

HW8. Self-referential Structures

(Open Source Programming 2019, Spring English class) **Class number: 060**

부산대학교 전기컴퓨터공학부 정보컴퓨터공학전공

201524582

HeeSeok Jeong(정희석)

Submission date: 2019-06-03

1. Explanation for your codes (60 Points)

(1) Explain of key variables

- A. This time, Main data is consist of linked-list named Node. Node has Contact object and Node pointer for next node. It's like a Array but not continuous. This program has three Pointer variable that head is pointer of Start of Node , tail is pointer of end of Node. Phonebook is symbolic variable of linked-list of Contact. Explain this below.

(2) Describe the main data structure

- linked list for PhoneBook

- A. PhoneBook is a node that consist of pointer of next node and object of Contact object, Node pointer head and tail point each first of linked list(phonebook), end of linked list.

(Head) PhoneBook

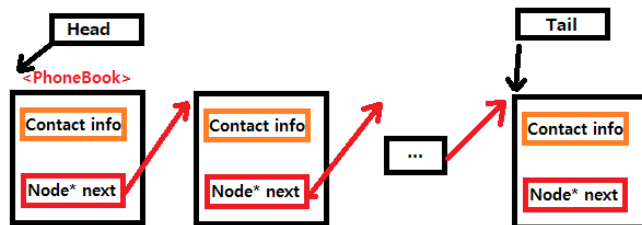
(Tail)

[Contact info|Node* next]->[Contact info|Node* next]->...->[Contact info|Node* next]

```
//define type of phonebook//
typedef struct contact_st
{
    char Name[N_MAX];
    char PhoneNumber[PN_MAX];
} Contact;

//defin type of Nodes//
typedef struct ContactNode
{
    Contact info;
    struct ContactNode *next;
} Node;

//define extern variable for using another files
//Nodes//
extern Node *PhoneBook;
extern Node *head;
extern Node *tail;
```



(3) Explain how to implement functions.

- if void pointers are used, provide detailed explanations for how void pointers work => Not Use
- > I delete codes that not using or unnecessary codes

- Phone.h

```

1 // array of sturcure
2 # ifndef _PHONE_H_
3 # define _PHONE_H_
4
5 //define constants for easy to change value
6 # define N_MAX 10
7 # define PN_MAX 15
8 # define PSWD_MAX 10
9
10 //function declaration//
11 void registerPhoneData();
12 void print();
13 void searchByName();
14 void deleteByName();
15 void sort();
16
17 //define type of phonebook//
18 typedef struct contact_st
19 {
20     char Name[N_MAX];
21     char PhoneNumber[PN_MAX];
22 } Contact;
23
24 //defin type of Nodes//
25 typedef struct ContactNode
26 {
27     Contact info;
28     struct ContactNode *next;
29 } Node;
30
31 //define extern variable for using another files
32 //Nodes//
33 extern Node *PhoneBook;
34 extern Node *head;
35 extern Node *tail;
36 //Others//
37 extern char password[PSWD_MAX];
38 extern int size; // store the actual numbers of PhoneBook
39
40 #endif

```

-> for line 23 same as homework 7's phone.h file. Declared function is driver function.

-> ContactNode is linked-list's Node.

-> PhoneBook is Symbolic variable of Node, head is first Node of PhoneBook, tail is end Node of PhoneBook.

