

HW6. 함수 포인터와 void 포인터

부산대학교 정보컴퓨터공학부

분반: 060

201824477

박은영

Github ID: ey5321@naver.com

제출일: 2020-06-08

보고서에는 다음의 내용을 포함할 것.

1. 구현 내용에 대한 설명 (60점)

- (1) 주요 변수 설명: 함수 포인터 설명
- (2) 주요 자료 구조 설명
- (3) 주요 함수 구현 방법 설명
 - Void 형 포인터 사용 시 동작 원리 자세히 설명
- (4) Makefile 설명

```
#include <stdio.h>
#include "phone7.h"
#include <string.h>

void registerPhoneData();
void print();
void printArray(void *arr, int size);
void searchByName();
void deleteByName();
void sort();

int size = 0;
int count_service = 0;

void (* pFuncs[5])()={registerPhoneData, print, searchByName, deleteByName,sort};

typedef int (*cmp)(const void*, int, int);
typedef int (*swap)(void*, int, int);

void sortPhoneBook(void *A, int size, cmp cfn, swap sfn);

int main()
{
    int service;          // a variable for storing user's request
    do
    {
        printf("===== Telephone Book Management =====");
        printf("\n <<<1. Register\t 2. Print All \t 3. Search by ID \t 4. Delete \t 5. Sort \t 6. Exit >>>\n");
        printf(" Please enter your service number (1-6)> ");
        scanf("%d", &service);
        if ( service > 0 && service <= 5 )
        {
            pFuncs[service-1]();
        }
        else if( service ==0 || service > 6)
        {
            printf("You choose a wrong service number\n");
        }
    } while (service != 6); // if Exit is not entered, the loop continues
    return 0;
}
```

-> hw7Main.c (과제 번호를 착각하여 이름을 hw7로 한 점 양해부탁드립니다)

```
#include <stdio.h>
#include "phone7.h"
#include <string.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 is deleted...\n\n", inname);
    }
    else
        printf("Oops! %s is not in the PhoneBook\n\n");
}
```

-> delete7.c

-> phone7.h

```
# define MAX 50

typedef struct contact_st
{
    char Name[10];
    char PhoneNumber[14];
}contact;

contact PhoneBook[MAX];
```

```

#include <stdio.h>
#include "phone7.h"
extern int size;

void print()
{
    printf("Print all contents in the PhoneBook\n");
    printArray(PhoneBook, size);
}

void printArray(void *arr, int size)
{
    int i;
    void *vp;
    vp= arr;
    for(i=0; i<size; i++)
    {
        printf("Addr vp:%p\t", (contact*)vp+i);
        printf("name:%s\t phone:%s\n", ((contact*)vp+i)->Name, ((contact*)vp+i)->PhoneNumber);
    }
    printf("\n");
}
~

```

-> print7.c

```

#include <stdio.h>
#include <string.h>
#include "phone7.h"
extern int count_service;
extern int size;

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("PhonNumber:");
            scanf("%s", PhoneBook[size].PhoneNumber);
            printf("Registered...\n\n");
            count_service=0;
            size++;
            break;
        }
        else{
            count_service+=1;
            if(count_service==1)
                printf(">>Not Matched!!!\n");
            if(count_service==2)
                printf(">>Not Matched(twice)!!!\n");
            if(count_service==3){
                printf("cccees PhoneBook.\n\n");
                count_service=0;
                break;
            }
        }
    }
}
}

```

-> register7.c

```
#include <stdio.h>
#include "phone7.h"
#include <string.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", PHONEBOOK[i].Name, PHONEBOOK[i].PhoneNumber);
    else
        printf("Oops! %s is not in the PhoneBook\n", inname);
}
```

-> search7.c

```

#include <stdio.h>
#include "phone7.h"
#include <string.h>
extern int size;

typedef int (*cmp)(const void*, int, int);
typedef int (*swap)(void*, int, int);

int contactCmpr(const void*, int, int);
int contactSwap(void*, int, int);

void sortPhoneBook(void *A, int size, cmp cfn, swap sfn)
{
    int i, j;
    for(i=0; i<size-1; i++)
    {
        for(j=0; j<size-1; j++)
        {
            if(cfn(A, j,j+1)>0)
                sfn(A,j,j+1);
        }
    }
}

void sort()
{
    printf("Sort fuction is called\n");
    printf("Before sorting\n");

    printArray(PhoneBook, size);

    sortPhoneBook(PhoneBook, size, contactCmpr, contactSwap);
    printf("After sorting\n");

    printArray(PhoneBook,size);
}

int contactSwap(void* arr, int i, int j)
{
    contact temp;
    contact* carr = (contact*)arr;
    temp = *(carr+j);
    *(carr+j) = *(carr+i);
    *(carr+i) = temp;
    return 1;
}

