

情報研究会CACTUS

第10回 講義資料

前回の解答 練習問題1

```
#include <stdio.h>

int main() {
    char word[128]; // 文字列の宣言
    char *p; // ポインタの宣言
    p = &word[0]; // ポインタに文字列の先頭アドレスを代入

    printf("文字列を入力\n");
    scanf("%s", word);

    printf("anser1 文字列の二文字目は%c\n", word[1]); // 文字列をそのまま使う場合
    printf("anser2 文字列の二文字目は%c\n", *(p+1)); // ポインタを用いた場合
}
```

前回の解答 練習問題2

```
#include <stdio.h>

int countlen(char*); //プロトタイプ宣言

int main() {
    char word[128]; //文字列の宣言

    printf("文字列を入力\n");
    scanf("%s", word);

    printf("文字列の長さは%d\n", countlen(word));
}

//文字列の長さを数える関数
int countlen(char *a) {
    int len = 0;
    while (*a != NULL) { //文字列の中身がNULL文字ではなかったらwhile文の中に入る
        len++;
        a++;
    }
    return len;
}
```

前回の解答 発展問題

```
#include <stdio.h>

int main() {
    char word1[128] = {0}; //入力した文字列を入れておく文字列
    char word2[128] = {0}; //数字を消去した文字列を入れる文字列
    int i = 0, j = 0;

    printf("文字列を入力\n");
    scanf("%s", word1);

    while (word1[i]) { //入力された文字列の中身の一つずつ見ていく
        if (word1[i] <= '0' || word1[i] >= '9') { //文字列の中身が数字じゃなかったら
            word2[j] = word1[i]; //文字を代入
            j++;
        }
        i++;
    }
    printf("数字を削除すると%s\n", word2);
}
```

今週の内容

- ファイル入出力

(練習問題などは今までの復習になっているよ)

ファイル入出力

今まではキーボードから入力したりしていたが、C言語ではファイルからデータを読み取ったり、ファイルにデータを書き込むことが出来る。

ファイルを用いることでプログラムの実行結果をファイルに保存しておくことが出来る。

莫大な量のデータなどはファイルを使うことで非常に扱いやすくなる。

(キーボード入力だといちいち入力しないといけない...(0 _ _)0パタッ...)

サンプルプログラム1

```
1 #include <stdio.h>
2
3 int main(){
4     FILE *fp;           //ファイルポインタの宣言
5     fp=fopen("abc.txt","r"); //ファイルを開く
6
7     if(fp==NULL){       //ファイルが存在しない場合
8         printf("ファイルが見つかりません\n");
9         return 1;
10    }
11    //ファイルが存在する場合
12    printf("ファイルを開きました\n");
13    fclose(fp);          //ファイルを閉じる
14 }
15 [EOF]
```

実行結果

ファイルが見つかりません

次にファイル内に"abc.txt"を追加して実行してみると...

ファイルの作る場所、作成手順

PC > ドキュメント > Visual Studio 2017 > Projects > kosyusiryo > kosyusiryo

名前	更新日時	種類	サイズ
Debug	2017/07/08 22:39	ファイル フォルダー	
kosyusiryo.vcxproj	2017/07/08 22:39	VC++ Project	7 KB
kosyusiryo.vcxproj.filters	2017/07/08 22:39	VC++ Project Filters F...	1 KB
rensyu1.c	2017/07/08 23:02	C Source	1 KB



Debug	2017/07/08 22:39	ファイル フォルダー	
kosyusiryo.vcxproj	2017/07/08 22:39	VC++ Project	7 KB
kosyusiryo.vcxproj.filters	2017/07/08 22:39	VC++ Project Filters F...	1 KB
rensyu1.c	2017/07/08 23:02	C Source	1 KB
新しいテキスト ドキュメント.txt	2017/07/08 23:25	テキスト ドキュメント	0 KB

1. __.cファイルのあるところで右クリック
- 2.新規作成の中にあるテキストドキュメントを選ぶ
- 3.名前を変更する

ファイルポインタ

ファイルを扱うときはFILE型のポインタの変数が必要となる。
FILE型は<stdio.h>内で定義されていて、ファイル入出力を行う際に必要な情報を管理している。

宣言の仕方

FILE *変数名;

例、

```
FILE *fp;
```

ファイル入出力の基本

ファイルを利用するときは、fopen関数であらかじめ開き、処理が終わったらfclose関数でファイルを閉じる必要がある。

おおまかなイメージとしては、ノートの中を見たり、中に書き込むのと同じだよ
(ノートを開く→中を見たり、書き込んだりする→ノートを閉じる)



fopen関数(1)

```
fp=fopen("abc.txt","r");
```

ファイルポインタ

ファイル名

モード

fopen関数によって、ファイルポインタに開きたいファイル名と、モードの種類が渡される。

fopen関数(2)

よく使うモードの種類

モード	種類	ファイル操作
“r”	読み取り	なし
“w”	書き込み	上書き
“a”	追加書き込み	ファイルの終わりの位置から書き込む

書き込みまたは追加書き込みのモードでは、ファイルが存在しない場合、ファイルが新規作成されるよ。

エラー表示

fopen関数ではファイルを開くことが出来なかったときNULLを返す。
これを利用して、ファイルが開けなかった場合のエラー処理を行うことが出来る。

```
if(fp==NULL){ //ファイルが存在しない場合  
    printf("ファイルが見つかりません\n");  
    return 1;  
}
```

