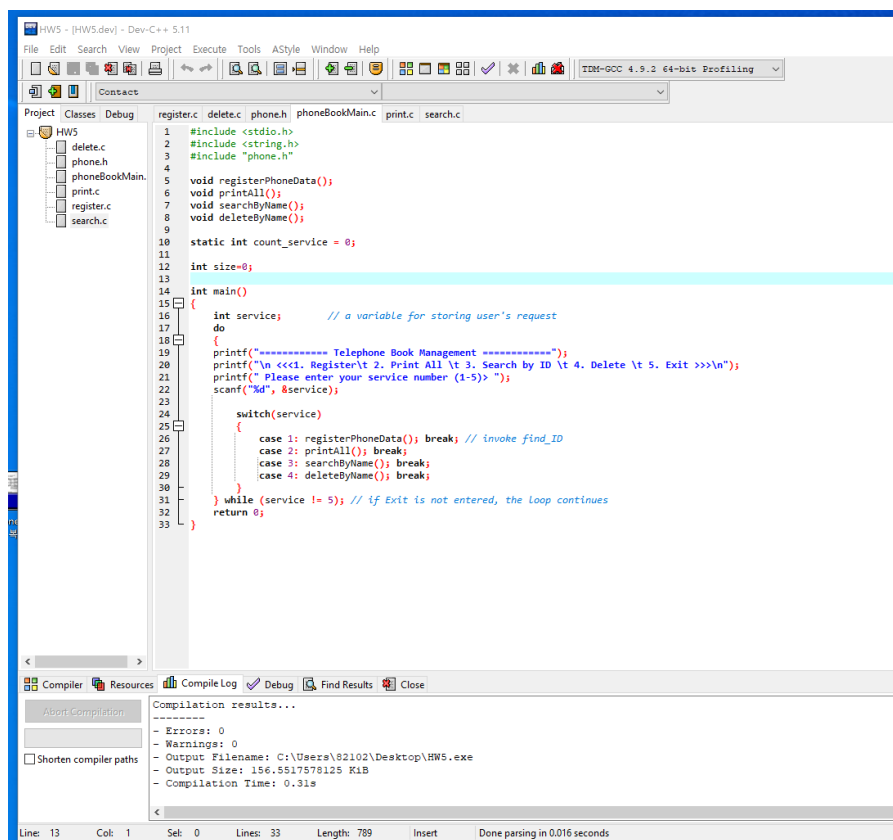
```
main: $(OBJJS)
$(CC) -o main $(OBJJS) $(LIBS)

clean:

rm -f main $(OBJJS)
```

## 2. 실행 방법 설명 (20점)

- (1) 사용한 운영체제 및 컴파일러의 종류 -> 윈도우 운영체제에서 Dev-C++ 컴파일러를 이용하여 프로그램을 구현했다.
- (2) 컴파일 방법 및 실행 방법 -> Dev-c++ 컴파일로 작성한 코드를 Compile & Run 한다. 컴파일은 전처리-> 컴파일-> 어셈블-> 링크 이 네 가지 단계로 진행된다. 우선, 전처리 단계에서는 전처리가 소스 코드에서 #으로 시작하는 지시자를 처리한다. 전처리는 #include를 만나면 뒤의 헤더 파일을 그 파일의 내용을 순차적으로 삽입 한다. .c 파일로부터 .i 파일이 생성된다. 컴파일 단계에서는 컴파일러가 전처리가 끝난 파일을 컴파일하여 운영체제가 인식할 수 있는 형태의 개체파일을 만든다. .i 파일로부터 .s 파일이 생성된다. 어셈블 단계에서는 어셈블러가 완전한 기계어로 바꾸어준다. .s 파일로부터 .o 파일이 생성된다. 링크 단계에서는 오브젝트 파일에 표준 C 라이브러리, 사용자 라이브러리가 결합되어 main 함수를 호출하여 프로그램의 코드가 실행되도록 한다. .o 파일로부터 .exe 파일이 생성된다.
- (3) 동작을 확인할 수 있는 실행 화면 캡처



```
C:\Users\W82102\Desktop\HHW5.exe
===== Telephone Book Management =====
<<<1. Register  2. Print All  3. Search by ID      4. Delete      5. Exit >>>
Please enter your service number (1-5)> 1
Password: qwert
Not Matched !!!
Password: qwerty12
Not matched (twice) !!!
Password: qwerty1234
New User Name:jisu
PhoneNumber:010-2542-3650
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
Password: jisu
Not Matched !!!
Password: qwerty1234
New User Name:mom
PhoneNumber:010-4306-3650
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
<< Display all contacts in the PhoneBook >>
jisu      010-2542-3650
mom       010-4306-3650
===== 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: dad
Oops! dad 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)> 3
Enter a name to search: mom
mom       010-4306-3650
===== 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 delete: mom
mom 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)> 1
Password: qwerty1234
New User Name:brother
PhoneNumber:010-7777-7777
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
<< Display all contacts in the PhoneBook >>
jisu      010-2542-3650
brother   010-7777-7777
===== 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: jisu
jisu      010-2542-3650
===== Telephone Book Management =====
<<<1. Register  2. Print All  3. Search by ID      4. Delete      5. Exit >>>
Please enter your service number (1-5)> 1
Password: qwerty1234
New User Name:dadd
PhoneNumber:010-4545-6565
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
<< Display all contacts in the PhoneBook >>
jisu      010-2542-3650
brother   010-7777-7777
dadd      010-4545-6565
===== Telephone Book Management =====
<<<1. Register  2. Print All  3. Search by ID      4. Delete      5. Exit >>>
Please enter your service number (1-5)> 5
service count: 11
=====
Process exited after 110 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .
```

