

情報研究会 CACTUS

第十回 C言語講習

今回の内容

- ・ ファイル入出力

ファイル入出力

C言語ではプログラム中からデータをファイルに書き込んだり、ファイルからデータを読み込みこんだりすることができる

ファイルをうまく使うことで今まで面倒だった手入力を省略できたり、実行結果を保存しておくことができるようになる

サンプルプログラム 1

実行後Debugフォルダの中にある
hoge.txtを確認してみよう

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int a = 1;
6     double b = 2.34;
7     char c = 'a';
8     FILE *fp; //FILE型ポインタの宣言
9
10    //ファイルを開く
11    fp = fopen("hoge.txt", "w");
12
13    //ファイルにデータを書き込む
14    fprintf(fp, "%d, %lf, %c\n", a, b, c);
15
16    //ファイルを閉じる
17    fclose(fp);
18
19    return 0;
20 }
```

ファイルポインタ

ファイルを開くためには、FILE型のポインタ（ファイルポインタ）が必要である

開かれたファイルはファイルポインタを介して参照される

ファイルポインタの宣言

FILE * 変数名

fopen関数

fopen関数を使用することでファイルを開くことができる

```
fp = fopen( "hoge.txt" , " w" )
```

ファイルポインタ

ファイル名

モード

引数として開きたいファイル名と、開くファイルの種類を指定する
「モード」が渡される

fopen関数

引数として渡したファイル名のファイルが存在しない場合は新規作成される

よく使用されるモードの種類

モード	種類	ファイル操作
"r"	読み込み専用	なし
"w"	書き込み専用	上書き
"a"	追加書き込み専用	最後の行に追加で書き込み

- ・ サンプルプログラムのfopenの引数に"r"、"a"を渡して実行してみよう

fprintf関数

fprintf関数を使用することでファイルにデータを書き込むことができる

```
fprintf(fp, "%d, %.2f, %c¥n", a, b, c)
```

ファイルポインタ

書式指定文字列

変数

printf関数にファイルポインタの引数が増えたようなもの

ファイルポインタの中身がないとエラーを起こすので注意

fclose関数

fclose関数を使用することでファイルを閉じることができる

fclose(fp)

ファイルポインタ

fclose関数を使用しなくてもプログラム終了時にファイルは自動的に閉じられるが、 fclose関数を使用して明示的にファイルを閉じたほうがよい

サンプルプログラム 2

実行結果

```
ファイルの名前の入力してください  
hoge.txt  
hoge.txtは存在しません
```

サンプルプログラム 1 の hoge.txt をサンプルプログラム 2 の Debug フォルダの中にコピーしてもう一度実行してみよう

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int a;
6     double b;
7     char c;
8     char fname[256];
9     FILE *fp; //FILE型ポインタの宣言
10
11     printf("ファイルの名前の入力してください\n");
12     scanf("%s", fname);
13
14     //エラー処理
15     if((fp = fopen(fname, "r")) == NULL){
16         printf("%sは存在しません\n", fname);
17         return 0;
18     }
19
20     //ファイルのデータを読み込む
21     fscanf(fp, "%d, %lf, %c\n", &a, &b, &c);
22
23     printf("%d, %lf, %c\n", a, b, c);
24
25     //ファイルを閉じる
26     fclose(fp);
27
28     return 0;
29 }
```

エラー処理

fopen関数を使用してファイルを開くことができない場合に、そのまま処理を続けたらエラーが起こる可能性は高い

fopen関数はファイルを開くことができないときは**NULL**を返すので、これを利用することでエラー処理を行うことができる

例)

```
if ( (fopen( "hoge.txt", "r" )) == NULL )  
{  
    エラー処理  
}
```

fscanf関数

fscanf関数を使用することでファイルからデータを読み込むことができる

```
fscanf(fp, "%d, %lf, %c\n", &a, &b, &c)
```