エラー処理を行うことで、エラーの原因がわかりやすくなるので、書いておくとよいよ。

fclose関数

ファイルを閉じるときはfclose関数を用いる。

```
fclose(fp); // ファイルを閉じる
```

ファイルポインタ

サンプルプログラム2

```
1 #include <stdio.h>
2
3 int main(){
4     int a=100; //int型の宣言
5     double b=0.88; //double型の宣言
6     char c[128]="CACTUS"; //char型の宣言
7
8     FILE *fp; //ファイルポインタの宣言
9     fp=fopen("abc.txt", "w"); //ファイルを開く (書き込みモード)
10
11     fprintf(fp, "%d %.2f %s", a, b, c); //ファイルの中に書き込む
12     fclose(fp); //ファイルを閉じる
13 }[EOF]
```

実行したら

“abc.txt”の中を見てみよう

また、“w”を“a”に変えて実行してみよう

abc.txt - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

100 0.88 CACTUS

fprintf関数

fprintf関数ではファイルにデータを書き込めるよ。

ファイル
ポインタ

書式指定
文字列

変数

```
fprintf(fp, "%d %.2f %s", a, b, c);
```

ファイルポインタの引数が増えただけであとは、printf関数と一緒に！！

ファイルポインタの中身がないとエラーとなるので注意

サンプルプログラム3

```
1 #include <stdio.h>
2
3 int main(){
4     int i=0;
5     char s[4][128]; //二次元文字列
6     FILE* fp; //ファイルポインタの宣言
7     fp=fopen("fruit.txt", "r"); //ファイルを開く
8
9     if(fp==NULL){ //ファイルが存在しない場合
10         printf("ファイルが見つかりません\n");
11         return 1;
12     }
13     //ファイルが存在した場合
14     printf("ファイルを開きました\n");
15     while(fscanf(fp, "%s", s[i]) != EOF){ //ファイルから文字列を読み込む
16         printf("%s\n", s[i]); //読み込んだ文字列を表示
17         i++;
18     }
19     fclose(fp); //ファイルを閉じる
20 }
```

ファイル内

fruit.txt - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

apple
banana
orange
pineapple

実行結果

ファイルを開きました
apple
banana
orange
pineapple

fscanf関数(1)

ファイルからデータを読み込む際はfscanf関数を用いるよ。
fprintf同様、ファイルポインタの引数が増えただけであとscanf関数と一緒にだよ。
ただし、読み取りモード(“r”)で開かれたファイルでないと読み込めないので注意！

```
fscanf(fp, "%s", s[i])
```

ファイルポインタ

指定文字列

変数のアドレス

fscanf関数(2)

fscanf関数はファイルのデータを1行ずつ読みこみ、ファイルポインタを次の行の先頭アドレスに移動する。

また、fscanf関数は入力項目の個数を返すので、以下のような構文をかくことで、ファイルのデータをすべて読み込むことが出来る。

(fscanf(中略) > 0) と書いてもOK !

```
while(fscanf(fp, "%s", s[i]) != EOF) { // ファイルから文字列を読み込む  
    printf("%s¥n", s[i]); // 読み込んだ文字列を表示  
    i++;  
}
```

処理

EOFは(End Of File)の略で、ファイルの終端を指しているよ。



練習問題1

キーボードから文字列5つを入力して、ファイルに書き込んでみよう。

練習問題2

ファイルから点数と名前を読み取ってから、平均点を求め平均点以上の人の名前と点数を別のファイルに書き込もう。

ヒント;
構造体を使うと楽かな

ファイルの例



score.txt - メモ帳

ファイル(F) 編集(E) 書式(O)

```
danmitu 60  
e-chan 90  
syatyo 78  
tentyo 99  
sontyo 55
```

発展問題(おまけ)暇な人はどうぞ

ファイル内に文字のアスキーコードを暗号化した数字が書き込まれているよ。("angobun.txt"とする)暗号化された数字を復号化して、文字を"fukugoka.txt"に書き込んでみよう。
暗号化、複合化のアルゴリズムは次の通りだよ。

元のコード m (10進数)

公開鍵: $N=25199$ 、 $e=503$ 秘密鍵: $d=22343$

暗号化の式: $c = m^e \bmod N$

復号化の式: $M = c^d \bmod N$

ヒント:

m^e は m の e 乗意味していて、 $\bmod N$ は N による剰余算(余り)を意味しているよ。
しかし m^e は桁数が大きくて通常の計算が出来ないので以下のようにするといいいよ。

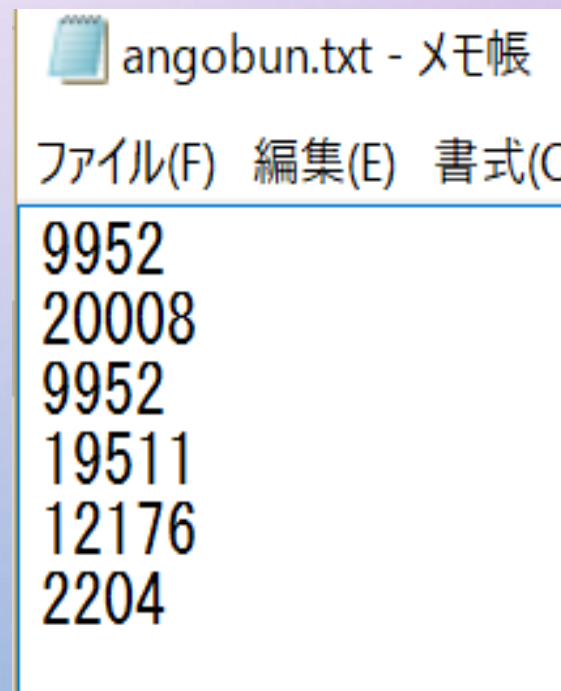
例、 $112^n \bmod 33$ の場合

$$112 * 112 \% 33 = 4$$

$$112^3 \% 33 = 112 * 4 \% 33 = 19$$

$$112^4 \% 33 = 112 * 19 \% 33 = 16$$

...



電電のプログラミングの授業の問題(改訂版)だよ