HW9. File I/O

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

부산대학교 전기컴퓨터공학부 정보컴퓨터공학전공

201524582

HeeSeok Jeong(정희석)

Submission date: 2019-06-12

1. Explanation for your codes (60 Points)

(1) Explain of key variables

A. This homework's reuse hw6,7,8, and using File I/O, I use file pointer, and implement file I/O in register.c.

From hw6, I use new Structure named Contact and use extern variable for Node pointer, from hw7, I use function pointer's array for replace switch-case condition, and from hw8, I can replace Phonebooks structure from array to Linked-List that named Node and use that structure in hw9main.c. In this homework,

Anyway, this program's key variables are sturcture Node's pointer PhoneBook, and head, tail for store data in hw9main.c, and for file I/O execution, FILE's pointer inp and out in register.c

(2) Describe the main data structure

- linked list for PhoneBook => same as homework 8

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.

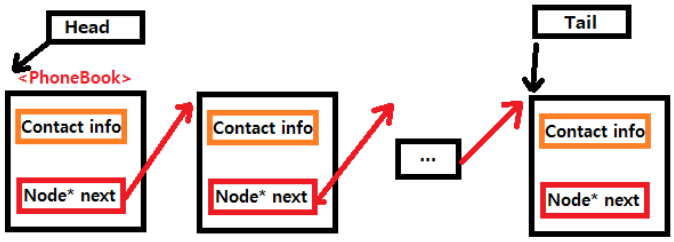
(Head) PhoneBook (Tail)

[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.

- Include explanations for File I/O

- phone.h

```c
// array of sturcure
# ifndef _PHONE_H_
# define _PHONE_H_

//define constants for easy to change value
# define N_MAX 10
# define PN_MAX 15
# define PSWD_MAX 10

//function declaration//
void registerFromFile();
void writeToFile();
void registerPhoneData();
void print();
void searchByName();
void deleteByName();
void sort();


//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;
//Anothers//
extern char password[PSWD_MAX];
extern int size;  // store the actual numbers of PhoneBook

#endif
```

=> add registerFromfile() function and writeToFile() function declaration for using that function on hw9main.c and implement on register.c

=> define N_MAX for name's length, PN_MAX for phonenumber's length, and PSWD_MAX for password's length

=> function declaration for using that functions on hw9main.c's service serving, and implementing that functions in each source file register.c, print.c, search.c, delete,c, sort.c

=> define contact_st structure type storing character array Name, and PhoneNumber for data store and rename to Contact using typedef.

=> define ContactNode structure type storing object structure data Contact, and for linked-list structure pointer next, and rename to Name using typedef.

=> define extern variable for using several source codes.

- hw9main.c

```c
#include <stdio.h>
#include <stdlib.h>
#include "phone.h"

//reference of FI/O
//file I/O ->
//r,w,r+,w+,rb,wb => read, write, readNwrite, writeNread(overwrite), write binary)
//fwrite(const void *arr, size_t size, size_t count, FILE *stream)
//fread(void *arr, size_t size, size_t count, FILE *stream)

//variables//
static int count_service = 0;   // Total number of service requests
int size = 0;
char password[PSWD_MAX];
Node *PhoneBook = NULL;
Node *head = NULL;
Node *tail = NULL;

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

//main function//
int main()
{
    int service;        // a variable for storing user's request
    printf("regist your passward(to 10 charactor) : ");
    fgets(password,PSWD_MAX,stdin);
    //========//
    //File Read//
//  temp = fopen("phone.dat","wb");
//  fclose(temp);
    registerFromFile();
    //========//
```

=>write this part(reference) for me

=> initialize and define variables, pointers, and array for password

=> function pointer's array for replace switch-case condition

=> register password for register process.(password is typing-error)