ファイルポインタ

指定文字列

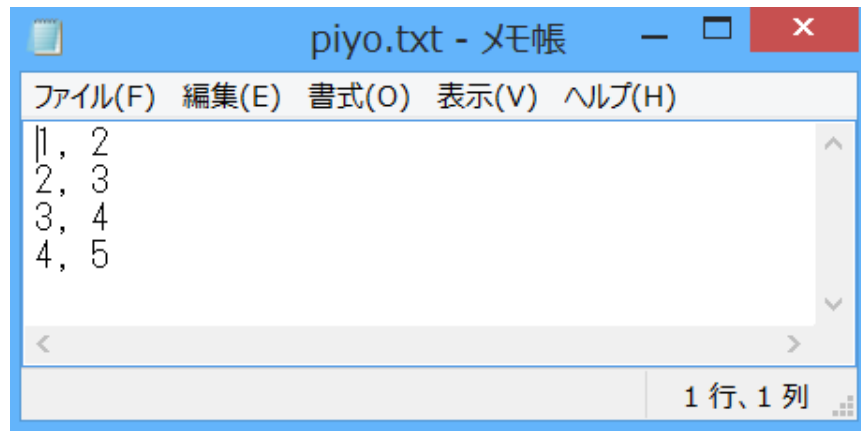
変数のアドレス

scanf関数にファイルポインタの引数が増えたようなもの

引数のファイルポインタがモード“**r**”で開かれたファイルでないとうまくデータを読み込めない

サンプルプログラム 3

サンプルプログラム 3 のフォルダに
“piyo.txt” を作成し、2つの値をカンマ
空白区切りで適当な個数書いてください



実行後Debugフォルダの中にあるfuga.txt
を確認してみよう

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     double a, b, sum;
6     FILE *ifp, *ofp; //FILE型ポインタの宣言
7
8     ifp = fopen("piyo.txt", "r");
9     ofp = fopen("fuga.txt", "w");
10
11     while(fscanf(ifp, "%lf, %lf", &a, &b) != EOF){
12         sum = a + b;
13         fprintf(ofp, "%f, %f, %f\n", a, b, sum);
14     }
15
16     fclose(ifp);
17     fclose(ofp);
18
19     return 0;
20 }
```

データの読み込み

fscanf関数はファイルのデータを一行ずつ読み込み、ファイルポインタを次の行の先頭アドレスに移動する

またすべての行のデータを読み込み終えたとき、EOF (-1) を返すので、以下のような構文を書くことでファイルのデータをすべて読み込むことができる

例)

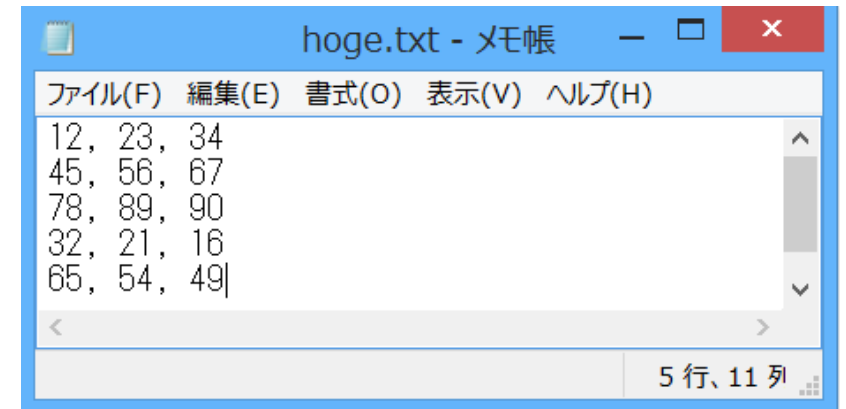
```
while (fscanf (fp, "%lf, %lf", &a, &b) != EOF)
{
    処理
}
```

練習問題 1

5回0～100までの整数を3つ入力し、テキストエディタに保存するプログラムを作成してください

実行例

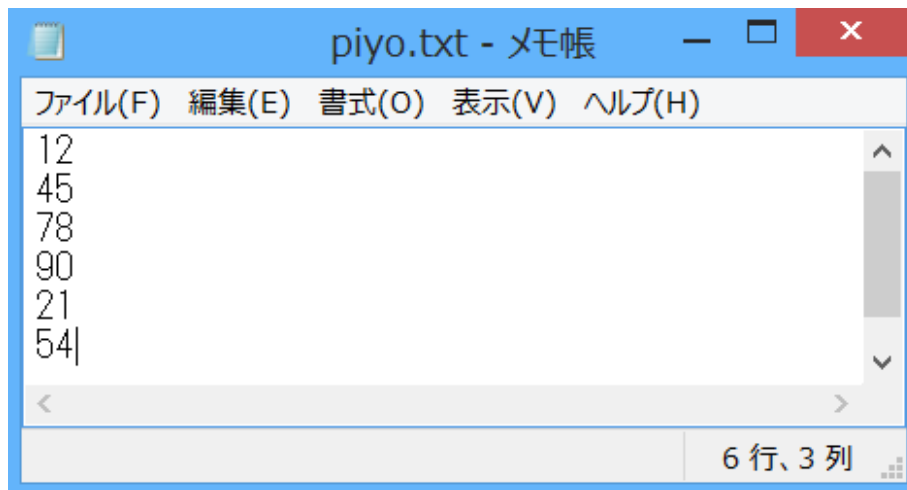
```
12 23 34
45 56 67
78 89 90
32 21 16
65 54 49
```



練習問題 2

練習問題 1 で保存したデータを読み込み、3 の倍数を他のテキストエディタに保存するプログラムを作成してください

実行例



練習問題 3

練習問題 2 で保存したデータを読み込み、平均値を出力するプログラムを作成してください

実行例

```
平均値:50.000000
```