- hw8Main.c

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "phone.h"
4
5 //variables//
6 static int count_service = 0; // Total number of service requests
7 int size = 0;
8 char password[PSWD_MAX];
9 Node *PhoneBook = NULL;
10 Node *head = NULL;
11 Node *tail = NULL;
12
13 //function pointers array//
14 void (* pFuncs[5])()={registerPhoneData, print, searchByName, deleteByName, sort};
15
16 //main function//
17 int main()
18 {
19     int service; // a variable for storing user's request
20     printf("regist your passward(to 10 character) : ");
21     fgets(password,PSWD_MAX,stdin);
22     do
23     {
24         printf("===== Telephone Book Management =====");
25         printf("\n <<<1. Register\t 2. Print All \t 3. Search by ID \t 4. Delete \t\n");
26         printf("5. Sort \t 6. Exit >>>\n");
27         printf(" Please enter your service number (1-6)> ");
28         scanf("%d", &service);
29         getchar(); //flush "\n"
30         if ( service > 0 && service <= 5 )
31         {
32             pFuncs[service-1]();
33         }
34         else if( service ==0 || service > 6)
35         {
36             printf("You choose a wrong service number\n");
37         }
38         count_service++;
39     } while (service != 6); // if Exit is not entered, the loop continues
40     printf("Thanks for Using! Bye Bye!\n");
41     return 0;
42 }
43

```

➔ This Code same as Homework 7's main source code. I'll skip this code.

- register.c

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include "phone.h"
5
6 typedef int (*chk)(); //define type of function pointer
7 static int try = 0;
8 static int checkPassword();
9 void initialization(Contact info);
10 void addNode(Contact info);
11
12 //register function//
13 void registerPhoneData(){
14     Contact temp;
15     chk chkpass = checkPassword; //do not need to use // but practicing
16     printf("Registration\n");
17     int check = chkpass(); //check == 0 no problem
18     if(check) return; //check == 1 can't access
19     printf("New User Name: ");
20     fgets(temp.Name,N_MAX,stdin); //contain \n
21     temp.Name[strlen(temp.Name)-1]='\0';//delete \n
22     printf("PhoneNumber : ");
23     fgets(temp.PhoneNumber,PN_MAX,stdin);
24     temp.PhoneNumber[strlen(temp.PhoneNumber)-1]='\0';//
25     if(PhoneBook == NULL)
26         initialization(temp);
27     else
28         addNode(temp);
29     size++;
30     printf("Registered...\n");
31 }
32
33 //checking password function//
34 int checkPassword(){
35     char temp[PSWD_MAX];
36     printf("type your password(to 10 charactor)");
37     printf("password : ");
38     fgets(temp,PSWD_MAX,stdin);
39     if(strcmp(temp,password)){
40         try++;
41         printf(">> Not Matched!!! \n");
42         printf("password : ");
43         fgets(temp,PSWD_MAX,stdin);
44         while((try<4)&&(strcmp(temp,password))){
45             switch(try){
46                 case 2:
47                     printf(">> Not Matched(twice)!!! \n");
48                     printf("password : ");
49                     fgets(temp,PSWD_MAX,stdin);
50                     break;
51                 case 3:
52                     printf(">> Not Matched(third)!!! \n");
53                     printf("password : ");
54                     fgets(temp,PSWD_MAX,stdin);
55                     break;
56             }
57             try++;
58         }
59         if(try==4){
60             printf(">> Your are not allow to access PhoneBook.\n");
61             return 1;
62         }
63     }
64     try = 0;
65     return 0;
66 }
67 //initialization function//
68 void initialization(Contact input){
69     PhoneBook = (Node*) malloc(sizeof(Node));
70     PhoneBook->info = input;
71     PhoneBook->next = NULL;
72     head = tail = PhoneBook;
73 }
74 //Node Append function//
75 void addNode(Contact info){
76     tail->next = (Node*) malloc(sizeof(Node));
77     tail = tail->next;
78     tail->info = info;
79     tail->next = NULL;
80 }

```

-> I append two function that initialization, and addNode.

-> basically registerPhoneData function and checkPassword function is same as homework7.

-> initialization (Contact input) function initialize Phonebook Node as first node. With Contact input object.

And first initialize process there's no any initilize, append head, tail's pointer point to PhoneBook Node.

-> addNode(Contact info) function append Node on tail with Contact info object, first. Memory allocate for next Node, second move to next Node(setting tail pointer), third add Contact object, last initialize next pointer to NULL..

- print.c

```
1 #include <stdio.h>
2 #include "phone.h"
3 /*****
4 ** Your code..
5 ** This function should be implemented in search.c
6 ** Please modify this code with your variables
7 *****/
8 void printNode(Node *inp);
9
10 void print()
11 {
12     printf("Print all contents in the PhoneBook\n");
13     printNode(PhoneBook);
14 }
15
16 //printNode//
17 void printNode(Node *inp){
18     Node* temp = inp;
19     while(temp != NULL){
20         printf("Addr vp:%p\t",temp);
21         printf("name:%s\t phone:%s\n",
22             temp->info.Name,temp->info.PhoneNumber);
23         temp = temp->next;
24     }
25 }
26
```

-> this is simple. first print Node inp, second move to next, loop this two process until Node arrive to last Node.

