

プログラミングII

<http://bit.ly/Prog3i>

コマンドライン引数

前期 第6週

2019/5/21

コマンドライン引数について

プログラム実行時（つまり「a.out」を実行する時）
に、ユーザがプログラムに与えることができる引数

これまでmain()で使っていた仮引数voidの代わりに、**仮引数argcとargvを使う** (p.421~)

```
1: #include <stdio.h>
2:
3: int main(int argc, char *argv[])
4: {
5:     int i;
6:
7:     printf("コマンドライン引数の个数: %d\n", argc);
8:     for(i=0; i<argc; i++) {
9:         printf("argv[%d]: %s\n", i, argv[i]);
10:    }
11:
12:    return 0;
13: }
```

argvが**文字列の配列**となり、argcが**配列の要素数**となって、プログラムが開始する

コマンドライン引数の例

「./a.out」の後に、文字列を空白で区切って指定する

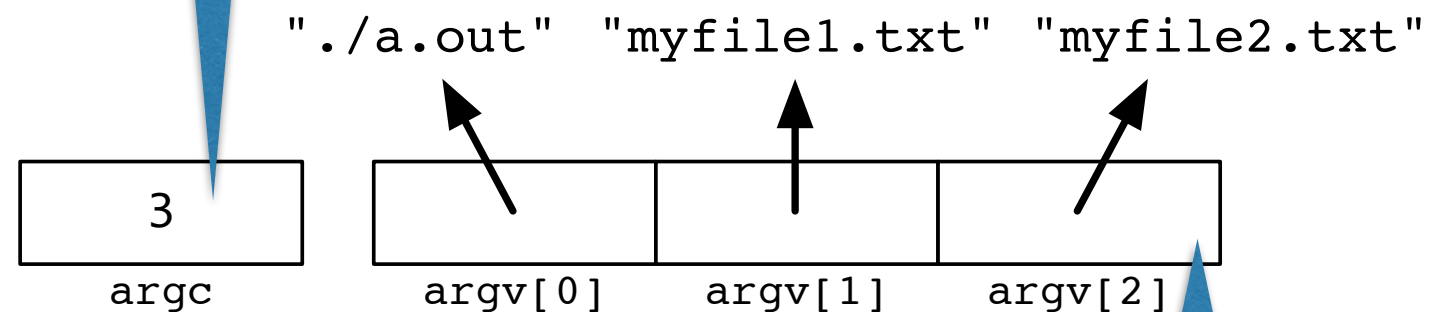
端末 (ターミナル)

```
$ ./a.out myfile1.txt myfile2.txt
```

argcは配列の要素数が入っている

実行

プログラム



「./a.out」も含めて、argvがこれらの文字列を参照している

```
int main(int argc, char *argv[])  
{  
    ...  
}
```

【課題の準備】

演習室で作業する前に、以下のコマンドを
入れるだけで準備が完了する

```
$ mygitclone 「自分のGitHubユーザ名」  
$ cd prog3i-(ユーザ名)  
$ ./myconf
```

※本体をシャットダウンするまでは、
上記「mygitclone」と「myconf」の設定は有効です

【課題の準備】

以下の流れで、課題のプログラムを作るためのフォルダを準備しましょう。

1. 端末を起動して、以下のコマンドを実行して前期第6週のフォルダを作る
\$ cd prog3i-(ユーザ名) (←既に移動しているなら不要)
\$ mkdir week106
\$ cd week106

※課題で作るファイル名は各自で決めて構いません。

【練習6-1】

コマンドライン引数のサンプルプログラムを入力して、実行結果を確認してみましょう。

実行する際は、「./a.out」の後に文字列をいくつか指定してみましょう。

【課題6-1】

練習6-1のサンプルプログラムに対して、配列argvの文字列を**逆順**に表示するプログラム（最後の要素の文字列から0番目の要素の文字列の順で表示する）ように変更したプログラムを作成して下さい。

[実行結果]

```
$ ./a.out myfile1.c myfile2.c
```

```
コマンドライン引数の個数: 3
```

```
argv[2]: myfile2.c
```

```
argv[1]: myfile1.c
```

```
argv[0]: ./a.out
```


【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題6-1提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザ名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザ名))

【課題6-2】

コマンドライン引数で指定した文字列に対して、「それぞれの文字列の長さ（つまり、文字数）とその合計を求める」プログラムを作成して下さい。

- ▶ argvの文字列に対して、標準ライブラリ関数strlen()を使って文字列の長さを求める
- ▶ 上記の長さを求める処理をargvの要素数分繰り返す処理を作る
- ▶ 出力する情報は、以下の実行結果を参照

strlenの例

文字列に関する標準ライブラリ
関数を使う時にインクルードする

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

```
int main(void)
{
    char *str = "Hello";
    int len;
    len = strlen(str);
    printf("文字列の長さ: %d\n", len);

    return 0;
}
```

関数strlen()は、引数が参照
している文字列の長さを返す

【課題6-2】

[実行結果]

```
$ ./a.out  pointer  malloc  free  stream  struct  
pointer(7)  
malloc(6)  
free(4)  
stream(6)  
struct(6)  
result: 29
```

【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題6-2提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザ名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザ名))

【課題6-3】

コマンドライン引数で指定した文字列に対して、「それぞれの文字列を整数に変換してその合計を求める」プログラムを作成して下さい。

- ▶ argvの文字列に対して、標準ライブラリ関数 `atoi()` を使って整数に変換する
- ▶ 出力する情報は、以下の実行結果を参照

atoiの例

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

int main(int argc, char *argv[])
{
    char *str = "123";
    int x;
    x = atoi(str); //文字列strを整数xに変換する
    printf("str: %s, x: %d\n", str, x);

    return 0;
}
```

【課題6-3】

[実行結果]

```
$ ./a.out 8 3
```

(2個の整数を指定した場合)

```
8 + 3 = 11
```

```
$ ./a.out 3 1 4 2 8 6
```

(6個の整数を指定した場合)

```
3 + 1 + 4 + 2 + 8 + 6 = 24
```


【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題6-3提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog3i/prog3i-\(ユーザ名\)](https://github.com/nit-ibaraki-prog3i/prog3i-(ユーザ名))

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）
- プログラムの提出はGitHubを使用する

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