

Report

오픈소스소프트웨어개발입문

-Linked list-

내용	목차
표지	1
사진1 : person.h	2
사진2 : process.c : Enroll() 함수	3
사진3 : process.c : List()함수	4
사진4 : process.c : Search() 함수	4
사진5 : process.c : Delete()함수	5-6
사진6 : process.c : Quit() 함수	6-7
사진7 : process.c : selection()함수	7
사진8 : main.c	8
사진9 : makefile	8
사진10 : makefile 실행	9
사진11 : Manage_system 실행	10
사진12 : Manage_system 실행	11
사진13 : Manage_system 실행2	12

오픈소스소프트웨어개발입문(062)

학번 : 201924451

이름 : 김태훈

```
minchocoin@ubuntu: ~/opensource/HW10
File Edit View Search Terminal Help
#ifndef _PERSON_H_
#define _PERSON_H_
typedef struct Person{
    char name[20];
    char phone[15];
    struct Person* next;
}person;
void Enroll();
void List();
person* Search();
person* Delete();
void Quit();
void selection();
#endif
```

<사진1 : person.h 파일>

person.h에는 struct person과 각 함수의 prototype이 정의되어있다. struct person의 멤버는 이름을 저장할 char name[20]과 전화번호를 저장할 phone[15] 그리고 다음 struct person의 주소이다. 그리고 person.h가 중복되어 들어가더라도 함수 prototype과 struct person이 중복 정의되지 않도록 #ifndef~#endif를 넣었다.

```
minchocoin@ubuntu: ~/opensource/HW10
File Edit View Search Terminal Help
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include "person.h"
person *head=NULL;
person *tail=NULL;

void Enroll()
{
    person* p = (person*)malloc(sizeof(person));
    assert(p!=NULL);
    printf("Name : ");
    scanf("%s",p->name);
    printf("Phone : ");
    scanf("%s",p->phone);
    if(head==NULL){
        head = p;
        tail = p;
        p->next=NULL;
    }
    else{
        p->next = head;
        head = p;
    }
    printf("Enroll Success!\n");
}
```

<사진2 ;process.c : Enroll() 함수>

Enroll함수는 이름과 전화번호를 새로 등록하는 함수이다. 먼저 person *p에 메모리를 할당한 다음 이름(p->name)과 전화번호(p->phone)을 입력받은 후 head 포인터가 NULL이라면 아무것도 저장된 것이 없다는 뜻이므로 head와 tail이 새로 입력받은 person 구조체 포인터 p를 가리키도록 하고 p의 다음 person 구조체의 포인터(p->next)는 NULL로 한다.

만약 head포인터가 NULL이 아니면 저장된 것이 1개 이상 있다는 뜻이므로 p의 다음 구조체 포인터를 head로 하고, head 포인터에 p를 대입하여 p를 새롭게 head로 만든다. 이렇게 되면 새로 입력받은 구조체는 head가 되고, 원래 head 포인터는 p->next가 된다.

```

void List()
{
    person* cur = head;
    printf("---List---\n");
    while(cur!=NULL){
        printf("%s\t%s\n",cur->name,cur->phone);
        cur=cur->next;
    }
}

```

<사진3 :process.c : List() 함수>

List 함수는 저장된 person 구조체를 보여주는 함수이다. head 포인터부터 시작하여 cur포인터가 NULL이 될때까지(다음 person 구조체가 없을때 까지) cur=cur->next를 반복하여 다음 구조체로 이동하면서 저장된 이름과 전화번호를 보여준다.

```

person* Search(){
    char Sname[20];
    printf("Name : ");
    scanf("%s",Sname);

    person* cur = head;
    while(cur!=NULL){
        if(strcmp(cur->name,Sname)==0)
            break;
        cur=cur->next;
    }
    if(cur==NULL)
        printf("%s Could not find\n",Sname);
    return cur;
}

