

プログラミング基礎

<http://bit.ly/prog2d>

ファイル入出力 (2)

ファイル入出力を使った繰り返し処理

後期 第9週

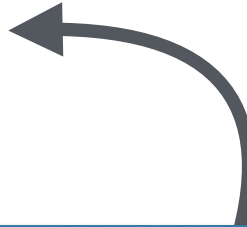
2017/11/27

fgets()の補足

最近のCコンパイラでは、fgets()の2番目の引数は、「**終端文字も含めた文字数**」を指定することになっています。そのため、サンプルプログラムで示した以下の場所は、「NUM-1」でなく「**NUM**」を指定します。

[変更後の練習8-2のサンプルプログラム]

```
17:      fgets(str1, NUM, fp);  
18:      fgets(str2, NUM, fp);
```



str1とstr2は、20個（つまりNUM個）の要素をもったchar型の配列なので、**終端文字を含めて最大20文字**読み込める


今回は

「ファイルをコピー（複製）する」プログラム
を通して、ファイル入出力を使った
繰り返し処理について説明します

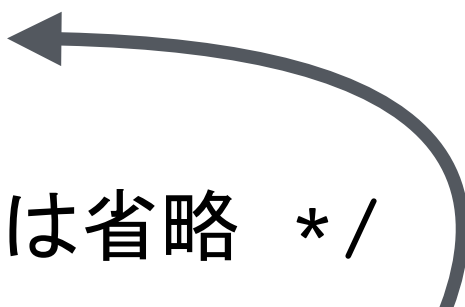
ファイル入出力の繰り返し処理 (1文字ずつ)

```
1: #include <stdio.h>
2:
3: int main(void)
4: {
5:     FILE *in, *out;
6:     char ch;
7:     in = fopen("testIn.txt", "r");
8:     out = fopen("testOut.txt", "w");
9:
10:    /* ファイルポインタがNULLの場合の処理は省略 */
11:
```

コピー元のファイルを
読み込みモードで開く



コピー先のファイルを
書き込みモードで開く



ファイル入出力の繰り返し処理（1文字ずつ）

「1文字ずつのコピー」であることが分かるようにコピー先に書き込む

```
12:      fprintf(out, "(character copy)\n");
13:      while( (ch=fgetc(in)) != EOF) {
14:          printf("ch: %c \n", ch);
15:          fputc(ch, out);
16:      }
17:      fclose(in);
18:      fclose(out);
19:      return 0;
20: }
```

whileの様子が画面でも確認できるように出力する

コピー先に一文字書き込む

2つのファイルを閉じる

1文字読み込む関数fgetc()は、ファイルの最後に到達すると、EOF(end-of-file)を返す（whileの条件は「読み込んだ文字をchに代入し、それがEOFではない間繰り返す」ことを表している）


ファイル入出力の繰り返し処理（1行ずつ）

```
1: #include <stdio.h>
2: #define NUM 100
3:
4: int main(void)
5: {
6:     FILE *in, *out;
7:     char str[NUM];
8:
9:     in = fopen("testIn.txt", "r");
10:    out = fopen("testOut.txt", "w");
11:
12:    /* ファイルポインタがNULLの場合の処理は省略 */
13:
```

読み込む行を格納する文字列



コピー元のファイルを
読み込みモードで開く



コピー先のファイルを
書き込みモードで開く



ファイル入出力の繰り返し処理（1行ずつ）

「1行ずつのコピー」であることが分かるようにコピー先に書き込む

```
13:
14:     fprintf(out, "(string copy)\n");
15:
16:     while((fgets(str, NUM, in)) != NULL) {
17:         printf("str: %s", str);
18:         fputs(str, out);
19:     }
20:     fclose(in);
21:     fclose(out);
22:     return 0;
23: }
```

whileの様子が画面でも確認できるように出力する

コピー先に一行書き込む

2つのファイルを閉じる

1行読み込む関数fgets()は、ファイルの最後に到達するとNULLを返す（whileの条件は「読み込んだ行をstrに代入し、戻り値が**NULL**ではない間繰り返す」ことを表している）

ファイル入出力の繰り返し処理（書式付き）

fscanf()の処理の例は、p.411 Sample12.cのプログラムを見て下さい。

fscanf()を使うと、指定した書式で読み込むことができ、ファイルに保存されているデータを読み込み時に、**文字（または文字列）以外の型に格納することができます。**

ファイル入出力の補足

- ▶ `fgets()`で読み込んだデータは、必ずしも`fputs()`で書き込む必要はありません。必要に応じて、`fgetc()`, `fputc()`, `fgets()`, `fputs()`, `fscanf()`, `fprintf()`を自由に組み合わせて使えます。
- ▶ 1つのプログラムの中で、`fopen()`を複数使うことで、今回のコピーの例のように、同時にいくつものファイルを開くことができますが、開いたファイルポインタの分だけ閉じることを忘れないで下さい。
- ▶ `fopen()`でファイル名のみを指定した場合、そのファイルは実行ファイル（つまり`a.out`）と同じフォルダに存在していることを意味します。（今回の例では「`testln.txt`」と指定しているので、ファイル`testln.txt`と`a.out`は同じフォルダに存在する必要があります。）
- ▶ 違うフォルダにあるファイルを指定する場合は、「`/usr/local/common/kogai/prog2d/testln.txt`」のようにフォルダの階層のパスを指定すれば良いです。