=> add registerFromfile() function on line : 32 => implemented in register.c for reading data from phone.dat and storing to Node* PhoneBook (initialize PhoneBook)

```
34      do
35      {
36          printf("============ Telephone Book Management ============");
37          printf("\n <<<1. Register\t 2. Print All \t 3. Search by ID \t 4. Delete \t");
38          printf(" 5. Sort \t 6. Exit >>>\n");
39          printf(" Please enter your service number (1-6)> ");
40          scanf("%d", &service);
41          getchar(); //flush "\n"
42          if ( service > 0 && service <= 5 )
43          {
44              pFuncs[service-1]();
45          }
46          else if( service ==0 || service > 6)
47          {
48              printf("You choose a wrong service number\n");
49          }
50          count_service++;
51      } while (service != 6); // if Exit is not entered, the loop continues
52      //========//
53      //File Write//
54      writeToFile();
55      //========//
56      printf("Thanks for Using! Bye Bye!\n");
57      return 0;
58  }
```

=> showing selection to user, get input from user using scanf function, getchar do role of flushing buffer, execute function for each process until user select exit service.

=>add writeToFile() function on line : 54

=> also implemented in register.c for writing data from current (appended, modified, deleted) datas to phone.dat file.(update)

- register.c

=> changed part

```
 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 from file// //allModify!//
13  void registerFromFile(){
14      FILE * inp;
15      Contact buffer;
16      inp = fopen("phone.dat","rb");
17      if(!inp)
18          return;
19      else if(fread(&buffer,sizeof(Contact),1,inp)==1)
20      {//find file not exist or empty//
21          initialization(buffer);
22      }
23      while(fread(&buffer,sizeof(Contact),1,inp)==1){
24          addNode(buffer);
25      }
26      fclose(inp);
27  }
28
29  //register to file//
30  void writeToFile(){
31      Node* temp = head;
32      FILE * out;
33      out = fopen("phone.dat","wb");
34      while(temp!=NULL){
35          fwrite(&(temp->info),sizeof(Contact),1,out);
36          temp = temp->next;
37      }
38      fclose(out);
39  }
```

=> define function pointer for exercise

=> using static variable and static function for password checking process.

=> pre-declaration for using addNode function, and initialization function.

=> add registerFromFile function, this function will use when start program first, this function is executed, read phonebook data from phone.dat(binary file) with fread function, and store data to Phonebook linked list. With initialization, addNode function.

=> add writeToFile function, this function will use for when finish program, this function is executed, from PhoneBook's head to tail, all data will store to phone.dat file with fwrite function.

=> in this code, in line : 16, File pointer initialize with fopen function, file name : phone.dat, and "rb" option(read binary), while fread can read 1-line it can store information to Phonebook.

=> if file not exist inp just try to open file, and close file. There's no problem.

=> and in line : 33, File pointer also initialize with fopen function same as line : 16 but different option "wb"(write binary).

=> Unchanged part

```
41 //register function//
42 void registerPhoneData(){
43     Contact temp;
44     chk chkpass = checkPassword; //do not need to use // but practicing
45     printf("Registration\n");
46     int check = chkpass(); //check == 0 no problem
47     if(check) return;           //check == 1 can't access
48     printf("New User Name: ");
49     fgets(temp.Name,N_MAX,stdin);   //contain \n
50     temp.Name[strlen(temp.Name)-1]='\0';//delete \n
51     printf("PhoneNumber : ");
52     fgets(temp.PhoneNumber,PN_MAX,stdin);
53     temp.PhoneNumber[strlen(temp.PhoneNumber)-1]='\0';//
54     if(PhoneBook == NULL)
55         initialization(temp);
56     else
57         addNode(temp);
58     size++;
59     printf("Registered...\n");
60 }
61
62 //checking password function//
63 int checkPassword(){
64     char temp[PSWD_MAX];
65     printf("type your password(to 10 charactor)");
66     printf("password : ");
67     fgets(temp,PSWD_MAX,stdin);
68     if(strcmp(temp,password)){
69         try++;
70         printf(">> Not Matched!!! \n");
71         printf("password : ");
72         fgets(temp,PSWD_MAX,stdin);
73         while((try<4)&&(strcmp(temp,password))){
74             switch(try){
75             case 2:
76                 printf(">> Not Matched(twice)!!! \n");
77                 printf("password : ");
78                 fgets(temp,PSWD_MAX,stdin);
79                 break;
80             case 3:
81                 printf(">> Not Matched(third)!!! \n");
82                 printf("password : ");
83                 fgets(temp,PSWD_MAX,stdin);
84                 break;
85             }
```

=> 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 50, Line 53 -> prevent print ₩n charactor.

