

친구관리 프로그램(C언어)

백한별

2020 년 2 월 3 일

목 차

1. 프로젝트 개요	2
2. 프로젝트 설명	2
2.1 개발환경	2
2.2 개발 기간	2
3. 화면 설계	3
4. 프로그램 구현 및 상세내용	8
5. 결과	17

1. 프로젝트 개요

C언어를 활용하여 친구 관리 프로그램을 구현하였다.

- 나만의 헤더파일을 만들어 `#include`, `#define`, 구조체, 함수 선언 부분을 분리하였다.
- 파일 입출력 소스를 통해 텍스트 파일에 데이터를 저장, 관리할 수 있도록 만들었다.

2. 프로젝트 설명

2.1 개발환경

IDE : Visual Studio 2019

Programming Language : C언어

2.2 개발 기간

2020.01.20. ~ 2020.02.03.

3. 화면 설계

```
===== [ 메뉴 ] =====  
1. 친구 추가  
2. 친구 삭제  
3. 친구 검색  
4. 친구 목록 보기  
5. 저장 및 나가기  
=====  
번호 선택:
```

< 그림 3-1 >

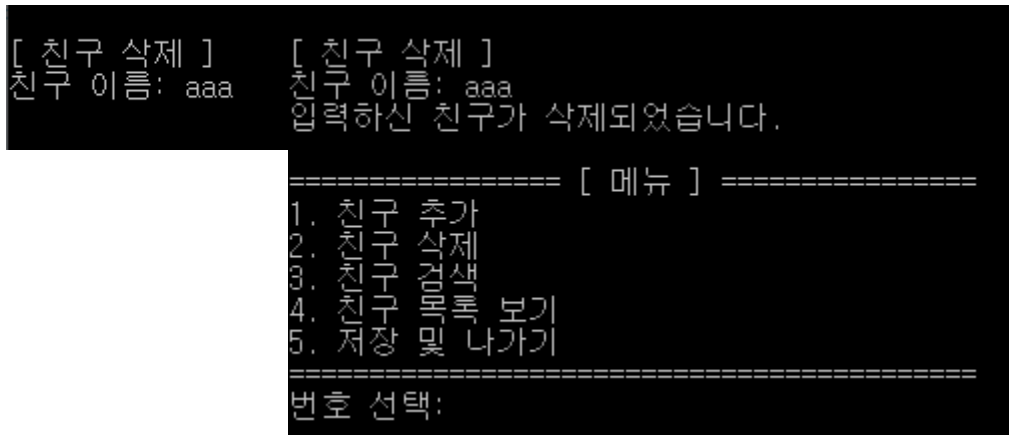
처음 프로그램 구동 시 그림 3-1과 같이 친구 관리에 필요한 메뉴를 출력하고 선택할 수 있도록 구현하였다.

```
[ 친구 추가 ]  
이름: aaa  
나이: 11  
전화번호: 010-1111-1111  
정보가 저장되었습니다.  
===== [ 메뉴 ] =====  
1. 친구 추가  
2. 친구 삭제  
3. 친구 검색  
4. 친구 목록 보기  
5. 저장 및 나가기  
=====  
번호 선택:
```

< 그림 3-2 >

< 그림 3-3 >

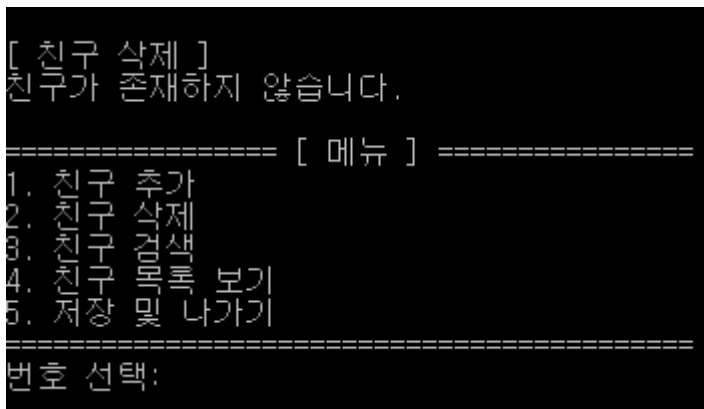
1번 메뉴 선택 시 그림 3-1에서 출력되었던 메뉴는 사라지고 그림 3-2와 같이 이름, 나이, 전화번호를 입력하고 모두 입력하고 나면 정보가 저장되었다는 메시지와 함께 다른 메뉴를 선택할 수 있도록 그림 3-1의 메뉴가 다시금 출력된다.



< 그림 3-4 >

< 그림 3-5 >

2번 메뉴 선택 시 그림 3-4와 같이 친구 이름을 입력할 수 있는 화면으로 넘어가며, 등록했던 친구의 이름을 입력하게되면 그림 3-5와 같이 삭제되었다는 문구와 함께 다시 메뉴가 나온다.



