|  |
| --- |
| HW7. Function pointer and void pointer  (Open Source Programming 2019, Spring English class) **Class number: 060**  부산대학교 전기컴퓨터공학부 정보컴퓨터공학전공  201524582  HeeSeok Jeong(정희석)  Submission date: 2019-05-27 |

# Explanation for your codes (60 Points)

**(1) Explain of key variables: function pointers should be explained.**

A. This time, hw7main.c has function pointers array -> this replace switch – case condition, register.c has integer function pointer for expressing checkpassword function, delete.c has void function pointer for expressing deleteNSort function. And basically sort.c has integer function pointers for expressing contactCompare with cmp and contactSwap with swap.

**(2) Describe the main data structure**

A. I use structure type array structure, that named PhoneBook[MAX=50]; Because This Program reuse code of HW6, I used array in HW6. This array is composed of this structure.

**[**Name[10byte]**|**PhoneNumber[15byte]**]**|…|**[**Name[10byte]**|**PhoneNumber[15byte]**]**

**(3) Explain how to implement functions.**

- if void pointers are used, provide detailed explanations for how void pointers work

**- Phone.h**

텍스트이(가) 표시된 사진

자동 생성된 설명

* this header file contain basic variable of this Program, and function’s declaration. Implement PhoneBook’s base structure contact\_st, Define Contact array PhoneBook with extern variable, can store 50 Contacts. Char array password, int size store current array size and define several Constant with number for easily change number of elements

**1st. Main**

텍스트이(가) 표시된 사진

자동 생성된 설명

* Using static integer variables for counting service, integer variables for storing phonebook’s size, character array for password, and structure Contact array for phonebook with 50 elements.
* And Using void function pointers array -> replace switch-case condition, can execute function directly. -> pFuncs[0] = register, pFuncs[1] = print, pFuncs[2] = search, pFuncs[3] = delete, pFuncs[4] = sort
* First main request input password from user for using check process on register process.
* Using do-while loop, until service input not same with 6, each function continuously execute.

**2nd. Register.c** -> same with hw6’s register.c

텍스트이(가) 표시된 사진

자동 생성된 설명

* Using integer function pointer chk-> with no argument function pointer -> It will use for CheckPassword() function.
* this code role register. And assignment require check password process, I append checkPassword function, devide password checking process from registration process
* I use fgets function instead of gets function because of checking string size. And append Line 17, Line 20 -> prevent print \n charactor.
* Use while loop and if condition, compare password and check trial count

**3rd. print.c**

텍스트이(가) 표시된 사진

자동 생성된 설명

* Print array’s information. in printArray function, using void pointer \*arr -> just store memory address, this pointer don’t know type of this address, then we need to type cast for using with (Contact\*)vp+i, and this time printArray print void pointer’s address value.

**4th. Search.c** -> same with hw6’s search.c

텍스트이(가) 표시된 사진

자동 생성된 설명

* - I use variable for store string name for comparing name in PhoneBook, and index, using fgets store name for search, compare with strcmp function in string.h
* If find same string in phonebook, break loop and then index not same with size -> matched case! All loop complete –> index same with size -> that case mismatch

**5th. sort.c**

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

-> define integer function pointer cmp with arguments that constant void pointer, integer, integer, for contactCmpr function and integer function pointer swap with arguments that void pointer, integer, interger for contactSwap function.

-> sort() function is driver function, sortPhoneBook function is navigator function. -> void pointer just get address value and not know type of address’s value. -> I need to use type casting for using.

-> then I use type casting on contactCmpr with strcmp function in string.h, and strcmp return integer that differ between first argument and second argument, and standard is first argument.

-> strcmp >0 : first argument is bigger than(later than) second argument.

-> contactSwap use memcpy for all of contents swap each with using temp pointer with memory allocation.

**6th delete.c** -> same with hw6’s delete.c

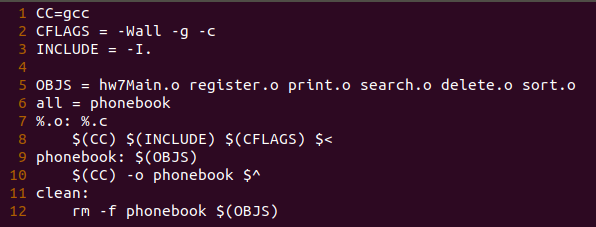
텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

<similar to search.c>

* I use static variable for store string name for comparing name in PhoneBook, and index (originally I define static int index but string.h has char index -> crash!). -> I devide delete and sort section from search and delete
* Define void function pointer type \*dnf for deleteNSort function.
* Pointer process express deleteNSort() function.
* Deletebyname -> search name same with input name
* If find name on phonebook -> execute deletensort , Else -> exit.
* deleteNSort() use strcpy’s copy feature and remove previous data, search matched index on static variable i, from that index to end of array, strcpy from next element to previous element and finally decrease size.