=> Use while loop and if condition, compare password and check trial count

- print.c => not changed from homework8's code

```
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 vairables
7  ************************/
8  void printNode(Node *inp);
9
10 void print()
11 {
12     printf("Print all contants 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
```

-> Call printNode function with startnode

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

- search.c => not changed from homework8's code

```c
#include <stdio.h>
#include <string.h>
#include "phone.h"

int searchNode(char* name);
//search function//
void searchByName(){

    char temp[N_MAX];
    int c = 0;
    printf("Search by Name\n");
    printf(">> Enter a name to search : ");
    fgets(temp,N_MAX,stdin);
    temp[strlen(temp)-1] = '\0';    //remove \n
    c = searchNode(temp);
    if(!c)
        printf("Oops! %s in not in the PhoneBook!\n",temp);
}

int searchNode(char* name){
    Node *temp = PhoneBook;
    int check = 0;
    while(temp!=NULL){
        check = strcmp(name,temp->info.Name);
        if(check != 0){
            temp = temp->next;
        }
        else{
            printf("%s\t %s\n",temp->info.Name,temp->info.PhoneNumber);
            break;
        }
    }
    if(check!=0)
        return 0;
    else
        return 1;
}
```

->I use variable for store string name for comparing name in PhoneBook using fgets store name for search, compare with strcmp function in string.h

-> 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, if move to Node's end, return 0, this case print incorrect information to user on searchByName function.

- sort.c => not changed from homework8's code

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "phone.h"

void sortNode();
void sort()
{
    printf("Sort fuction is called\n");
    printf("Before sorting\n");
    print();
    sortNode();
    printf("After sorting\n");
    print();
}
//================================================//
void sortNode(){
    Node* start;
    Node* temp;
    Node* next;
    Contact tempCont;
    start = temp = head;
    next = head->next;
    while(start!=NULL){
        while(next!=NULL){
            if(strcmp(temp->info.Name,next->info.Name)>0){
                tempCont = temp->info;
                temp->info = next->info;
                next->info = tempCont;
            }
            temp = next;
            next = next->next;
        }
        start = start -> next;
        temp = head;
        next = temp->next;
    }
}
//=========================================//
```

-> I use Bubble sort because 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.

- delete.c => not changed from homework8's code

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "phone.h"

static char temp[N_MAX];

typedef void (*dnf)(char*);//define function pointer
void deleteNode(char* name);

//delete function's driver function//
void deleteByName(){
    dnf process = deleteNode;
    printf("Delete by Name\n");
    printf(">> Enter a name to delete : ");
    fgets(temp,N_MAX,stdin);
    temp[strlen(temp)-1]='\0';  //delete \n
    process(temp);
}

void deleteNode(char* name){
    Node *temp = PhoneBook;
    Node *parent = temp;
    int check = 0;
    while(temp!=NULL){
        check = strcmp(name,temp->info.Name);
        if(check != 0){
            parent = temp;
            temp = temp->next;
        }
        else{
            if(temp == head){
                head = temp->next;
                PhoneBook = head;
            }
            else if(temp == tail){
                parent->next = NULL;
                tail = parent;
            }
            else{
                parent->next = temp -> next;
            }
            parent = NULL;
            free(temp);
            temp = NULL;
            printf("%s is deleted...\n",name);
            break;
        }
    }
    if(check!=0){
        parent = NULL;
        temp = NULL;
        printf("Oops! %s is not in the PhoneBook!\n",name);
    }
}
```

-> 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.

(4) Describe makefile

```
CC=gcc
CFLAGS = -Wall -g -c
INCLUDE = -I.

OBJS = hw9Main.o register.o print.o search.o delete.o sort.o
all = phonebook
%.o: %.c
    $(CC) $(INCLUDE) $(CFLAGS) $<
phonebook: $(OBJS)
    $(CC) -g -o phonebook $^
clean:
    rm -f phonebook $(OBJS)
```

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. <-

<Macro variable>

Setting of gcc -> CC, CFLAGS, INCLUDE

Object Files -> OBJS

All -> select main execution(phonebook)

%.o: %.c -> compile each c source code to same name of object file.

phonebook : link all OBJS to executable file named phonebook

clean : remove all object file and executable file.

