

# Command Line Arguments

## Definition:

A command-line utility is a way of giving operating system **instructions** using lines of texts. Command-line programs operate via **command line or PowerShell**. It will interact with a command-line script.

## Advantages of using command-line utility:

Coming towards the part of why we should use the command-line utility in our program. We can call a command line program in python or any other language into a different language program quickly as each program has calling support in it for calling the command lines program. So in cases where we are writing a program in some other language, but we want to perform a task in C and call it in our program, then the command line can help us do that.

Now we are going to discuss the practical part of this tutorial. For creating a Command Line Utility In C, we should first have some knowledge about argc and argv.

## argc:

If we divide the keyword into two parts, the first will be 'agr', a short form for arguments. The second one will be 'c,' which stands for **count**. Some argc as a whole stands for argument count, which means that it stores the total number of arguments passed to the utility. The first argument count is reserved for the executable program's name, and the next ones are for the arguments passed to the program.

## argv:

Same as in argc, the arg stands for argument, but the 'v' stands for **vector**. Vector can be said as a **one-dimensional array** in this case because argv stores the pointer to the arguments passed to the program in an array of strings.

Note: argv does not store the actual argument, but the pointer to that argument.

## Examples:

Suppose that we have a .exe executable, named xyz. We are passing two arguments to xyz, i.e., a and b.

```
xyz.exe a b
```

Now the value of argc will be **three** as there are two arguments.

And argv will store **pointers** xyz, a, and b.

To create a command-line utility, we must pass `int argc` and `char const *argv[]` to the program as arguments. Then we can make any kind of programs that take arguments from the terminal to display the result instead of accessing the compiler every time.

We can create many applications using command-line utility such as a calculator, as one of the simplest example.

```
#include <stdio.h>
int main(int argc, char const *argv[])
{
    printf("%d\n", argc);
    return 0;
}
```

In the above example, we are creating a simple command-line utility that will print the number of arguments we provided to it.

The benefit of doing so is that we would not have to open the compiler as we did for every program before this one. We can directly execute the file and provide it with the terminal or PowerShell window parameters, whichever we prefer to use.