
Programming for Engineers Command Line Arguments



ICEN 200 – Spring 2018
Prof. Dola Saha

Command Line Argument

- It's possible to pass arguments to main from a command line by including parameters `int argc` and `char *argv[]` in the parameter list of main.
- Parameter `argc` receives the number of command-line arguments that the user has entered.
- Parameter `argv` is an array of strings in which the actual command-line arguments are stored.

Command Line

- A typical command line for the myprog program on a Linux/UNIX system is
 - `$ myprog input1 input2 input3`

Print Arguments

```
#include <stdio.h>

int main (int argc, char *argv[])
{
    int count;

    printf ("This program was called with \"%s\".\n",argv[0]);

    if (argc > 1)
    {
        for (count = 1; count < argc; count++)
        {
            printf("argv[%d] = %s\n", count, argv[count]);
        }
    }
    else
    {
        printf("The command had no other arguments.\n");
    }

    return 0;
}
```

Command line Argument (1)

```
1 // Fig. 14.3: fig14_03.c
2 // Using command-line arguments
3 #include <stdio.h>
4
5 int main(int argc, char *argv[])
6 {
7     // check number of command-line arguments
8     if (argc != 3) {
9         puts("Usage: mycopy infile outfile");
10    }
11    else {
12        FILE *inFilePtr; // input file pointer
13
14        // try to open the input file
15        if ((inFilePtr = fopen(argv[1], "r")) != NULL) {
16            FILE *outFilePtr; // output file pointer
17
18            // try to open the output file
19            if ((outFilePtr = fopen(argv[2], "w")) != NULL) {
20                int c; // holds characters read from source file
21            }
```

Command line Argument (2)

```
22         // read and output characters
23         while ((c = fgetc(inFilePtr)) != EOF) {
24             fputc(c, outFilePtr);
25         }
26
27         fclose(outFilePtr); // close the output file
28     }
29     else { // output file could not be opened
30         printf("File \"%s\" could not be opened\n", argv[2]);
31     }
32
33     fclose(inFilePtr); // close the input file
34 }
35 else { // input file could not be opened
36     printf("File \"%s\" could not be opened\n", argv[1]);
37 }
38 }
39 }
```

Add integers from Command Line

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

int main(int argc, char *argv[])
{
    // tell user if improper arguments were passed
    if (argc < 3) {
        puts("Usage: add num1 num2 num3");
    }
    else {
        int sum=0;
        // Add all the numbers
        for (int i=1; i<argc; i++)
            sum += strtol(argv[i], NULL, 10);
        printf("SUM=%d\n", sum);
    }
}
```