(using like "~/> make clean")

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:~/homework9/osp-eng-hw9-NgaAdrain$ make clean
rm -f phonebook hw9Main.o register.o print.o search.o delete.o sort.o
adrain@ubuntu:~/homework9/osp-eng-hw9-NgaAdrain$ make
gcc -I. -Wall -g -c hw9Main.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 -g -o phonebook hw9Main.o register.o print.o search.o delete.o sort.o
```

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

1st.phone.dat file => this file created by program. (when I test program, this file created)

```
1 Alice^@^@^@b+010-4563-7852^@^@AntMan^@^@b;010-2323-4545^@^@Hulk^@^@^@^@b»010-2345-6789^@^@IronMan^@^@»010-1234-5678^@^@
```

2nd.check phone.dat's information and register

```
adrain@ubuntu:~/homework9/osp-eng-hw9-NgaAdrain$ ./phonebook
regist your password(to 10 charactor) : 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)> 2
Print all contants in the PhoneBook
Addr vp:0x5601fa0e8cc0  name:Alice       phone:010-4563-7852
Addr vp:0x5601fa0e8cf0  name:AntMan      phone:010-2323-4545
Addr vp:0x5601fa0e8d20  name:Hulk        phone:010-2345-6789
Addr vp:0x5601fa0e8d50  name:IronMan     phone:010-1234-5678
============ 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 charactor)password : 123456
New User Name: Reika
PhoneNumber : 010-4352-1325
Registered...
```

3rd.search information on 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)> 3
Search by Name
>> Enter a name to search : Reika
Reika    010-4352-1325
```

4th. Delete information on 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)> 2
Print all contants in the PhoneBook
Addr vp:0x5601fa0e8cc0  name:Alice       phone:010-4563-7852
Addr vp:0x5601fa0e8cf0  name:AntMan      phone:010-2323-4545
Addr vp:0x5601fa0e8d20  name:Hulk        phone:010-2345-6789
Addr vp:0x5601fa0e8d50  name:IronMan     phone:010-1234-5678
Addr vp:0x5601fa0e7cb0  name:Reika       phone:010-4352-1325
============ 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 : Reika
Reika 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)> 2
Print all contants in the PhoneBook
Addr vp:0x5601fa0e8cc0  name:Alice       phone:010-4563-7852
Addr vp:0x5601fa0e8cf0  name:AntMan      phone:010-2323-4545
Addr vp:0x5601fa0e8d20  name:Hulk        phone:010-2345-6789
Addr vp:0x5601fa0e8d50  name:IronMan     phone:010-1234-5678
```

5th. Sort information on PhoneBook

```
New User Name: Emily
PhoneNumber : 010-0108-2456
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 contants in the PhoneBook
Addr vp:0x5601fa0e8cc0  name:Alice       phone:010-4563-7852
Addr vp:0x5601fa0e8cf0  name:AntMan      phone:010-2323-4545
Addr vp:0x5601fa0e8d20  name:Hulk        phone:010-2345-6789
Addr vp:0x5601fa0e8d50  name:IronMan     phone:010-1234-5678
Addr vp:0x5601fa0e7cb0  name:Emily       phone:010-0108-2456
============ 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:0x5601fa0e8cc0  name:Alice       phone:010-4563-7852
Addr vp:0x5601fa0e8cf0  name:AntMan      phone:010-2323-4545
Addr vp:0x5601fa0e8d20  name:Hulk        phone:010-2345-6789
Addr vp:0x5601fa0e8d50  name:IronMan     phone:010-1234-5678
Addr vp:0x5601fa0e7cb0  name:Emily       phone:010-0108-2456
After sorting
Print all contants in the PhoneBook
Addr vp:0x5601fa0e8cc0  name:Alice       phone:010-4563-7852
Addr vp:0x5601fa0e8cf0  name:AntMan      phone:010-2323-4545
Addr vp:0x5601fa0e8d20  name:Emily       phone:010-0108-2456
Addr vp:0x5601fa0e8d50  name:Hulk        phone:010-2345-6789
Addr vp:0x5601fa0e7cb0  name:IronMan     phone:010-1234-5678
```

6th. Delete information on PhoneBook(that originally existed on phone.dat) and write to file with exit.

