



Programming with C I





Fangtian Zhong
CSCI 112

Gianforte School of Computing
Norm Asbjornson College of Engineering
E-mail: fangtian.zhong@montana.edu

Command line arguments

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

```
...
```

-  argc is argument count
-  argv is argument vector (array) of **strings**
-  To convert string x to int a:
 - int a = atoi(x);
 - int a = strtol(x, NULL, 10) (last arg is base)
-  To convert string x to double b:
 - double b = atof(x);
 - double b = strtod(x, NULL)

Passing parameters to C (1)

- Often a user wants to pass parameters into the program from the command line
- This is accomplished in C using `argc` and `argv`

– For example:

```
int main (int argc, char *argv[ ] ) {  
    /* Statements go here */  
}
```

array of strings
(char *)

Number of arguments

- Call this program from Linux

```
progname arg1 arg2 arg3
```

Passing parameters to C (2)

```
#include <stdio.h>
int main(int argc, char *argv[]) {
    int count;
    printf ("Program name: %s\n", argv [0]);
    if (argc > 1) {
        for (count=1; count<argc; count++)
            printf ("Argument %d: %s\n",count,argv[count]);
    }
    else
        printf("No command line arguments entered.\n");
    return 0;
}
```

Passing parameters to C (3)

🛡️ Suppose we compiled the previous program to the executable file `cmd`

🛡️ `cmd first "second arg" 3 4 > cmd.out`

🛡️ `myargs.out` contains the following lines:

Program name: `./cmd`

Argument 1: `first`

Argument 2: `second arg`

Argument 3: `3`

Argument 4: `4`

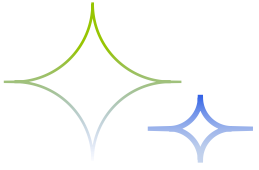
Passing arguments to C (4)

```
#include <stdio.h>

int main(int argc, char *argv[]) {
    FILE *fp;
    char k[100];
    if (argc != 2) {
        printf("Usage: %s filename\n",
            argv[0]);
        return 0;
    }
    if ((fp = fopen(argv[1], "r")) ==
        NULL) {
        printf("Cannot open file!\n");
        return 1;
    }
}
```

```
while (fgets(k, sizeof(k), fp) !=
    NULL) {
    printf("%s", k);
}
fclose(fp);
return 0;
}
```

- Generally a main function of a C program for checks whether the arguments are valid and prints simple help information.



THE END

Fangtian Zhong
CSCI 112

Gianforte School of Computing
Norm Asbjornson College of Engineering
E-mail: fangtian.zhong@montana.edu