```

<사진4 : process.c : Search() 함수>

Search 함수는 저장된 person 구조체 중에서 찾고자 하는 person 구조체를 찾아주는 함수이다. 이름을 입력받고 head 포인터부터 시작하여 cur포인터가 NULL이 될때까지(다음 person 구조체가 없을때 까지) cur=cur->next를 반복하여 다음 구조체로 이동하면서 저장된 이름과 입력받은 이름을 strcmp 함수로 비교하여 검색하고, 찾지 못했다면 “Could not find”메시지를 보여주고 NULL을 리턴한다. 찾았다면 그 이름이 들어있는 person 구조체 포인터를 반환한다.

```

person* Delete(){
    person *p = NULL;
    p=Search();
    if(p!=NULL){
        if(p==head){
            if(head!=NULL){
                if(head->next)
                    head = head->next;
                else{
                    head = NULL;
                    tail = NULL;
                }
            }
        }
        else if(p == tail){
            person* cur, *prev=NULL;
            cur = head;
            if(tail!=NULL){
                while(cur->next!=NULL){
                    prev = cur;
                    cur=cur->next;
                }
                if(prev!=NULL)
                    prev->next=NULL;
            }
            else{
                head=NULL;
                tail=NULL;
            }
        }
        else{
            person* cur, *prev=NULL;
            cur = head;
            while((cur->next)!=p->next){
                prev = cur;
                cur=cur->next;
            }
            if(prev!=NULL){
                prev->next=p->next;
                if(p==tail)
                    tail=prev;
            }
        }
    }
    return p;
}

```

<사진5 : process.c : Delete() 함수>

Delete 함수는 사용자로부터 이름을 입력받아 그 이름과 일치하는 person 구조체를 삭제하는 함수이다.

먼저 Search 함수를 호출하여 반환된 포인터(=삭제해야하는 구조체 포인터)를 p에 저장하고 p가 head, tail, head도 tail도 아닌 포인터일때 삭제하는 방법을 다르게 하였다.

1)삭제해야할 포인터가 head인 경우

이 경우 head와 head->next가 NULL 함수가 아닌지 확인하고, 아니라면 head에 head->next를 대입하여 head->next가 새롭게 head가 되도록 하고, p를 반환한다.

2)삭제해야할 포인터가 tail인 경우

이 경우 head 포인터부터 시작하여 cur->next가 NULL일때 까지 계속 cur=cur->next를 반복하여 다음 포인터로 이동하고 prev=cur을 통해 cur->next의 이전 person 구조체 포인터를 prev에 저장한다. 이렇게 되면 cur->next==NULL 일때 tail의 이전 포인터를 prev 가 가지게 된다. 따라서 prev->next=NULL을 해주게 되면 prev포인터가 새롭게 tail 포인터가 된다.

3)삭제해야할 포인터가 head, tail 모두 아닌 경우

이 경우 head 포인터부터 시작하여 cur->next가 NULL일때 까지 계속 cur=cur->next를 반복하여 다음 포인터로 이동하고 prev=cur을 통해 cur->next의 이전 person 구조체 포인터를 prev에 저장한다. 이렇게 되면 cur->next==p->next 일때 p의 이전 포인터를 prev 가 가지게 된다. 따라서 prev->next=p->next를 해주게 되면 p의 이전 포인터와 다음 포인터가 연결된다.

```
}
void Quit(){
    while(head!=NULL){
        if((head->next)!=NULL){
            person *temp = head;
            head=head->next;
            free(temp);
        }
        else{
            free(head);
            head=NULL;
            tail=NULL;
        }
    }
}
```

<사진6 : process.c : Quit() 함수>

head->next가 NULL이 아닐때(이 경우 저장되어있는 person 구조체가 2개 이상이다.) 기존 head 포인터를 temp에 저장하고 head에 head->next를 대입하여 head->next가 새롭게 head가 되도록하고 기존 head 포인터(=temp)는 해제하여 준다.

