CST 334 (Operating Systems)

Dr. Glenn Bruns

# Lab: C programming

The purpose of this lab is to get you started with C programming. You will use a text editor, compile a C program, and run it. You’ll also get some practice with the command line.

1. On mlc104, copy /home/CLASSES/brunsglenn/cst334/labs/c-prog-lab/hello.c to your home directory. If you can't access mlc104, use a virtual machine or Cloud 9, and this [hello.c](https://drive.google.com/open?id=1eVByHWretwap8b4TnLHwh87UZqUtxuBk).
2. Compile and run hello.c.
   * to compile: $ gcc -o hello hello.c
   * to run: $ ./hello
3. Copy hello.c to show-inputs.c. At the terminal, you can do this with the 'cp' command, like this:

$ cp hello.c show-inputs.c

Using a text or program editor (like vi, vim, or emacs), modify the first line of the main function from

int main(void) {

to

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

The function parameter argc shows you the number of arguments provided on the command line when the function is called. The parameter argv is an array of the arguments (it is an array of strings).

Compile and run the program. It should give the same output as before.

Now modify show-inputs.c so that it outputs the number of arguments provided on the command line (and doesn’t print ‘hello’). Here’s an example of what the program should do:

$ ./show-inputs foo bar

2 inputs

Use the C statement printf to print the number of inputs. Use man printf and google to find out how to use printf.

1. Now modify show-inputs.c so that it outputs both the number of arguments plus the arguments themselves, one on each line. Here’s an example of what the program should do:

$ ./show-inputs foo bar

2 inputs:

foo

bar

Your program should work with any number of command-line inputs.

1. If you still have time, modify the program so that it prints an error message and exits if no command-line arguments are provided (not counting the command itself).
2. If you still have time, modify the program so that it accepts numbers on the command line, and then prints the sum of the numbers.
3. If you still have time, write a C program to generate a random Boggle board position (a 4x4 matrix of letters), download a dictionary of English words, and generate a list of all words on the board of at least 6 characters, listed in lexicographic order.