< 그림 3-6 >

등록한 친구가 한명도 없을 때 2번을 누르게되면 그림 3-6과 같이 친구가 존재하지 않는다는 문구가 출력되고 다시 메뉴를 입력할 수 있도록 한다.

```

[ 친구 검색 ]
친구를 찾을 수 없습니다.

===== [ 메뉴 ] =====
1. 친구 추가
2. 친구 삭제
3. 친구 검색
4. 친구 목록 보기
5. 저장 및 나가기
=====
번호 선택:

```

< 그림 3-7 >

등록된 친구가 없을 때 3번을 누르게되면 친구를 찾을 수 없다는 문구가 출력되며 다시 메뉴를 선택할 수 있도록 한다.

```

[ 친구 검색 ]
친구 이름:

```

< 그림 3-8 >

등록된 친구가 있을 경우 그림 3-8과 같이 친구 이름을 검색할 수 있도록 한다.

```

[ 친구 검색 ]
친구 이름: aaa
이름 : aaa
나이 : 11
전화번호 : 010-1111-1111

===== [ 메뉴 ] =====
1. 친구 추가
2. 친구 삭제
3. 친구 검색
4. 친구 목록 보기
5. 저장 및 나가기
=====
번호 선택:

```

< 그림 3-9 >

등록되어있는 친구의 이름을 검색하면 그림 3-9와 같이 입력했던 친구의 정보가 출력된다.

```

[ 친구 목록 보기 ]
이름 : aaa
나이 : 11
전화번호 : 010-1111-1111
저장된 모든 친구가 표시되었습니다.

===== [ 메뉴 ] =====
1. 친구 추가
2. 친구 삭제
3. 친구 검색
4. 친구 목록 보기
5. 저장 및 나가기
=====
번호 선택:

```

< 그림 3-10 >

4번을 누르면 현재까지 등록되어있는 모든 친구의 정보가 표시된다.

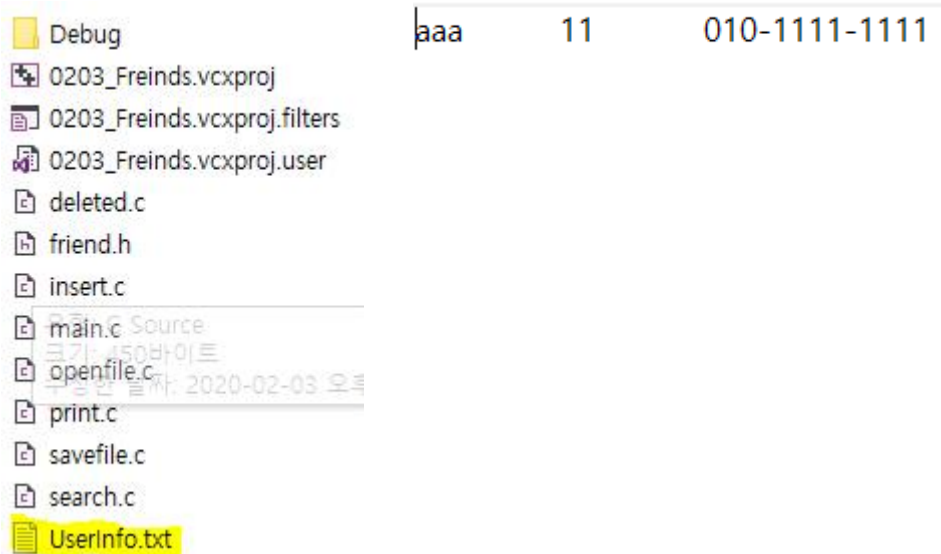
```

데이터가 모두 저장되었습니다.
D:\백한별\02언어\0203_Freinds\Debug\0203_Freinds.exe(프로세스 27656개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

```

< 그림 3-11 >

5번을 누르면 데이터가 모두 저장되었다는 문구와 함께 프로그램이 종료된다.



< 그림 3-12 >

< 그림 3-13 >

정상적으로 프로그램이 종료되고나면 그림 3-12처럼 텍스트파일이 존재하지 않았을 경우 생성되며, 그림 3-13과 같이 프로그램에서 입력했던 친구 정보가 저장된다.

4. 프로그램 구현 및 상세내용

4.1 Header File

Source Code : friend.h
<pre>#define _CRT_SECURE_NO_WARNINGS #include<stdio.h> #include<string.h> #include<Windows.h> #define MAX_COUNT 5 typedefstruct Friend { char name[30]; char age[5]; char phonNo[30]; }User; int openfile(User* ptr, int* num); void insert(User* ptr, int* num); int deleted(User* ptr, int* num); int search(User* ptr, int* num); void print(User* ptr, int* num); int savefile(User* ptr, int* num);</pre>

위 코드는 헤더파일을 가져온 것이다.

