



Intro to C and Binary Numbers

8/27/2007





Opening Discussion

- Let's look at three answers to the interclass question. What are the steps in building a C program?
- Do you have any questions about the class based on what we covered last time?





Linux Command Line

- Let's all log into the machines and spend a few minutes getting used to the Linux command line.
- The PDF file on the web lists a number of different commands in Linux.
- The Linux directory structure is just like the folder structure you are used to in Windows. We will navigate it with text commands instead of clicking.
- Tab completion is your friend as are up and down arrows.
- Command-line execution allows multiple arguments.
- The man command will give you the “manual page” entry for any command. This also works for C-library routines as we will see later.

- When you are programming you can use any text editor that you want to. Text editors, unlike word processors, edit and save straight text. For comparison, open a .doc file in notepad to see all the extra stuff that Word throws into a file.
- I use vi and teach it in this class. It is a powerful programming editor that is ubiquitous on Linux/Unix systems.
- Start vi by entering vi and the name of the file you want to edit on the command line.
- There are two modes in vi. You start off in command mode. Several commands put you an editing mode. Esc goes back to command mode.



Simple C Program

- Lets go into vi together and edit our first program. We will call it hello.c. All of your C programs will be in files that end with .c.





Structure of a C Program

- All of your C programs will have several parts.
- The top part is pre-processor directives. These begin with a # sign. They are handled in a processing step before the compilation of the program.
- Next you can declare global identifiers. I will strongly limit the use of global variables in this class and generally code in a way where you won't use this section.
- Below the global identifiers are your methods. Your book puts main as the first method. I will make it last.



- Comments are pieces of text that are thrown out when a program is compiled. These are things you put in for the benefit of humans.
- You will want to put your name and a pledge of your code in a comment at the top of each assignment.
- Standard C comments are text between `/*` and `*/`. They can span multiple lines.
- C99 allows C++ style single-line comments of the form `//`. If you use these you need to put in an extra compiler flag saying you are using the C99 standard. I generally won't do this.



- Many of the things you put in your C programs will need names. These names are called identifiers and they have certain rules.
- Identifiers can contain letters, numbers, and underscores. The first character can't be a number.
- C is case sensitive so IDENT and ident are two different identifiers in C.
- Identifiers beginning with an underscore are typically used by the system so we won't name anything that way.
- I will typically follow the “camel” naming scheme.



Compiling and Executing

- We compile our programs with the gcc compiler. Simply specify gcc and the name of the .c file you want to compile.
- Many different options can be specified as well. I recommend including -Wall and -pedantic for stronger error checking. If you want to use C99 features use -std=c99.
- By default this will make an executable file called a.out that you can run. The -o option allows you to specify a different name to use for the executable.



Binary Numbers

- We like to use the decimal (base 10) number system, but numbers can be done in any positive integer base.
- Because of the simplicity in making the electronics, computers typically use a binary (base 2) system.
- In binary each digit is a power of two and they will have either a 0 or a 1 in it.





Converting Decimal to Binary

- One way to convert binary to decimal is to find the largest power of two smaller than the number and subtract that out. That will be a one bit. Each power of two you skip is a zero bit.
- Many people prefer the “divide by two” method. I'll write it here as an algorithm.
 - ◆ `while(n>0)`
 - if(n is odd) write a 1
 - else write a 0
 - Divide n by 2 throwing away fraction
- This second method is related to bit shifting which we will talk about next week.



Hexadecimal and Octal Numbers

- Computers might work in binary, but writing numbers with only 1s and 0s can be a real pain.
- Hexadecimal (base 16) and octal (base 8) are common substitutes. They are closer to binary, but don't take nearly so many digits.
- Hex numbers have 0-9 and A-F. Octal contains only 0-7.
- Converting from binary to hex or octal is as simple as grouping together bit.
- Starting at the ones bit, group bits in groups of 4 for hex and groups of 3 for octal.



- In what ways can command line processing be superior to GUIs?
- Interclass Problem : Convert the following numbers from decimal to binary (8 bit), hex (2 digit), and octal (3 digit). Write your answers on these systems with vi and show some work.
 - ♦ 23
 - ♦ 137