```
============ 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 : Hulk
Hulk 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)> 2
Print all contants in the PhoneBook
Addr vp:0x5601fa0e8cc0  name:Alice       phone:010-4563-7852
Addr vp:0x5601fa0e8cf0  name:AntMan      phone:010-2323-4545
Addr vp:0x5601fa0e8d20  name:Emily       phone:010-0108-2456
Addr vp:0x5601fa0e7cb0  name:IronMan     phone:010-1234-5678
============ 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)> 6
Thanks for Using! Bye Bye!
```

Final. Phone.dat's result

```
1 Alice^@^@^@b+010-4563-7852^@^@AntMan^@^@b;010-2323-4545^@^@Emily^@^@^@b^[010-0108-2456^@^@IronMan^@^@»010-1234-5678^@^@
```

3. Github repository (20 points)

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

<add file on track waiting for commit> with "git add *" command

```
adrain@ubuntu:~/homework9/osp-eng-hw9-NgaAdrain$ git add *
adrain@ubuntu:~/homework9/osp-eng-hw9-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)

        new file:   delete.o
        modified:   hw9Main.c
        new file:   hw9Main.o
        modified:   phone.dat
        new file:   phonebook
        new file:   print.o
        modified:   register.c
        new file:   register.o
        new file:   search.o
        new file:   sort.o
```

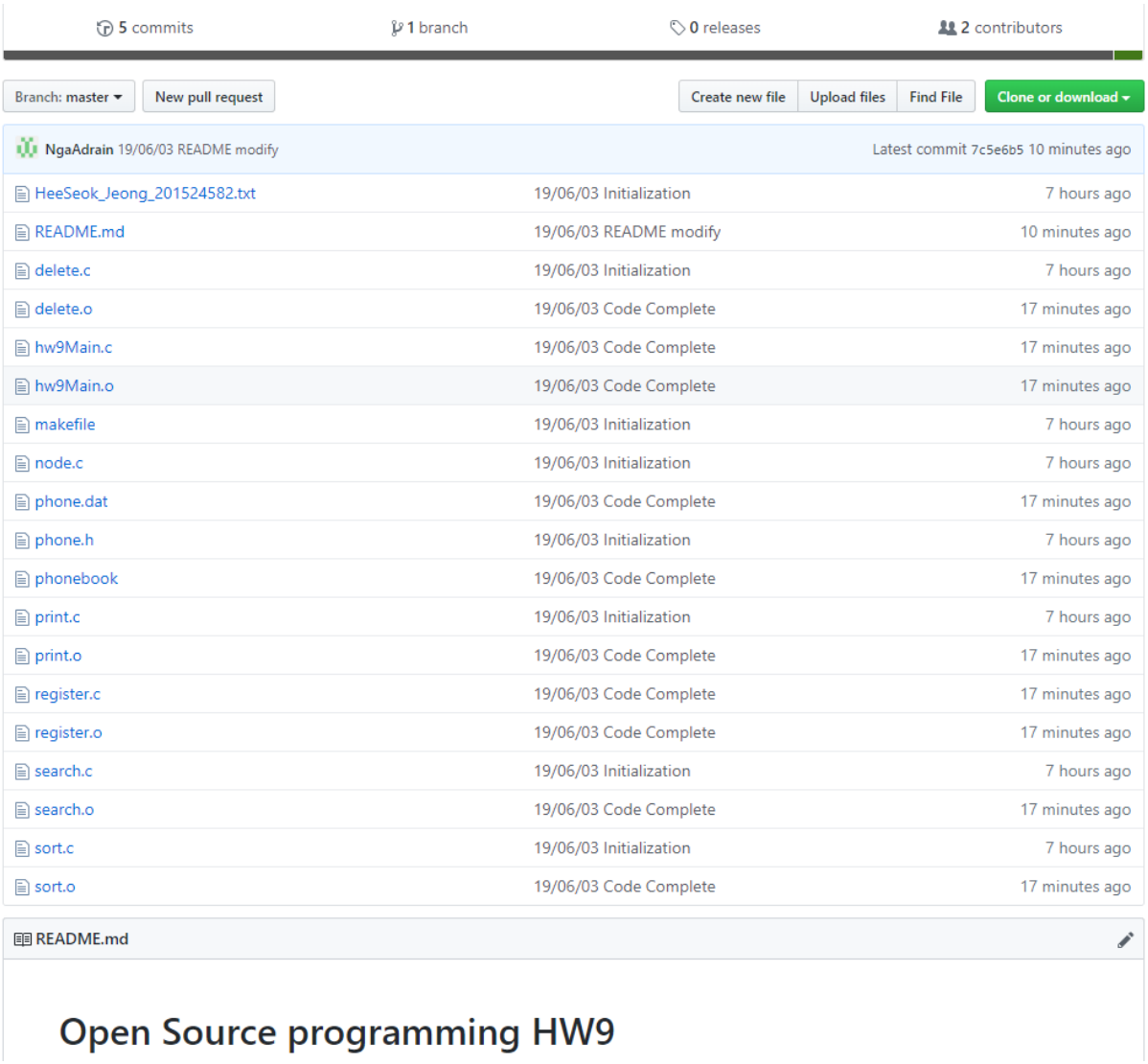
<commit tracked file to local repository with comment("git commit" command)>