- search.c

```
1 #include <stdio.h>
2 #include <string.h>
3 #include "phone.h"
4
5 int searchNode(char* name);
6 //search function//
7 void searchByName(){
8
9     char temp[N_MAX];
10    int c = 0;
11    printf("Search by Name\n");
12    printf(">> Enter a name to search : ");
13    fgets(temp,N_MAX,stdin);
14    temp[strlen(temp)-1] = '\0';    //remove \n
15    c = searchNode(temp);
16    if(!c)
17        printf("Oops! %s is not in the PhoneBook!\n",temp);
18 }
19
20 int searchNode(char* name){
21     Node *temp = PhoneBook;
22     int check = 0;
23     while(temp!=NULL){
24         check = strcmp(name,temp->info.Name);
25         if(check != 0){
26             temp = temp->next;
27         }
28         else{
29             printf("%s\t %s\n",temp->info.Name,temp->info.PhoneNumber);
30             break;
31         }
32     }
33     if(check!=0)
34         return 0;
35     else
36         return 1;
37 }
```

-> searchByName similar to homework 7's function, append searchNode(char* name)

-> searchNode compare name with input, Node search start on Phonebook, strcmp compare name to Node's name, if name correct, print that Node's Contact information, if not correct go to next Node until arrive to Node's end, return 0, this case print incorrect information to user on searchByName function.

- delete.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include "phone.h"
5
6 static char temp[N_MAX];
7
8 typedef void (*dnf)(char*); //define function pointer
9 void deleteNode(char* name);
10
11 //delete function's driver function//
12 void deleteByName(){
13     dnf process = deleteNode;
14     printf("Delete by Name\n");
15     printf(">> Enter a name to delete : ");
16     fgets(temp,N_MAX,stdin);
17     temp[strlen(temp)-1]='\0'; //delete \n
18     process(temp);
19 }
20
21 void deleteNode(char* name){
22     Node *temp = PhoneBook;
23     Node *parent = temp;
24     int check = 0;
25     while(temp!=NULL){
26         check = strcmp(name,temp->info.Name);
27         if(check != 0){
28             parent = temp;
29             temp = temp->next;
30         }
31         else{
32             if(temp == head){
33                 head = temp->next;
34                 PhoneBook = head;
35             }
36             else if(temp == tail){
37                 parent->next = NULL;
38                 tail = parent;
39             }
40             else{
41                 parent->next = temp -> next;
42             }
43             parent = NULL;
44             free(temp);
45             temp = NULL;
46             printf("%s is deleted...\n",name);
47             break;
48         }
49     }
50     if(check!=0){
51         parent = NULL;
52         temp = NULL;
53         printf("Oops! %s is not in the PhoneBook!\n",name);
54     }
55 }
```

-> using function pointer type *dnf with char* argument for deleteNode function.

-> deleteNode get name for deleted. And search Node from head (PhoneBook) to end of Node.

-> compare name to Node's name, if incorrect, move to next Node, if correct, check Node's position, if head position or tail position, relocation this Node, and previous Node's next pointer connect to removed Node's next Node. after this process, free memory of this Node.

- sort.c

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include "phone.h"
5
6 void sortNode();
7 void sort()
8 {
9     printf("Sort fuction is called\n");
10    printf("Before sorting\n");
11    print();
12    sortNode();
13    printf("After sorting\n");
14    print();
15 }
16 //=====//
17 void sortNode(){
18     Node* start;
19     Node* temp;
20     Node* next;
21     Contact tempCont;
22     start = temp = head;
23     next = head->next;
24     while(start!=NULL){
25         while(next!=NULL){
26             if(strcmp(temp->info.Name,next->info.Name)>0){
27                 tempCont = temp->info;
28                 temp->info = next->info;
29                 next->info = tempCont;
30             }
31             temp = next;
32             next = next->next;
33         }
34         start = start -> next;
35         temp = head;
36         next = temp->next;
37     }
38 }
39 //=====//

```

-> I use Bubble sort because Homework 7 use this sort algorithm. And It is easy for implement.

-> using three Pointer of Node, and two while loop, while start move to end Node, check two Node's name with strcmp

-> if Next Node's name's Alphabet(based on ASCII) smaller than Current Node's Alphabet, swap two Node's Contact object to ascending order.

-> There's No change of Node's pointer. And Node pointer Move to end of Node. repeat this process.

(4) Describe makefile

```

1 CC=gcc
2 CFLAGS = -Wall -g -c
3 INCLUDE = -I.
4
5 OBJS = hw8Main.o register.o print.o search.o delete.o sort.o
6 all = phonebook
7 %.o: %.c
8     $(CC) $(INCLUDE) $(CFLAGS) $<
9 phonebook: $(OBJS)
10    $(CC) -o phonebook $^
11 clean:
12    rm -f phonebook $(OBJS)

```

- ➔ same with hw8 and change hw7Main.o to hw8Main.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.
- ➔ SIMILAR TO HOMEWORK 7