### 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

mkdir homework

cd homework

로 homework 디렉토리를 만들고 그 디렉토리로 옮겨가서 cloning 한다.

git clone https://github.com/datalab-pnu/unix-hw5-YeoJiSu

로 cloning하고

touch jis\_u\_yeo\_202055565.txt

git add jis\_u\_yeo\_202055565.txt

하여 txt 파일을 add 한다.

```
duwlt1301@ubuntu:~/homework$ git clone https://github.com/datalab-pnu/unix-hw5-YeoJiSu
Cloning into 'unix-hw5-YeoJiSu'...
Username for 'https://github.com': duwlt1301@naver.com
Password for 'https://duwlt1301@naver.com@github.com':
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), done.
duwlt1301@ubuntu:~/homework$ ls -al
total 16
drwxrwxr-x  4 duwlt1301 duwlt1301 4096 May 19 04:02 .
drwxr-xr-x 28 duwlt1301 duwlt1301 4096 May 19 03:55 ..
drwxrwxr-x  3 duwlt1301 duwlt1301 4096 May 19 03:56 github.com
drwxrwxr-x  3 duwlt1301 duwlt1301 4096 May 19 04:06 unix-hw5-YeoJiSu
duwlt1301@ubuntu:~/homework$ ls
github.com  unix-hw5-YeoJiSu
duwlt1301@ubuntu:~/homework$ cd unix-hw5-YeoJiSu
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ ls -al
total 24
drwxrwxr-x  3 duwlt1301 duwlt1301 4096 May 19 04:06 .
drwxrwxr-x  4 duwlt1301 duwlt1301 4096 May 19 04:02 ..
drwxrwxr-x  8 duwlt1301 duwlt1301 4096 May 19 04:06 .git
-rw-rw-r--  1 duwlt1301 duwlt1301 1578 May 19 04:06 phoneBookMain.c
-rw-rw-r--  1 duwlt1301 duwlt1301  202 May 19 04:06 phone.h
-rw-rw-r--  1 duwlt1301 duwlt1301  993 May 19 04:06 README.md
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ touch jis_u_yeo_202055565.txt
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ git add jis_u_yeo_202055565.txt
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   jis_u_yeo_202055565.txt
```

touch 명령어로 헤더파일, c파일, makefile을 생성하고 이를 모두 git add 했다.

```
JIAU - duwlt1301@ubuntu: ~/homework/unix-hw5-YeoJiSu - Xshell 7 (Free for Home/School)
파일(F) 편집(E) 보기(V) 도구(T) 탭(B) 창(W) 도움말(H)
ssh://duwlt1301:*****@192.168.96.129:22
원격 버튼을 클릭하여 현재 세션을 추가할 수 있습니다.
세션 관리
모든 세션
hw
JIAU
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ touch phone.h
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ touch phoneBookMain.c
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ touch register.c
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ touch delete.c
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ touch print.c
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ touch search.c
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ touch makefile
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ ls
delete.c      makefile      phone.h      README.md    search.c
jisu_yeo_202055565.txt  phoneBookMain.c  print.c      register.c
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ git add *.c
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ git add *.h
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ git add makefile
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   delete.c
    new file:   jisu_yeo_202055565.txt
    new file:   makefile
    new file:   print.c
    new file:   register.c
    new file:   search.c
```

touch 명령어로 생성한 파일들에는 아무 내용이 없기에  
내가 기존에 작성했던 헤더파일, c파일, makefile을 /home/duwlt1301/homework/unix-hw5-YeoJiSu 디렉토리로 옮기고  
git commit -m "Add a txt file and source codes" 명령을 했다. 그 후  
git push origin master  
을 하면 파일들이 모두 잘 업로드 되어 있다.

```
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ git commit -m "Add a txt file and source codes"
[master ce7666f] Add a txt file and source codes
 6 files changed, 159 insertions(+), 73 deletions(-)
 rewrite phoneBookMain.c (80%)
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
duwlt1301@ubuntu:~/homework/unix-hw5-YeoJiSu$ git push origin master
Username for 'https://github.com': duwlt1301@naver.com
Password for 'https://duwlt1301@naver.com@github.com':
Counting objects: 8, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 2.10 KiB | 536.00 KiB/s, done.
Total 8 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/datalab-pnu/unix-hw5-YeoJiSu
 a596758..ce7666f master -> master
```

## 4. 논의 사항 (10점)

- 새롭게 알게된 점은 extern으로 변수를 선언하면 여러 개의 소스코드에서 변수를 공유할 수 있다는 점이다. 그런데 extern으로 선언했는데도 중복 선언이 되었다는 에러 메시지가 계속 떴서, 분할 컴파일 하는 부분이 조금 힘들었다. main함수가 들어있는 소스코드에만 int 변수 = 0; 이라 선언하고 다른 모든 소스코드에 external int 변수; 로 선언을 하니 해결이 되었다. 함수에도 extern을 붙일 수 있다는 것을 새롭게 알게 되었는데, 함수의 경우는 extern이 없으면 default가 external linkage이고 static을 붙여서 internal로 만든다는 것을 알게되었다.