

プログラム解説 (電話帳)

山本昌志*

2005 年 1 月 14 日

1 はじめに

前回の授業で、3 題の練習問題を課した。そのうち、2 題は教科書に解答があるので、分からない人はそれを見て、理解してほしい。教科書に解答が示されていない 3 題目の問題について、ヒントを与える。前回のプリントに示したように、それは 3 つのステップで考えるの良いであろう。ここでは、この最初の 2 ステップについて、解説する。

2 STEP 1(データ作成)

最初のステップは電話帳のデータベースを作成するプログラムである。

2.1 動作内容

電話帳のデータベース作成のプログラムは、次のような動作が必要である。

1. データを書き込むためのファイルを開く。
2. 人数分のデータをキーボードから読み込み、ファイルに書き出す。
 - 名字の読み込み
 - 名前の読み込み
 - 電話番号の読み込み
 - 1 行を”名字 名前”のフォーマットでファイルに書き込む
 - 次の 1 行に電話番号を書き込む
3. ファイルを閉じる。

* 国立秋田工業高等専門学校 電気情報工学科

2.2 プログラム例

```
#include <stdio.h>

int main(void)
{
    FILE *write_list;
    char myoji[32], namae[32];
    char tel[16];
    int i;

    write_list = fopen("address.txt","w");

    for(i=0;i<=9;i++){
        printf("%d 番目の人の名前を入力してください(ローマ字)\n",i+1);
        printf("\t 名字:");
        scanf("%s",myoji);
        printf("\t 名前:");
        scanf("%s",namae);
        printf("\t 電話番号:");
        scanf("%s",tel);

        fprintf(write_list,"%s %s\n",myoji, namae);
        fprintf(write_list,"%s\n",tel);
    }

    fclose(write_list);

    return 0;
}
```

3 STEP 2(検索)

つぎのステップは電話帳のデータベースから、目的のデータを検索するプログラムである。

3.1 動作内容

氏名を検索する場合、“名字”と“名字 名前”の場合がある。その両方に対応するためには、教科書に書いてあるとおり、

- 名字の比較には、“名字+空白”を比較
- 氏名の比較には、“名字+空白”を比較

のようなことが必要である。すなわち、空白を上手にを使って、氏名や名字の区切りに使うのである。詳細については、教科書の p.333 ~ 334 を見よ。

また、氏名をキーボードから 1 度に読み込み場合、空白付きで読み込む必要がある。そのためには、`gets` あるいは `fgets` を使うことになる。`scanf` を使うことも考えられるが、その場合は空白が読み込めないののでその処理が必要である。それぞれは、

`gets` 1 行空白付きで読み込める。また、キーボードからデータを入力したときの最後の改行 (`\n`) は付かない。ただし、配列より多くのデータがキーボードから送られるとそれを越えてメモリーに格納しようとするので問題が発生することがある。

`fgets` 1 行を空白付きで読み込める。しかし、データを入力したときの最後の改行も配列の中に残る。この処理が必要である。

`scanf` 名字と名前の間の空白が読み込めない。この処理が必要である。

のような得失があり、どれを使うか考えなくてはならない。ここでは、`fgets` を使ったので、以降はそれに仮定して説明する。ただし、諸君はどれを使っても良いが、それなりの処理をする必要がある。

この空白を考慮すると、電話帳のデータベースを検索するプログラムには、次のような動作が必要である。

1. 検索する名前の読み込みと空白の追加

- キーボードから 1 行を読み込む
- 改行を空白に変更

2. 比較する文字数 (空白付き) のカウント

3. データベースファイルのオープン

4. 人数分、データベースから読み込み、検索する文字列と比較する。そして、一致したら、名前と電話番号を表示する。

- キーボードから 1 行を読み込む
- 格納されたデータの最後に空白を追加
 - ”`\0`” を空白に変更
 - ”`\0`” を追加
- 電話番号の読み込み
- データベースから読み込んだ氏名と検索する名前の比較。もし、一致すれば、氏名と電話番号を表示する。

5. ファイルのクローズ

3.2 プログラム例

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    FILE *read_list;
    char r_shimei[16][64];
    char r_tel[16][16];
    char search_name[32];
    int compare_char;
    int i;

    /*----- 検索する名前の読み込みと改行処理 ----- */

    printf("\n\t 検索したい人の名前を入力してください(ローマ字):",i+1);
    fgets(search_name,64,stdin);          /* 検索する名前の読み込み */
    printf("\n");
    search_name[strlen(search_name)-1]=' '; /* 改行を空白に変換 */

    compare_char = strlen(search_name);   /* 比較する文字数(空白を含む) */

    /*----- ファイルのオープン ----- */

    if((read_list = fopen("address.txt","r"))==NULL){
        printf("ファイルが開けません!!! \n");
        return 1;
    }

    /*----- データの読み込みと比較、表示 --- */

    for(i=0;i<=9;i++){

        fgets(r_shimei[i],64,read_list); /* 氏名の読み込み(改行付) */
        r_shimei[i][strlen(r_shimei[i])-1]=' '; /* 改行を空白に変換 */

        fgets(r_tel[i],16,read_list); /* 電話番号の読み込み(改行付) */
        r_tel[i][strlen(r_tel[i])-1]=' '; /* 改行を空白に変換 */

        if(strncmp(r_shimei[i],search_name,compare_char)==0){
            printf("\t\t 氏名:%s\t 電話番号:%s\n",r_shimei[i], r_tel[i]);
        }
    }

    printf("\n");
    fclose(read_list); /* ファイルのクローズ */
    return 0;
}
```

3.3 プログラム解説

`fgets(search_name,64,stdin)`; キーボード (`stdin`) から 1 行の文字列を読み込んで、それを配列 (`search_name`) に格納している。最大読み込みバイト数は 64 バイトである。読み込まれた最後の文字は、改行 (`\n`) となる

`search_name[strlen(search_name)-1]=' '`; 改行を含んだ文字列の文字数は、関数 `strlen` でカウントしている。この文字数から 1 を引いた値が、改行がある添え字を示すことになるので、その場所を空白に置き換えている。

`compare_char = strlen(search_name)`; 比較すべき空白を含んだ文字数をカウントして、`compare_char` に格納している。

`strncmp(r_shimei[i],search_name,compare_char)` 文字列を比較する関数で、等しければ 0 を返す。比較する文字列は、配列 `r_shimei[i]` と `search_name` の先頭の `compare_char` で示された数である。