```

-> sort7.c

```

return 1;
}

int contactCmpr(const void* arr, int i, int j)
{
    if( strcmp( ((contact*)arr+i)->Name, ((contact*)arr+j)->Name) > 0)
        return 1;

    else
        return 0;
}

```

```
CC = gcc
CFLAGS = -Wall -g
OBJS = hw7Main.o register7.o print7.o search7.o delete7.o sort7.o
main7 : $(OBJS)
    $(CC) -o main7 $(OBJS)
%.o : %.c
    $(CC) -c -o $@ $(CFLAGS) $<
clean :
    rm *.o main7
```

-> Makefile

Makefile을 만들어서 gcc과정을 간단하게 하였습니다. register7.o sort7.o print7.o search7.o delete7.o hw7Main.c를 매크로로 OBJS로 간략하게 해줍니다. 그리고 suffix rule을 사용해서 OBJS를 link해서 main을 만들어 줍니다. .o로 끝나는 것과 main을 rm해줍니다.

register7.c에서 hw7Main.c에 정의된 count_service를 사용하기 위해 extern을 사용합니다. register7.c sort7.c search7.c print7.c delete7.c hw7Main.c에서 정의된 size를 사용하기 위해 extern을 사용합니다.

typedef int (*cmp)(const void*, int, int); typedef int (*swap)(void*, int, int);를 써줌으로써 cmp cfn 은 cfn(const void*, int, int)이고 swap sfn은 sfn(void*, int, int)입니다. void*는 임의의 타입의 포인터 변수를 대입할 수 있지만 메모리를 가지지 못해 주소연산이 불가능합니다. 그러므로 타입 캐스팅을 해줘야 합니다. contactSwap(void* arr, int i, int j)에서 void*은 주소연산이 안되므로 contact*로 변환해준 뒤 주소연산을 합니다. contactSwap(void* arr, int i, int j)에서 void*는 contact[i]의 Name을 참조하기 위해 contact*로 타입을 변환해준 뒤 주소연산을 합니다. void (* pFuncs[5])()는 void*형 함수를 5개를 가지는 배열입니다.

sort7.c에서 contactCmpr(const void* arr, int i, int j)에서 strcmp로 i번째 contact.Name과 j번째 contact.Name을 비교해서 i번째가 j번째보다 뒤순서이면 contactSwap(void* arr, int i, int j)을 실행

해서 순서를 바꿔줍니다.

2. 실행 방법 설명 (10점)

- (1) 사용한 운영체제 및 컴파일러의 종류
- (2) 컴파일 방법 및 실행 방법
- (3) 동작을 확인할 수 있는 실행 화면 캡처

linux에서 gcc명령어로 컴파일하였습니다.

실행결과입니다.

```
eunyoung@ubuntu:~/Desktop$ ./main7
=====Telephone Book Management=====
<<<1. Register  2. Print All  3. Search by ID      4.Delete      5. sor
t      6. Exit >>>
Please enter your service number (1-6)> 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. sor
t      6. Exit >>>
Please enter your service number (1-6)> 1
Ask to type password
Password:
qwerty1234
New User Name: kim
PhoneNumber: 111-1111-1111
Registered...

=====Telephone Book Management=====
<<<1. Register  2. Print All  3. Search by ID      4.Delete      5. sor
t      6. Exit >>>
Please enter your service number (1-6)> 1
Ask to type password
Password:
qwerty1234
New User Name: lee
PhoneNumber: 222-2222-2222
Registered...

=====Telephone Book Management=====
<<<1. Register  2. Print All  3. Search by ID      4.Delete      5. sor
t      6. Exit >>>
Please enter your service number (1-6)> 1
Ask to type password
Password:
qwerty1234
New User Name: jung
PhoneNumber: 333-3333-3333
Registered...
```

```

=====Telephone Book Management=====
<<<1. Register  2. Print All  3. Search by ID      4.Delete      5. sort
t      6. Exit >>>
Please enter your service number (1-6)> 2
Print all contents in the PhoneBook
Add vp: 0x55ba249b1080  name: park      phone: 010-2869-8033
Add vp: 0x55ba249b1098  name: kim       phone: 111-1111-1111
Add vp: 0x55ba249b10b0  name: lee       phone: 222-2222-2222
Add vp: 0x55ba249b10c8  name: jung      phone: 333-3333-3333

=====Telephone Book Management=====
<<<1. Register  2. Print All  3. Search by ID      4.Delete      5. sort
t      6. Exit >>>
Please enter your service number (1-6)> 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. sort
t      6. Exit >>>
Please enter your service number (1-6)> 5
Sort function is called
Before sorting
Add vp: 0x55ba249b1080  name: park      phone: 010-2869-8033
Add vp: 0x55ba249b1098  name: kim       phone: 111-1111-1111
Add vp: 0x55ba249b10b0  name: lee       phone: 222-2222-2222
Add vp: 0x55ba249b10c8  name: jung      phone: 333-3333-3333

After sorting
Add vp: 0x55ba249b1080  name: jung      phone: 333-3333-3333
Add vp: 0x55ba249b1098  name: kim       phone: 111-1111-1111
Add vp: 0x55ba249b10b0  name: lee       phone: 222-2222-2222
Add vp: 0x55ba249b10c8  name: park      phone: 010-2869-8033

=====Telephone Book Management=====
<<<1. Register  2. Print All  3. Search by ID      4.Delete      5. sort
t      6. Exit >>>
Please enter your service number (1-6)> 4
>>Enter a name to delete: jung
jung is deleted...

=====Telephone Book Management=====
<<<1. Register  2. Print All  3. Search by ID      4.Delete      5. sort
t      6. Exit >>>

=====Telephone Book Management=====
<<<1. Register  2. Print All  3. Search by ID      4.Delete      5. sort
t      6. Exit >>>
Please enter your service number (1-6)> 2
Print all contents in the PhoneBook
Add vp: 0x55ba249b1080  name: kim       phone: 111-1111-1111
Add vp: 0x55ba249b1098  name: lee       phone: 222-2222-2222
Add vp: 0x55ba249b10b0  name: park      phone: 010-2869-8033

=====Telephone Book Management=====
<<<1. Register  2. Print All  3. Search by ID      4.Delete      5. sort
t      6. Exit >>>
Please enter your service number (1-6)> 6

