

11장. 실전프로젝트

박종혁 교수

UCS Lab

Tel: 970-6702

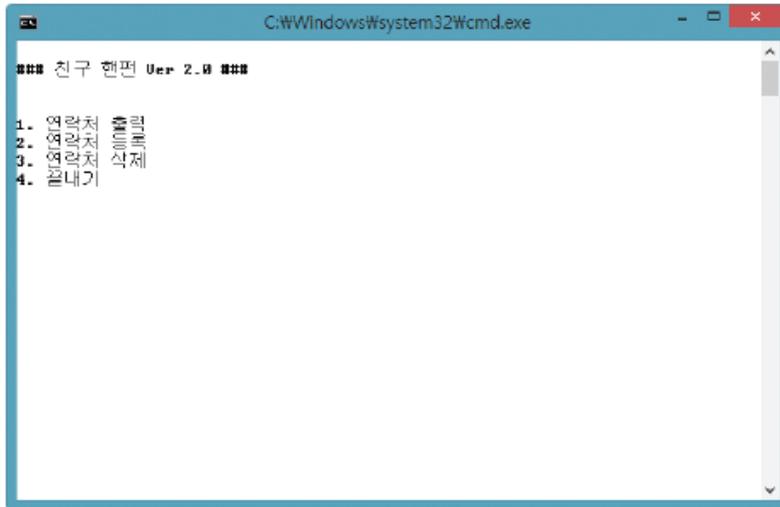
Email: jhpark1@seoultech.ac.kr

핸드폰 친구 연락처 관리 프로그램

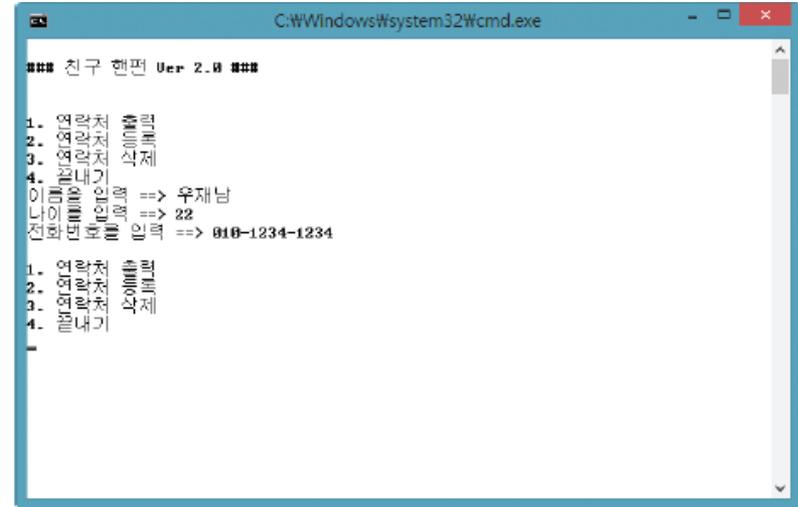
• 개요

- 파일에 연락처를 저장, 필요시 꺼내보면서 삭제 및 관리하는 프로그램
- 프로그램을 실행 → 1~4번 중 하나 입력 → 실행
 - 처음 입력된 연락처가 없으므로 2번 눌러 필요연락처 입력 (필요한 만큼 반목)
 - 입력내용 c:\temp\juso.txt 에 저장됨
 - 현재 연락처 확인해보거나 더 이상 필요없어진 연락처 삭제도 가능
 - 프로그램 마치기 4 (끝내기)