2. Program Execution (10 points)

(1) Types of operating systems and compilers used

A. 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

(2) How to compile and execute the program

- A. I can compile with gcc and make makefile & use make command -> make executable file (named phonebook) -> execute ./phonebook

```
adrain@ubuntu:~/homework8/osp-eng-hw8-NgaAdrain$ make clean
rm -f phonebook hw8Main.o register.o print.o search.o delete.o sort.o
adrain@ubuntu:~/homework8/osp-eng-hw8-NgaAdrain$ make
gcc -I. -Wall -g -c hw8Main.c
gcc -I. -Wall -g -c register.c
gcc -I. -Wall -g -c print.c
gcc -I. -Wall -g -c search.c
gcc -I. -Wall -g -c delete.c
gcc -I. -Wall -g -c sort.c
gcc -o phonebook hw8Main.o register.o print.o search.o delete.o sort.o
```

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

1. Register and Print

```
adrain@ubuntu:~/homework8/osp-eng-hw8-NgaAdrain$ ./phonebook
register your password(to 10 character) : 123456
===== Telephone Book Management =====
<<<1. Register  2. Print All  3. Search by ID      4. Delete      5. Sort      6. Exit >>>
Please enter your service number (1-6)> 1
Registration
type your password(to 10 character)password : 123456
New User Name: IronMan
PhoneNumber : 010-1234-5678
Registered...
===== Telephone Book Management =====
<<<1. Register  2. Print All  3. Search by ID      4. Delete      5. Sort      6. Exit >>>
Please enter your service number (1-6)> 1
Registration
type your password(to 10 character)password : Hulk
>> Not Matched!!!
password : 123456
New User Name: Nulk
PhoneNumber : 010-3245-1648
Registered...
===== Telephone Book Management =====
<<<1. Register  2. Print All  3. Search by ID      4. Delete      5. Sort      6. Exit >>>
Please enter your service number (1-6)> 2
Print all contents in the PhoneBook
Addr vp:0x562fcf141a80  name:IronMan  phone:010-1234-5678
Addr vp:0x562fcf141ab0  name:Nulk     phone:010-3245-1648
```

- ➔ I can execute register process, and checking password, print with pointer's address, name, phone number.

2. Search and delete

```
===== Telephone Book Management =====
<<<1. Register  2. Print All  3. Search by ID      4. Delete      5. Sort      6. Exit >>>
Please enter your service number (1-6)> 3
Search by Name
>> Enter a name to search : IronMain
Oops! IronMain in not in the PhoneBook!
===== Telephone Book Management =====
<<<1. Register  2. Print All  3. Search by ID      4. Delete      5. Sort      6. Exit >>>
Please enter your service number (1-6)> 4
Delete by Name
>> Enter a name to delete : Nulk
Nulk is deleted...
===== Telephone Book Management =====
<<<1. Register  2. Print All  3. Search by ID      4. Delete      5. Sort      6. Exit >>>
Please enter your service number (1-6)> 3
Search by Name
>> Enter a name to search : IronMan
IronMan  010-1234-5678
```

- ➔ I can search with name, delete with name, if name not exist on Contact, print error message

※ For check delete correction on try to delete Nulk Object, this exist on upper screenshot and not exist on below screen shot. (Because Nulk is mistyping of Hulk, I try to delete)

3. Sort

```
===== Telephone Book Management =====
<<<1. Register  2. Print All  3. Search by ID      4. Delete      5. Sort      6. Exit >>>
Please enter your service number (1-6)> 5
Sort fuction is called
Before sorting
Print all contants in the PhoneBook
Addr vp:0x562fcf141a80  name:IronMan      phone:010-1234-5678
Addr vp:0x562fcf141ab0  name:AntMan      phone:010-1594-8786
Addr vp:0x562fcf141ae0  name:Doctor     phone:010-1400-0605
Addr vp:0x562fcf141b10  name:Hulk       phone:010-3546-8787
Addr vp:0x562fcf141b40  name:Loki       phone:018-3546-4942
Addr vp:0x562fcf141b70  name:Spider     phone:010-9658-7523
After sorting
Print all contants in the PhoneBook
Addr vp:0x562fcf141a80  name:AntMan      phone:010-1594-8786
Addr vp:0x562fcf141ab0  name:Doctor     phone:010-1400-0605
Addr vp:0x562fcf141ae0  name:Hulk       phone:010-3546-8787
Addr vp:0x562fcf141b10  name:IronMan     phone:010-1234-5678
Addr vp:0x562fcf141b40  name:Loki       phone:018-3546-4942
Addr vp:0x562fcf141b70  name:Spider     phone:010-9658-7523
```

➔ After delete Nulk, I append AntMan, Doctor, Hulk, Loki, Spider, and execute sorting process, Phonebook successfully sorted to ascending order of ASCII rule.