```

1번에서 park, kim, lee, jung과 전화번호를 등록합니다. 2번을 눌러 입력된 순서대로 이름과 전화번호를 출력합니다. 3번으로 park을 치면 이름과 전화번호가 출력됩니다. 5번으로 이름 순서대로 sort 되어 바뀐 순서대로 출력됩니다. 4번에서 jung을 입력하여 jung의 이름과 전화번호가 삭제됩니다. 6번을 눌러 프로그램을 종료합니다.

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

```
eunyoung@ubuntu:~/Desktop$ cd hw6
eunyoung@ubuntu:~/Desktop/hw6$ git init
Initialized empty Git repository in /home/eunyoung/Desktop/hw6/.git/
eunyoung@ubuntu:~/Desktop/hw6$ git add *7.c
eunyoung@ubuntu:~/Desktop/hw6$ git add hw7Main.c
eunyoung@ubuntu:~/Desktop/hw6$ git add Makefile
eunyoung@ubuntu:~/Desktop/hw6$ git add phone7.h
eunyoung@ubuntu:~/Desktop/hw6$ git commit -m "commit"
[master (root-commit) 8159135] commit
 8 files changed, 228 insertions(+)
 create mode 100644 Makefile
 create mode 100644 delete7.c
 create mode 100644 hw7Main.c
 create mode 100644 phone7.h
 create mode 100644 print7.c
 create mode 100644 register7.c
 create mode 100644 search7.c
 create mode 100644 sort7.c
eunyoung@ubuntu:~/Desktop/hw6$ git remote add origin http://github.com/Parkeunyoung201824477/hw6.git
eunyoung@ubuntu:~/Desktop/hw6$ git push -u origin master
Username for 'https://github.com': Parkeunyoung201824477
Password for 'https://Parkeunyoung201824477@github.com':
warning: redirecting to https://github.com/Parkeunyoung201824477/hw6.git/
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 2.64 KiB | 1.32 MiB/s, done.
Total 10 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To http://github.com/Parkeunyoung201824477/hw6.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

Parkeunyoung201824477 commit	
Makefile	commit
delete7.c	commit
hw7Main.c	commit
phone7.h	commit
print7.c	commit
register7.c	commit
search7.c	commit
sort7.c	commit

4. 논의 사항 (10점)

- 숙제를 하면서 새로이 알게 된 내용(수업 시간 이외의 내용),
- 숙제를 하는 중에 어려웠던 점 등을 기술

void*의 사용법을 깨달았고 어떻게 사용하는지 새롭게 알게 되었습니다. 지금 파일의 이름이 hw7Main.c로 되어 있는데 이것은 제가 과제번호를 헛갈려서 번호를 잘못 적었습니다. 이 점 양해 부탁드립니다. 그리고 저번 과제부터 Github을 사용해서 명령어들을 사용하면서 과제를 하니 재미있었습니다.