【練習9-1】

「**1文字ずつ**コピーする」 サンプルプログラムを入力して動作を確認してみましょう。実行後にファイル「testOut.txt」を開いて、書き込まれた内容を確認してみましょう。

コピー元のファイルは、以下のようにcpコマンドを使って、自分の所にコピーして使うことができます。

```
$ cp /usr/local/common/kogai/prog2d/testIn.txt . (←ここにピリオド)
```

【練習9-2】

「1行ずつコピーする」 サンプルプログラムを入力して動作を確認してみましょう。実行後にファイル「testOut.txt」を開いて、書き込まれた内容を確認してみましょう。

【課題9-1】

練習9-1で入力したプログラムをもとに、「アルファベットを**大文字に変換して**コピーする」プログラム作成してください。

(画面の出力で変換する必要はありません。)

- ▶ アルファベットを大文字にするには、標準ライブラリ関数 `toupper()` が使えます。(p.466参照)
- ▶ `toupper()` は、アルファベットであれば大文字を返しますが、それ以外の文字は**そのまま返します**。

【課題9-2】

練習9-2で入力したプログラムをもとに、「各行の先頭に“---”（ハイフン3つ）という文字列を追加してコピーする」プログラムを作成してください。
（画面の出力に追加する必要はありません。）

▶ fputs()の処理をもう一つ追加して、行の先頭に付け足したい文字列をファイルに出力します。

[実行結果（コピー先ファイルの中身）]

```
--- In Japan, Masayoshi Son is known as the eccentric Internet  
--- billionaire who upended the country's telecom industry.  
(以下省略)
```

【課題9-3】

課題9-2で入力したプログラムをもとに、「各行の先頭に**行番号を追加して**コピーする」プログラムを作成してください。

- ▶ 行番号をカウントする変数を用意する
- ▶ 課題9-2のプログラムで、`fputs()`の処理を**`fprintf()`**に変えて、「行番号」と「読み込んだ行の文字列」を出力するようにする

[実行結果（コピー先ファイルの中身）]

```
1: In Japan, Masayoshi Son is known as the eccentric Internet  
2: billionaire who upended the country's telecom industry.  
(以下省略)
```


【練習9-3】

p.411 Sample12.cのプログラムを入力して、
fscanf()を使った読み込みの動作を確認してみま
しょう。

【課題9-4】

練習9-3のプログラムで使ったファイル「test3.txt
(p.411参照)」に、以下のようなデータを追加する。
(後ろに1列追加する)

[追加後の「test3.txt」の中身]

80,91

68,18

22,36

33,82

56,71

78,65

33,88

56,95

【課題9-4】

練習9-3で作成したプログラムをもとに、「追加した列に対しても、値を読み込み、最大値と最小値を求める」プログラムを作成してください。

- ▶ 配列testと同様に、追加した列の値を格納する配列と最大値と最小値を格納する変数を用意する。
- ▶ fscanf()の読み込む処理を変更する。
「"%d"」を「"%d,%d"」に変更すると、「カンマ (,) で区切られた2つの整数値を読み込み、それぞれを指定した変数に格納する」ようになる。
(それぞれをどの変数に格納するのかは、scanf()と同じように指定する必要がある。)
- ▶ 最大値と最小値を求める繰り返し処理に、追加した列に対する処理を追加する。
- ▶ 点数が区別できるように出力を工夫する。(実行結果を参照)

【課題9-4】

[実行結果 (標準出力)]

No.1 (80) (91)

No.2 (68) (18)

No.3 (22) (36)

No.4 (33) (82)

No.5 (56) (71)

No.6 (78) (65)

No.7 (33) (88)

No.8 (56) (95)

max: 80 95

min: 22 18

まだ余裕のある人は…

【課題9-5】

課題8-4で作成したプログラムをもとに、「標準出力へ出力した情報を全て**ファイルへ出力する**」プログラムを作ってください。

- ▶ 書き込みモードで別のファイルを開く。
(閉じるも追加する。)
- ▶ `printf()` の処理を全て `fprintf()` に変更する。

小テストについて

小テストの注意点

- 他人の力は借りずに、自分だけでプログラムを作成する。（つまり定期試験と同様）
- 小テスト中は、演習室外へのネットワークアクセスは遮断される。

小テストについて

小テスト中に参照できるもの

- 教科書, 配付資料
- 自分のホームディレクトリ（ホームフォルダ）以下に保存されているファイル
- 小テストでは紙媒体のものは参照可能
- 上記以外の情報を参照することは不正行為とする
例：USBで接続された機器に保存されているファイルの参照
ネットワークを介した情報の参照、など