3. Github repository (20 points)

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

```
adrain@ubuntu:~/homework8/osp-eng-hw8-NgaAdrain$ git add *
adrain@ubuntu:~/homework8/osp-eng-hw8-NgaAdrain$ 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)

        modified:   delete.c
        new file:   delete.o
        modified:   hw8Main.c
        new file:   hw8Main.o
        new file:   node.c
        modified:   phone.h
        new file:   phonebook
        modified:   print.c
        new file:   print.o
        modified:   register.c
        new file:   register.o
        modified:   search.c
        new file:   search.o
        modified:   sort.c
        new file:   sort.o
```

➔ First, add all changed, created file on tract with command 'git add *'(wild card), check status of git with 'git status'

```
adrain@ubuntu:~/homework8/osp-eng-hw8-NgaAdrain$ git commit -m "19/06/01 source code complete"
[master 3f5ee39] 19/06/01 source code complete
15 files changed, 255 insertions(+), 212 deletions(-)
rewrite delete.c (66%)
create mode 100644 delete.o
create mode 100644 hw8Main.o
create mode 100644 node.c
create mode 100755 phonebook
rewrite print.c (77%)
create mode 100644 print.o
create mode 100644 register.o
create mode 100644 search.o
rewrite sort.c (93%)
create mode 100644 sort.o
```

➔ Second, Commit all file on track to local repository with command git commit, and give -m option, add comment.


```

adrain@ubuntu:~/homework8/osp-eng-hw8-NgaAdrain$ git push
Username for 'https://github.com': NgaAdrain
Password for 'https://NgaAdrain@github.com':
Counting objects: 17, done.
Compressing objects: 100% (17/17), done.
Writing objects: 100% (17/17), 23.13 KiB | 4.63 MiB/s, done.
Total 17 (delta 8), reused 0 (delta 0)
remote: Resolving deltas: 100% (8/8), completed with 3 local objects.
To https://github.com/pnusystemprog/osp-eng-hw8-NgaAdrain
b25e8b0..3f5ee39 master -> master

```

➔ Finally, push all file from local repository to remote repository, with login github.

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

8 commits
1 branch
0 releases
2 contributors

Branch: master
New pull request
Create new file
Upload files
Find File
Clone or download

| NgaAdrain Append Log of homework 8 on README.md | | Latest commit c7474fb 1 minute ago |
|---|---------------------------------------|------------------------------------|
| HeeSeok Jeong_201524582.txt | Initialization on 19/05/27 | 6 days ago |
| README.md | Append Log of homework 8 on README.md | 1 minute ago |
| delete.c | 19/06/01 source code complete | 22 hours ago |
| delete.o | 19/06/01 source code complete | 22 hours ago |
| hw8Main.c | 19/06/01 source code complete | 22 hours ago |
| hw8Main.o | 19/06/01 source code complete | 22 hours ago |
| makefile | Initialization on 19/05/27 | 6 days ago |
| node.c | 19/06/01 source code complete | 22 hours ago |
| phone.h | 19/06/01 source code complete | 22 hours ago |
| phonebook | 19/06/01 source code complete | 22 hours ago |
| print.c | 19/06/01 source code complete | 22 hours ago |
| print.o | 19/06/01 source code complete | 22 hours ago |
| register.c | 19/06/01 source code complete | 22 hours ago |
| register.o | 19/06/01 source code complete | 22 hours ago |
| search.c | 19/06/01 source code complete | 22 hours ago |
| search.o | 19/06/01 source code complete | 22 hours ago |
| sort.c | 19/06/01 source code complete | 22 hours ago |
| sort.o | 19/06/01 source code complete | 22 hours ago |

README.md

Open source programming HW8 @PNU

➔ This is result of commit to github.

4. Discussion (10 points)

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

A. This homework can exercise making linked list replace array, using linked list that make, add, delete, search, sort. And I think this homework unnecessary to use void pointer, because of using linked list, that need to use pointer, and each pointer has pointer type. That is why I not use void pointer this time. Last I can exercise too memory allocation and deallocation with malloc, free function.

- Describe difficulties during homework

A. This is not a big problem, there's no template code, when accept this homework first, I feel confuse what should I do first? And first, copy my homework7's code to this project. Second, modify makefile. Third create node.c file for node operation. But I think node operation's declaration, and implement can be implemented on each source code(print.c, register.c, ... etc). and remove node.c and after finish function implementation, remove code that about homework6, 7 and not using code.

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