만약 head->next가 NULL 일때(이 경우 저장되어있는 person 구조체가 1개이다.) head포인터를 해제하고 head포인터를 NULL로 바꾼다. 이것을 head가 NULL일때 까지 계속 반복한다.

```
void selection(){
    int select=0;
    person* p;
    while(select!=5){
        printf("\n[1] Enroll\t[2] List\t[3] Search\t[4]Delete\t[5]Quit\n");
        printf("Selection : ");
        scanf("%d",&select);

        switch(select){
            case 1:
                Enroll();
                break;
            case 2:
                List();
                break;
            case 3:
                p = Search();
                if(p!=NULL){
                    printf("%s %s\n",p->name,p->phone);
                }
                break;
            case 4:
                p= Delete();
                if(p!=NULL){
                    printf("%s Delete Success\n",p->name);
                    free(p);
                }
                break;
            case 5:
                Quit();
                printf("Program Quit.\n");
                break;
            default:
                printf("Wrong Number\n");
                break;
        }
    }
}
```

<사진7 : selection() 함수>

select가 5가 아닐동안 select값을 받아 1,2,3,4,5 default 에 따라 처리한다.

1일 경우 Enroll 함수를 호출하여 2일 경우 List 함수를 호출하고, 3일 경우 Search 함수를 호출하고 반환된 포인터가 가리키는 person 구조체의 멤버를 출력한다.

4일 경우 Delete() 함수를 호출하고 반환된 포인터가 가리키는 person 구조체의 멤버를 출력하고 그 포인터를 해제한다.

5일 경우 Quit() 함수를 호출한다. 1,2,3,4,5외 다른 번호일 경우 Wrong Number를 출력하고 다시 숫자를 입력받는다.


```
minchocoin@ubuntu: ~/opensource/HW10
File Edit View Search Terminal Help
#include <stdio.h>
#include "person.h"
int main(int argc, char *argv[]) {
    printf("---Enrollment Manage System---\n");
    selection();
    return 0;
}
```

<사진8 : main.c : main 함수>

“---Enrollment Manage System---”를 출력하고 selection() 함수를 호출 후 종료한다.

```
minchocoin
File Edit View Search Terminal Help
#makefile for linked list person
CC=gcc
CFLAGS = -I. -Wall
OBJ = main.o process.o
all:Manage_system

Manage_system:$(OBJ)
    $(CC) $(CFLAGS) -o $@ $^

main.o : main.c person.h
    $(CC) -c -o $@ $(CFLAGS) $<
process.o : process.c person.h
    $(CC) -c -o $@ $(CFLAGS) $<

clean:
    rm Manage_system $(OBJ)
```

<사진9 : makefile>

main.c와 person.h로부터 main.o를 만들고, process.c와 person.h로부터 process.o를 만들고 process.o와 main.o로부터 Manage_system 실행파일을 만든다.


```
minchocoin@ubuntu:
File Edit View Search Terminal Help
minchocoin@ubuntu:~/opensource/HW10$ ls
main.c makefile person.h process.c
minchocoin@ubuntu:~/opensource/HW10$ make all
gcc -c -o main.o -I. -Wall main.c
gcc -c -o process.o -I. -Wall process.c
gcc -I. -Wall -o Manage_system main.o process.o
minchocoin@ubuntu:~/opensource/HW10$ make clean
rm Manage_system main.o process.o
minchocoin@ubuntu:~/opensource/HW10$
```