**(4) Describe makefile**



* same with hw6 and change phoneBookMain.o to hw7Main.o, and append sort.o
* using macro variable, and special macro variable
* each source code with .c file compile to object file first, and link with -o option and create executable file phonebook.

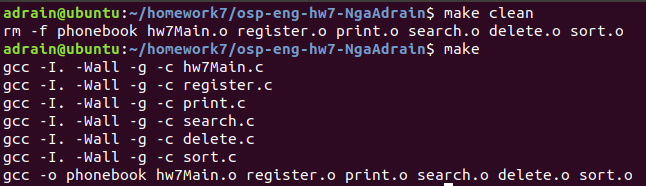
# Program Execution (10 points)

1. **Types of operating systems and compilers used**
2. OS : Ubuntu 18.04.2 LTS based on debian linux

GCC : gcc (Ubuntu 7.3.0-27ubuntu1~18.04) 7.3.0

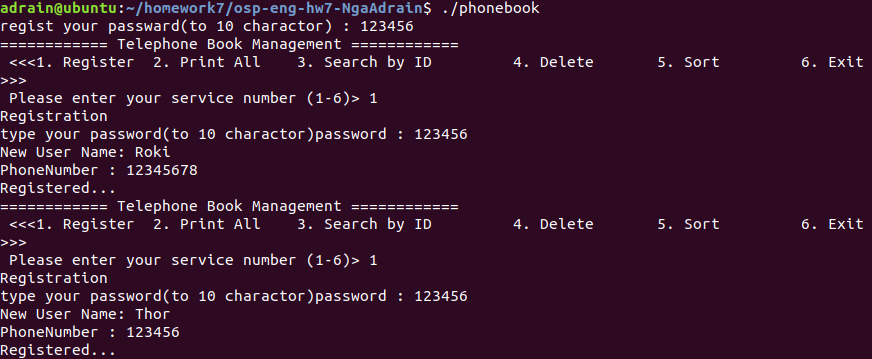
=> I can check Ubuntu version with ‘cat /etc/\*release’ and gcc version with ‘gcc –version’ command

1. **How to compile and execute the program**
2. I can compile with gcc and make makefile & use make command -> make executable file(named phonebook) -> execute ./phonebook



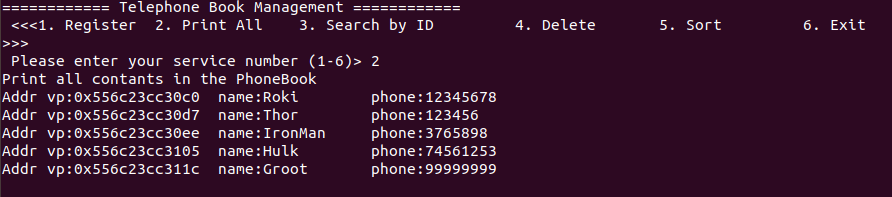
**(3) Include a screen capture for illustrating how the program works**

**1. register**



-> first execute program, register password, and check password. And input name, and phonenumber.

**2. print**



-> all input printed with memory address, name, phonenumber.

**3. search and delete**

스크린샷, 실외이(가) 표시된 사진

자동 생성된 설명

-> search with name, delete with name -> successfully executed.

**4. sort**

실외, 텍스트이(가) 표시된 사진

자동 생성된 설명

* I can sort with name’s character value lowercase.
* Address of array’s elements are not changed that mean each value of array’s element swap deep in array not swallow(array’s elements are changed each others)

# Github repository (20 points)

(1) include github commands for cloning, adding, committing and pushing

스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명

* Check status with ‘git status’ command and git add \*(with wild card) append all changes to track

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

* Commit file from track to local repository with message “0524 complete”
* Check status and push to remote repository.

(2) After pushing the source code and the makefile to your github directory, include the screen capture image of the Github repository

스크린샷이(가) 표시된 사진

자동 생성된 설명

* This is my repository, I can commit and push successfully.

# Discussion (10 points)

**- What you learned while doing your homework (contents other than class hours),**

A. I can learn about function pointer, and void pointer. And when I was modify README.md file, after I try to push my local repository, error occurred. That case try once pull request, retry push. It can solve problem. And function pointer can replace switch-case condition, I think this is good for machine, but if we use this way, less readability. And I think void pointer is similar to c++’s template class. Because we use void pointer for express unknown type’s address, In c++’s class use template class for express unknown class. Just my opinion. and I can review memory copy, string compare function.  
**- Describe difficulties during homework**

A. It was easy homework because I can reuse previous homework’s code, and sorting algorithm already determined with bubble sort. I wonder why select bubble sort? Not selection sort. Both of two algorithm are inefficient but we usually use selection sort. One difficulty thing is confusing of pointer’s type casting on string compare. But lecture note was very helpful.

Anyway, I think this homework is good for exercising void pointer, function pointer.

**Thanks for reading my report and sorry about my poor English.**