CSCE 3600: Systems Programming

Recitation Assignment 1 – ASCII Bits & Bytes

Due: 11:59 PM on Friday, September 10, 2021

PROGRAM DESCRIPTION:

In this recitation assignment, will write a complete C program that will prompt the user to enter an ASCII printable character and then print out its decimal (base 10), hexadecimal (base 16) and all 8 binary digits of its binary (base 2) representation.

You may assume that the user enters a valid ASCII printable character when prompted. Examples of printable characters include A, B, C, #, $, &, +, 0, 1, 2, <, a, b, c, and so forth, and they typically range in decimal value from 33 to 126. You do not have to do any error checking for non-printable characters.

Although not required, you may find Chapter 2 on Bits, Bytes, and Data Types in the *System Programming with C and Unix* optional reference textbook by Adam Hoover to be helpful.

For this recitation assignment, complete the following tasks. You may receive guidance from your TA or fellow students.

1. Prompt the user to enter a printable ASCII character using `printf` and then read in the user's response using `scanf`, storing your character as an `unsigned char`.
2. For the decimal and hexadecimal bases, simply take advantage of the *format specifiers* in `printf` to print the decimal and hexadecimal representations. See `man 3 printf` or any of the reference or tutorial material provided on Canvas for help if you're having trouble with this.
3. Although there are several ways to accomplish printing out the binary representation, you are being asked to accomplish this functionality using bitwise operators as described below:
   a. Since it binary representation can only be 8 digits in length, use a `for` loop that decrements the control variable from 7 down to 0, inclusively.
   b. The right shift operator (`>>`) is used to shift bits to the right. Use this operator to right shift the printable character by the amount in the control variable (i.e., the number being decremented in each iteration of the `for` loop) and store in a temporary integer variable.
   c. Next, the bitwise *and* operator (`&`) will set a bit to 1 if and only if both of the corresponding bits in the operands are 1. For example, if `x=0110` and `y=1011`, then `x&y=0010`. Now, if the bitwise *and* of the temporary integer variable from (b) above and the integral literal value 1 is true, simply print "1" (with no newline) to the terminal; otherwise, print "0" (with no newline). After

all 8 binary digits are printed (including leading 0s), then print a newline to the terminal.

REQUIREMENTS:

- No additional comments are required for this recitation assignment, although you should ensure that your name and EUID is included in your code as comments.
- Your program should be named “`rec01.c`”, without the quotes.
- Your program will be graded based largely on whether it works correctly on the CSE machines (e.g., `cse01`, `cse02`, …, `cse06`), so you should make sure that your program compiles and runs on one of the CSE machines.
- Although this assignment is to be submitted individually (i.e., each student will submit his/her own source code), you may receive assistance from your TA, IA (i.e., Grader), and even other classmates. Please remember that you are ultimately responsible for learning and comprehending this material as the recitation assignments are given in preparation for the minor assignments, which must be completed individually.

SAMPLE OUTPUT (user input shown in **bold**)**:**

```
$ ./a.out
Enter an ASCII character: A
The ASCII value of A is:
    dec -- 65
    hex -- 41
    bin -- 01000001
$ ./a.out
Enter an ASCII character: a
The ASCII value of a is:
    dec -- 97
    hex -- 61
    bin -- 01100001
```

SUBMISSION:

- You will electronically submit your `rec01.c` program file to the **Recitation 1** dropbox in Canvas by the due date and time.