<사진10 : makefile 실행>

```
minchocoin@ubuntu: ~/opensource/HW10
File Edit View Search Terminal Help
minchocoin@ubuntu:~/opensource/HW10$ ./Manage_system
---Enrollment Manage System---

[1] Enroll      [2] List      [3] Search      [4]Delete      [5]Quit
Selection : 1
Name : Seosu
Phone : 01012345678
Enroll Success!

[1] Enroll      [2] List      [3] Search      [4]Delete      [5]Quit
Selection : 1
Name : Yeoju
Phone : 01099997777
Enroll Success!

[1] Enroll      [2] List      [3] Search      [4]Delete      [5]Quit
Selection : 2
---List---
Yeoju  01099997777
Seosu  01012345678

[1] Enroll      [2] List      [3] Search      [4]Delete      [5]Quit
Selection : 3
Name : Seosu
Seosu 01012345678

[1] Enroll      [2] List      [3] Search      [4]Delete      [5]Quit
Selection : 3
Name : Yamco
Yamco Could not find

[1] Enroll      [2] List      [3] Search      [4]Delete      [5]Quit
Selection : 4
Name : Seosu
Seosu Delete Success

[1] Enroll      [2] List      [3] Search      [4]Delete      [5]Quit
Selection : 2
---List---
Yeoju  01099997777

[1] Enroll      [2] List      [3] Search      [4]Delete      [5]Quit
Selection : 4
Name : Yamco
Yamco Could not find

[1] Enroll      [2] List      [3] Search      [4]Delete      [5]Quit
Selection : 5
```

<사진11 : Manage_system 실행 : 사진12로 이어짐>

```
minchocoin@ubuntu: ~/opensource/HW10
File Edit View Search Terminal Help

[1] Enroll      [2] List      [3] Search    [4]Delete    [5]Quit
Selection : 1
Name : Seosu
Phone : 01012345678
Enroll Success!

[1] Enroll      [2] List      [3] Search    [4]Delete    [5]Quit
Selection : 1
Name : Yeoju
Phone : 01099997777
Enroll Success!

[1] Enroll      [2] List      [3] Search    [4]Delete    [5]Quit
Selection : 2
---List---
Yeoju  01099997777
Seosu  01012345678

[1] Enroll      [2] List      [3] Search    [4]Delete    [5]Quit
Selection : 3
Name : Seosu
Seosu 01012345678

[1] Enroll      [2] List      [3] Search    [4]Delete    [5]Quit
Selection : 3
Name : Yamco
Yamco Could not find

[1] Enroll      [2] List      [3] Search    [4]Delete    [5]Quit
Selection : 4
Name : Seosu
Seosu Delete Success

[1] Enroll      [2] List      [3] Search    [4]Delete    [5]Quit
Selection : 2
---List---
Yeoju  01099997777

[1] Enroll      [2] List      [3] Search    [4]Delete    [5]Quit
Selection : 4
Name : Yamco
Yamco Could not find

[1] Enroll      [2] List      [3] Search    [4]Delete    [5]Quit
Selection : 5
Program Quit.
minchocoin@ubuntu:~/opensource/HW10$
```

<사진12 : Manage_system 실행>

```
minchocoin@ubuntu: ~/opensource/HW10
File Edit View Search Terminal Help
minchocoin@ubuntu:~/opensource/HW10$ ./Manage_system
---Enrollment Manage System---

[1] Enroll      [2] List      [3] Search     [4]Delete     [5]Quit
Selection : 1
Name : Seosu
Phone : 01012341234
Enroll Success!

[1] Enroll      [2] List      [3] Search     [4]Delete     [5]Quit
Selection : 1
Name : Yeoju
Phone : 01099997777
Enroll Success!

[1] Enroll      [2] List      [3] Search     [4]Delete     [5]Quit
Selection : 1
Name : Minsu
Phone : 01055553333
Enroll Success!

[1] Enroll      [2] List      [3] Search     [4]Delete     [5]Quit
Selection : 2
---List---
Minsu  01055553333
Yeoju  01099997777
Seosu  01012341234

[1] Enroll      [2] List      [3] Search     [4]Delete     [5]Quit
Selection : 3
Name : Yeoju
Yeoju 01099997777

[1] Enroll      [2] List      [3] Search     [4]Delete     [5]Quit
Selection : 4
Name : Yeoju
Yeoju Delete Success

[1] Enroll      [2] List      [3] Search     [4]Delete     [5]Quit
Selection : 2
---List---
Minsu  01055553333
Seosu  01012341234

[1] Enroll      [2] List      [3] Search     [4]Delete     [5]Quit
Selection : 5
Program Quit.
minchocoin@ubuntu:~/opensource/HW10$
```

<사진13 : Manage_system 실행2>