CS 121
Bruce Bolden
January 22, 2015

Lab Assignment #1
10 Points
Due: February 5, 2015

1 Command Line Processing

Using the command line to pass arguments to programs is very common in the Unix world. Long before there were Graphical User Interface (GUI) based programs, the command line was used to specify program options.

Consider the following simple examples

```
% ls -l
```

```
% make -f app.make
```

In the first command, the `-l` (letter ell) option requests that the *long* form of a directory listing be used. In the second, the `-f app.make` option tells `make` to read the rules defined in `app.make`. We will learn more about `make` later.

1.1 Handling Command Line Arguments

The standard prototype for the main function in C/C++ (and a number of other programming languages) in the Unix environment is:

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

or

```
int main( int argc, char **argv );
```

The first argument, `argc`, holds the number of items in the `argv` array. Note that `argv` contains character strings (an array of characters). This requires numbers entered on the command line to be converted from a character string to the necessary numerical data type.

1.2 Activities

1. Write a small program (`ShowArgs`) to print out all arguments given to a program.
   - What is the first argument? (surprised?)
   - What is the last argument?
2. Revise your program to accept at least three command line arguments.
   - Add error handling code to verify that the program has been supplied with the required minimum number of arguments.
3. Revise your program to convert some of the command line arguments.
   - Convert the second argument to an integer value
   - Convert the third argument to a floating point value
   - What happens when the argument to one or both of these is of the wrong type, e.g., `two` instead of `2`?
4. Modify your program so that it handles simple arithmetic computations, e.g., `calc 2 * 3  -->  6`.
   - Add error handling.
   - Does your program work for all the standard mathematical operators?

1.3 Deliverables

1. Document any issues/problems as you find them (a *programming log*).
2. Argument manipulation program: source code **and** sample output
3. Calculator program: source code **and** sample output
4. Brief answers to the four (4) questions above. They end in question marks.

1.4 Unix References

O'Reilly, "Unix in a Nutshell", 1998

O'Reilly, "Vi Editor", 6th edition, 1998