

CENG328 Operating Systems

Laboratory Chapter 1 Extra

1 Command Line Arguments in C/C++ Programs

Laboratory Chapter 1 had introduced Linux command line utilities that accepted different arguments earlier. Some examples are:

```
ls
ls -l
ls -l -a
sort functions.txt
gcc test.c -o testprogram
```

The keywords that follow the main program names (ls, sort, gcc) in above examples are named **Command Line Arguments**. They are used to input data to programs before runtime. All programs written in C / C++ languages can parse such command line arguments. In order to do this, you must provide two special variables as parameters to main function: one integer and one string array. An example program is given below (arguments.c):

```
#include <stdio.h>

int main(int argc, char** argv) {
    int i;
    for (i = 0; i < argc; ++i) {
        printf("argv[%d] = %s\n", i, argv[i]);
    }
    return 0;
}
```

Compile this program and execute it with as many command line arguments as you want:

```
./arguments
./arguments hello world
./arguments ceng328 "operating systems"
./arguments 2 times 2 is 4
```

What output do you see?

One important thing to pay attention is that **all** arguments are **string** variables, no exceptions. Aside from strings, all integer and floating point numbers entered as command line arguments will still be strings, therefore you must convert the corresponding strings to integers or floats by using **atoi()** and **atof()** functions.

1.1 Exercises

1. Convert the given arguments program such that it behaves the same as the “echo” command line utility.
2. Write a program which takes two integers as command line arguments and prints the sum of these integers.