```
adrain@ubuntu:~/homework9/osp-eng-hw9-NgaAdrain$ git commit -m "19/06/03 Code Complete"
[master eef0900] 19/06/03 Code Complete
 10 files changed, 7 insertions(+), 7 deletions(-)
 create mode 100644 delete.o
 create mode 100644 hw9Main.o
 create mode 100755 phonebook
 create mode 100644 print.o
 create mode 100644 register.o
 create mode 100644 search.o
 create mode 100644 sort.o
adrain@ubuntu:~/homework9/osp-eng-hw9-NgaAdrain$ git push
Username for 'https://github.com': NgaAdrain
Password for 'https://NgaAdrain@github.com':
Counting objects: 12, done.
Compressing objects: 100% (12/12), done.
Writing objects: 100% (12/12), 22.20 KiB | 5.55 MiB/s, done.
Total 12 (delta 6), reused 0 (delta 0)
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
To https://github.com/pnusystemprog/osp-eng-hw9-NgaAdrain
   29d73e1..eef0900  master -> master
```

<push from local repository to remote repository with "git push" command>

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

<after pushing, this is my repository for homework 9, I can check successfully commited>



4. Discussion (10 points)

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

A. I can learn usage of file pointer, and practice modifying my program like update with additional feature. And I can review from my program while write this report. From hw6, I can review usage of structure, from hw7, usage of void pointer and function pointer (but in hw9, not use void pointer), from hw8, usage of Node with Linked-List, in hw8's Node, and Linked-list concept leads me can solve another class's algorithm problem!. And in hw9, I can practice using file pointer with binary file Input/Output with fopen, fread, fwrite function. While I do several homework, the most impressed feature is memory allocation. Before register this class, I usually use c++, in this language, not use malloc, I just study malloc 3-years ago, after then no use of malloc, just use new function. Also, structure replaced to class I c++, this class, and this several homework is good for reminding my c language usability and understanding about memory control. Thanks for these homework for professor, also thanks for TA.

- Describe difficulties during homework

A. This time, I have file open problem, when phone.dat file not exist, program terminated with error, I tried to change file open option, then occur segmentation fault. Finally I can solve this problem with compare fread function's return value. In register.c, I try to open file with read binary option, I expect when try to open file but file not exist -> file pointer is null and return and continue, but program not execute my expect, then I can append condition with compare fread function's return value. Except this difficulty, this homework also easy homework. I think why this homework is easy, my opinion is that I can use source code that I made last time. This is very good for exercising, I think.

Anyway, this is final homework, thanks for reading my report, and sorry for my poor English.

한 학기간 고생 많으셨습니다.

Thanks for your effort during this spring semester.