소스파일에서 사용할 scanf의 오류를 방지하고, 출력소스를 위한 stdio.h, 문자열소스들을 사용하기위한 string.h, 콘솔화면을 깨끗이 만들어주기위한 windows.h 등과 같은 헤더파일들을 참조해주었다.

그리고 프로그램 구성에 필요한 최대값을 정의하고 구조체를 만들어 소스 구성을 더욱 쉽게하였으며, 함수 선언부를 모두 friend.h파일에 넣어 다른 소스파일에 일일이 적지 않도록 분리하였다.

4.2 Source File

Source Code : main.c

```
#include "friend.h"
int main() {
    int person = 0, input;
    User user[MAX_COUNT];
    openfile(user, &person);
    while (1) {
        printf("===== [ 메뉴 ] =====\n");
        printf("1. 친구 추가\n");
        printf("2. 친구 삭제\n");
        printf("3. 친구 검색\n");
        printf("4. 친구 목록 보기\n");
        printf("5. 저장 및 나가기\n");

        printf("===== \n");
        printf("번호 선택: ");
        scanf("%d", &input);
        rewind(stdin);
        system("cls");
        if (input == 1) {
            printf("\n[ 친구 추가 ]\n");
            insert(user, &person);
        }
        elseif (input == 2) {
            printf("\n[ 친구 삭제 ]\n");
            deleted(user, &person);
        }
        elseif (input == 3) {
            printf("\n[ 친구 검색 ]\n");
            search(user, &person);
        }
    }
}
```

```

    }
    elseif (input ==4) {
        printf("Wn[ 친구 목록 보기 ]Wn");
        print(user, &person);
    }
    elseif (input ==5) {
        savefile(user, &person);
        return0;
    }
    else
        printf("오류가 발생했습니다. 다시 시도해주세요.Wn");
}
return0;
}

```

위 코드는 메인 함수를 가져온 것이다.

메뉴를 선택할 input값과 인원을 카운트할 person값을 선언하였으며, person값은 0으로 초기화하였다.

구조체 User의 배열을 헤더파일에서 정의한 최대값만큼 지정하였다.

while 무한 반복문을 통하여 오류가 발생하거나 자체적으로 종료할 수 있게끔 구성하였다.

Source Code : openfile.c

```

#include"friend.h"
//저장된 데이터 불러오기
int openfile(User* ptr, int* num) {
    int state;
    //char temp;
    FILE* fp = fopen("UserInfo.txt", "rt");
    if (fp ==NULL) {
        //printf("File Open Error!Wn");
        return1;
    }
    //파일 데이터 구조체에 입력

```

```

        while (1) {
            fscanf(fp, "%sWt%sWt%s", ptr[*num].name, ptr[*num].age,
ptr[*num].phonNo);
            if (feof(fp) !=0)
                break;
            (*num)+ +;
        }
        //fclose 종료 오류 발생 시 안내 후 종료
        state = fclose(fp);
        if (state !=0) {
            printf("File Close Error!Wn");
            return 1;
        }
        return 0;
    }
}

```

위 코드는 텍스트 파일을 생성하며, 저장된 데이터값을 불러오는 파일 입력 부분의 소스 코드이다.

fclose를 통해 파일을 종료할 때 오류가 발생하면 0이 아닌 다른값을 반환하기 때문에 이를 확인할 수 있도록 임의의 state 변수를 선언하였다.

파일의 형태는 읽기쓰기 모두 가능하며, 파일이 존재하지않을 때 생성까지할 수 있도록 rt모드로 하였다.

while 반복문을 통해 파일에 저장된 데이터를 구조체 배열에 저장할 수 있도록 구성하였다.

Source Code : insert.c

```

#include "friend.h"
//친구 추가
void insert(User* ptr, int* num) {
    //MAX_COUNT 확인
    if (*num < MAX_COUNT) {
        printf("이름: ");
        scanf("%s", ptr[*num].name);
        rewind(stdin);
    }
}

```

```

        printf("나이: ");
        scanf("%s", ptr[*num].age);
        rewind(stdin);
        printf("전화번호: ");
        scanf("%s", ptr[*num].phonNo);
        rewind(stdin);
        (*num)++;
        printf("정보가 저장되었습니다.WnWn");
    }
    else
        printf("친구 목록이 모두 차있습니다.WnWn");
}

```

위 코드는 친구를 등록하는 소스코드이다.

최대값을 확인하며 데이터를 입력할 수 있도록 구현하였다.

