

Command Line Arguments

CPE 211

UAHuntsville

Command Line Arguments

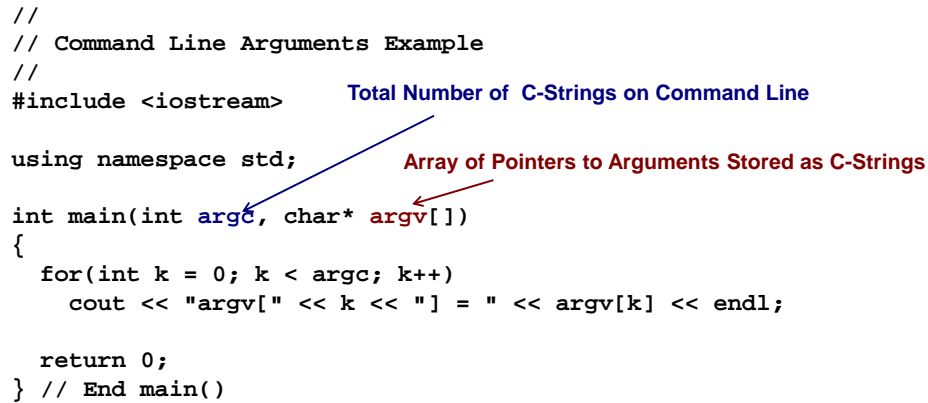
- It is possible to give your program argument values when invoking the program from the terminal window
 - ***Command Line Arguments***
- Many Unix/Linux utility programs make use of this mechanism for the input of options, file names, or directory names
 - Examples:

```
$ mkdir project01
$ cp source.txt destination.txt
$ ls -l
```

UAHuntsville

Array of Pointers Example - 1

```
//  
// Command Line Arguments Example  
//  
#include <iostream> Total Number of C-Strings on Command Line  
  
using namespace std; Array of Pointers to Arguments Stored as C-Strings  
  
int main(int argc, char* argv[])  
{  
    for(int k = 0; k < argc; k++)  
        cout << "argv[" << k << "] = " << argv[k] << endl;  
  
    return 0;  
} // End main()
```



UAHuntsville

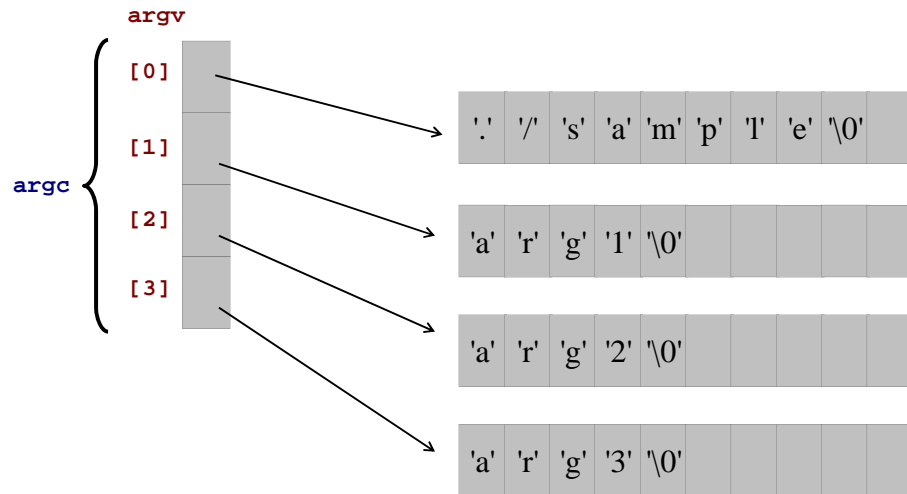
Array of Pointers Example - 2

```
$ g++ main.cpp -o sample  
$ ./sample arg1 arg2 arg3  
argv[0] = ./sample  
argv[1] = arg1  
argv[2] = arg2  
argv[3] = arg3  
$
```

Here the value of argc is 4: 1 for the program executable
and 3 more for the number of arguments

UAHuntsville

Array of Pointers Example - 3



UAHuntsville

Pointer to Pointer Example - 1

```
//  
// Command Line Arguments Example - Revisited  
//  
#include <iostream>  
  
using namespace std;  
  
int main(int argc, char** argv)  
{  
    for(int k = 0; k < argc; k++)  
        cout << "argv[" << k << "] = " << argv[k] << endl;  
  
    return 0;  
} // End main()
```

UAHuntsville

Pointer to Pointer Example - 2

```
$ ./sample2 arg1 arg2 arg3
argv[0] = ./sample2
argv[1] = arg1
argv[2] = arg2
argv[3] = arg3
$
```

Same output as before.

UAHuntsville

C-Style String to C++ String

```
// C-Style String to C++ String
#include <iostream>
#include <string>

using namespace std;

int main(int argc, char** argv)
{
    string value0 = argv[0];
    string value1 = argv[1];

    string value2(argv[2]);
    string value3(argv[3]);

    cout << "value0 = " << value0 << endl;
    cout << "value1 = " << value1 << endl;
    cout << "value2 = " << value2 << endl;
    cout << "value3 = " << value3 << endl;

    return 0;
} // End main()
```

UAHuntsville

C-Style String to C++ String- 2

```
$ ./sample2 arg1 arg2 arg3
value0 = ./sample2
value1 = arg1
value2 = arg2
value3 = arg3
$
```

Same output as before.

UAHuntsville