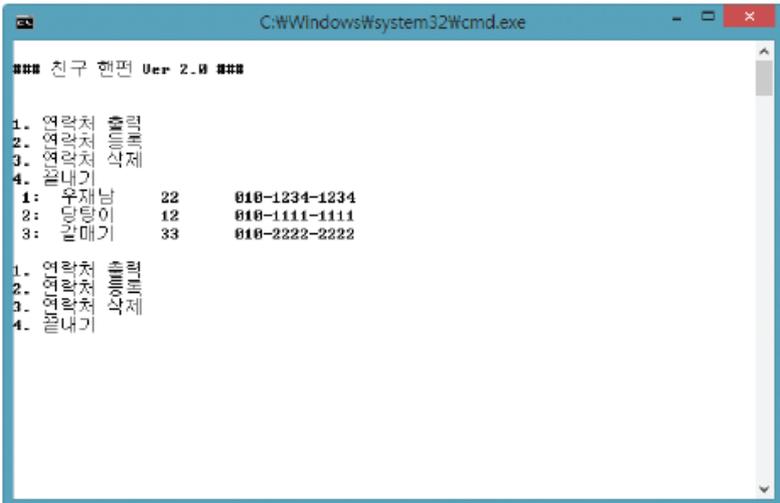
UI - 실행화면



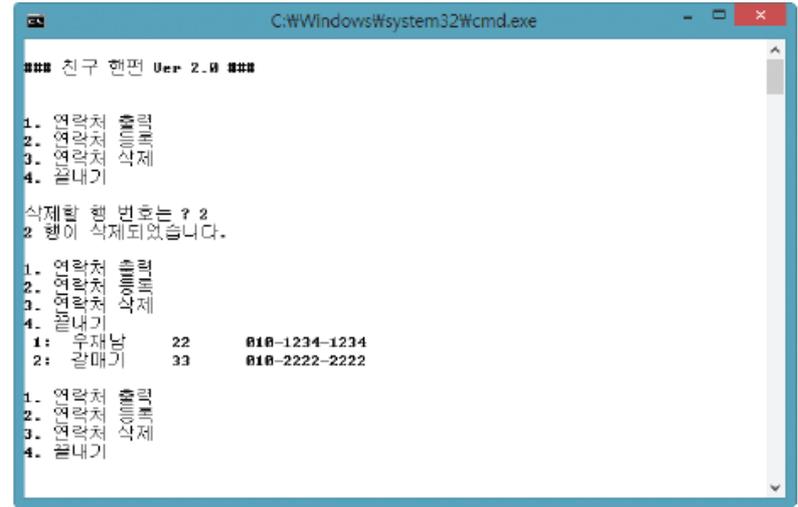
프로그램 초기 화면



연락처 등록 화면



저장된 연락처 보기

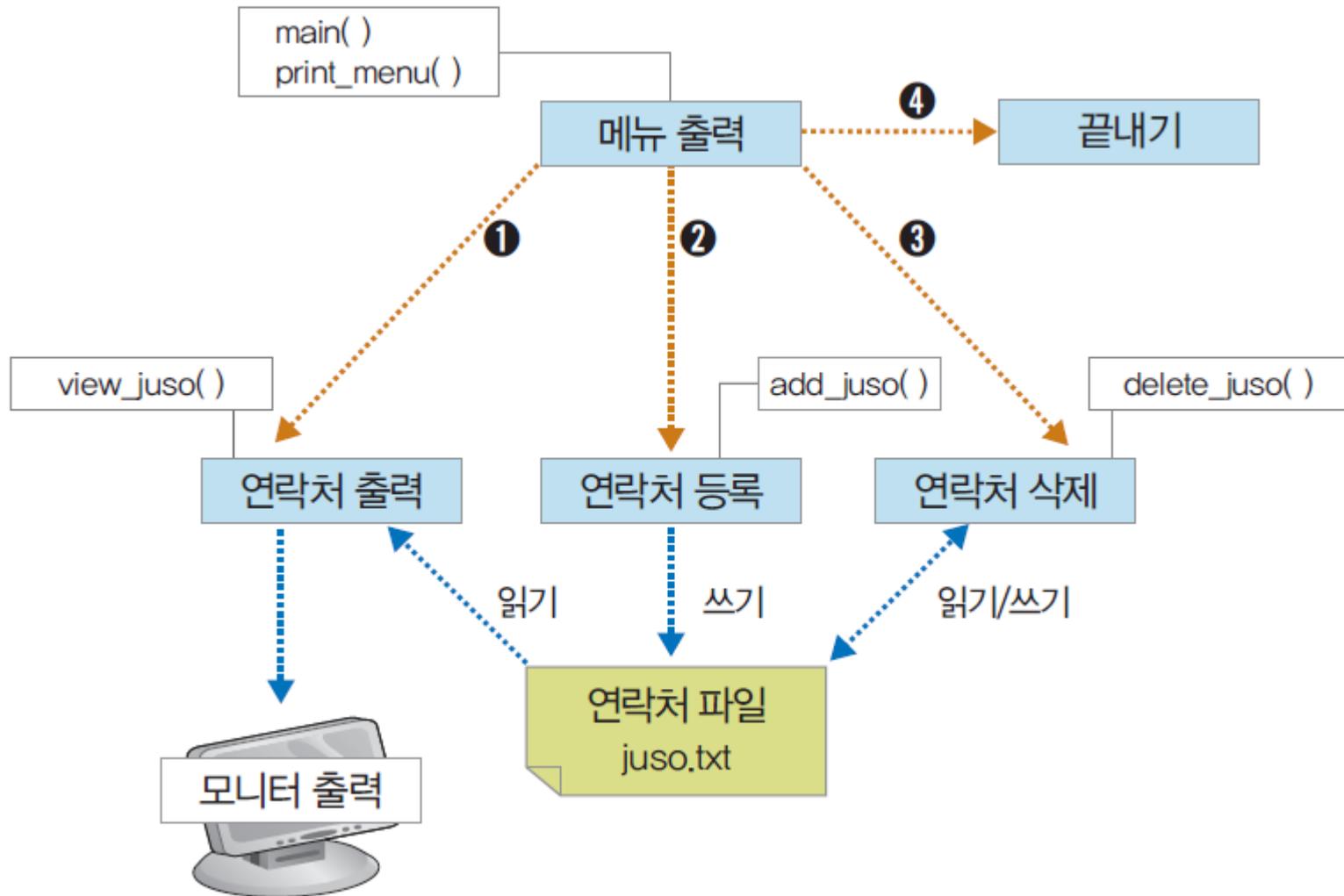


연락처 삭제 후 현재 연락처 출력

• 프로그램 구현

- ❶ 친구의 연락처는 파일에 저장해서, 프로그램을 종료해도 기록이 남게 한다.
→ 파일 처리 함수인 `fopen_s()`, `fclose()`, `fgets()`, `fputs()` 함수를 사용
- ❷ 연락처를 출력, 입력, 삭제하는 기능들은 별도의 함수로 작성
- 4가지 중 하나를 선택할 수 있는 메뉴 화면도 별도의 함수로 작성
- ❸ `main()` 함수에서는 `while`문을 써서 사용자가 종료를 선택할 때까지 반복
- 사용자의 입력에 따라 별도의 함수를 각각 호출
→ `while`문 및 `switch~case`문을 사용

프로그램 전체 흐름도



부분 소스 코드

-
- // 연락처 구조체 정의
- struct address {
- • };
- void print_menu();
- void view_juso();
- void add_juso();
- void delete_juso();
- • // 연락처 파일명을 전역변수로 고정
- const char *fname = "c:\\temp\\juso.txt";
- • int main ()
- {
- while(select != 4)
- {
- print_menu();
- select = _getch();
- • switch(select)
- {
- case '1' : view_juso();
- break;
-
- default : printf("\n 잘못 입력했어요. 다시 선택하세요.\n");
- }
• }
• return 0;
- }

부분 소스 코드

- // 연락처 구조체 정의
- struct address {
- • };
- void print_menu();
- void view_juso();
- void add_juso();
- void delete_juso();
- // 연락처 파일명을 전역변수로 고정
- const char *fname = "c:\\temp\\juso.txt";
- int main ()
- {
- • while(select != 4)
- {
- print_menu();
- select = _getch();
- • switch(select)
- {
- case '1' : view_juso();
- break;
-
- default : printf("\n 잘못 입력했어요. 다시 선택하세요.\n");
- }
- }
- • return 0;
- }

부분 소스 코드

```

• // 처음에 사용자의 선택을 위한 메뉴 출력
• void print_menu()
• {
•     printf("\n");
•     printf("1. 연락처 출력 \n");
•     .....
•     printf("4. 끝내기 \n");
• }

• // 연락처 파일에서 기존 입력된 내용을 읽어서 출력