Source Code : deleted.c

```

#include"friend.h"
//친구 목록 삭제
int deleted(User* ptr, int* num) {
    char name[30];
    int i, j;
    //친구 목록에 저장된 값이 있을 경우
    if (*num > 0) {
        printf("친구 이름: ");
        scanf("%s", name);
        rewind(stdin);
        //이름 비교
        for (i = 0; i < MAX_COUNT; i++) {
            //문자열 비교 strcmp 사용
            if (strcmp(name, ptr[i].name) == 0) {
                (*num)--;
                printf("입력하신 친구가 삭제되었습니다.WnWn");
                //목록에 공간이 남아있을때
                if (i != MAX_COUNT - 1) {

```

```

        for (j = i; j < MAX_COUNT; j++) {
            //문자열 복사 strcpy 사용
            strcpy(ptr[j].name,      ptr[i
+ 1].name);

            strcpy(ptr[j].age,      ptr[i
+ 1].age);

            strcpy(ptr[j].phonNo,    ptr[i
+ 1].phonNo);

        }
        //user 구조체 마지막 NULL로 변경
        *ptr[MAX_COUNT - 1].name =NULL;
        *ptr[MAX_COUNT - 1].age =NULL;
        *ptr[MAX_COUNT - 1].phonNo =NULL;
    }
    //목록 가득 찼을때
    else {
        *ptr[MAX_COUNT - 1].name =NULL;
        *ptr[MAX_COUNT - 1].age =NULL;
        *ptr[MAX_COUNT - 1].phonNo =NULL;
    }
}

}

}
//친구 정보 찾을 수 없을때
else {
    printf("친구가 존재하지 않습니다.WnWn");
    return0;
}
}

```

위 코드는 친구를 삭제하는 코드이다.

친구 목록에 데이터가 저장되어있을 경우 if문을 통해 친구를 삭제할 수 있는 단계로 넘어간다.

입력한 친구이름과 데이터에 등록되어있는 친구이름을 비교하기위해 strcmp코

드를 사용하였다.

Source Code : search.c
<pre>#include"friend.h" //친구 검색 int search(User* ptr, int* num) { char name[30]; int i; //저장된 친구 있을때 if (*num >0) { printf("친구 이름: "); scanf("%s", name); rewind(stdin); for (i =0; i < MAX_COUNT; i++) { //문자열 비교 strcmp 참일때 return 0 if (!strcmp(name, ptr[i].name)) { printf("이름 : %s\n", ptr[i].name); printf("나이 : %s\n", ptr[i].age); printf("전화번호 : %s\n\n", ptr[i].phonNo); return0; } } } else { printf("친구를 찾을 수 없습니다.\n\n"); return0; } }</pre>

위 코드는 친구를 검색할 때 사용하는 함수이다.

문자열을 비교하는 strcmp는 참일 경우 0을 반환하기 때문에 조건에 이를 부정 해주며 시작하였다.

Source Code : print.c
<pre>#include"friend.h" //모든 친구 목록 불러오기 void print(User* ptr, int* num) {</pre>

```

int i =0;
if (*num >0) {
    for (i =0; i <*num; i++) {
        printf("이름 : %s\n", ptr[i].name);
        printf("나이 : %s\n", ptr[i].age);
        printf("전화번호 : %s\n", ptr[i].phonNo);
    }
    printf("저장된 모든 친구가 표시되었습니다.\n\n");
}
else
    printf("저장된 친구가 없습니다.\n\n");
}

```

위 코드는 등록된 친구 목록을 모두 불러오는 함수이다.

Source Code : savefile.c

```

#include "friend.h"
//입력한 친구데이터 파일에 저장
int savefile(User* ptr, int* num) {
    if (*num >0) {
        int i, state;
        FILE* fp = fopen("UserInfo.txt", "wt");
        //fopen 함수 오류발생 시 프로그램 종료
        if (fp ==NULL) {
            printf("File Open Error!\n");
            return 1;
        }
        //입력된 데이터 파일에 줄바꿈 구분 저장
        for (i =0; i <*num; i++) {
            fprintf(fp, "%s\t%s\t%s", ptr[i].name, ptr[i].age,
ptr[i].phonNo);

            fputc('\n', fp);
        }
        //fclose 오류 시 안내 후 프로그램 종료
        state = fclose(fp);
        if (state !=0) {

```



```

        printf("File Close Error!Wn");
        return 1;
    }
    printf("데이터가 모두 저장되었습니다.Wn");
    return 0;
}
else {
    printf("프로그램을 종료합니다.Wn");
    return 0;
}
}

```

위 코드는 입력한 데이터들을 모두 파일에 저장하는 함수이다.
fputc를 통해 파일에 줄바꿈을 같이 넣어주었다.

5. 결과

openfile에서 소스 파일에서 파일을 여는 과정에서 오류가 발생하는 것을 발견하여 향후 수정이 필요하다.