• void view_juso()
• {
•     char str[200] = "";
•     FILE *rfp, *wfp;
•     int i;

•     fopen_s(&rfp, fname, "r");

•     // 만약, 연락처 파일이 없다면 빈파일을 만들
•     if (rfp == NULL)
•     {
•         .....
•     }

•     // 기존의 연락처를 모두 읽어서 출력
•     for(i = 1; ; i++) // i는 계속 1씩 증가하는 무한루프
•     {
•         fgets(str, 200, rfp);
•         if (feof(rfp))
•         .....
•     }

•     // i가 1이면 실제 파일에는 내용이 없음
•     if (i == 1)
•         puts("\n ** 연락처 파일에 전화번호가 하나도 없어요. **\n");
•     fclose(rfp);
• }

```

부분 소스 코드

- // 연락처를 추가
- void add_juso()
- {
- struct address adr = {"", "", ""};
- char wstr[200] = "";
- FILE *wfp;
- fopen_s(&wfp, fname, "a"); // 파일 추가 모드로 열기
- printf("이름을 입력 ==> ");
- gets_s(adr.name, 30);
-
- // 입력된 3개의 값을 하나의 문자열로 만듦
- strcat_s(wstr, 200, adr.name);
- strcat_s(wstr, 200, "\t");
-
- fputs(wstr, wfp); // 파일에 문자열 쓰기
- fclose(wfp);
- }

부분 소스 코드

```

• // 연락처 파일에서 선택한 연락처를 제거
• void delete_juso()
• {
• // 연락처 파일의 내용 전체를 저장하기 위한 2차원 배열
• char read_str[50][200] = { "", }; // 최대 연락처 개수를 50개로 가정
• char str[200] = "";
• FILE *rfp, *wfp;
• int del_line, i, count=0;
• fopen_s(&rfp, fname, "r");

• // 연락처 파일이 없으면 돌아간다.
• .....
• }

• printf("\n삭제할 행 번호는 ? ");
• scanf_s("%d", &del_line);

• for(i = 0; i < 50; i++) // 파일에 있는 동안에 수행, 단 최대 50개
• {
• strcpy_s(str, 200, "");
• fgets(str, 200, rfp);

• if(i+1 != del_line) // 삭제하는 행이 아니라면, 추가
• .....
• else
• ....
• if (feof(rfp))
• break;
• }

• fclose(rfp);

• // 파일을 쓰기 모드로 열고, 새로운 내용을 쓴다.
• .....
• fclose(wfp);
• }

```

참고문헌

- 우재남, C언어 for Beginner, 한빛아카데미, 2018