Binary Numbers and C

8-29-2006

Opening Discussion

- What did we talk about last class?
- Do you have any questions about the reading?
- Anyone interested in majoring in CS should sign up for the Survey course, CSCI 1194. Slots will be made for you.

Logging In

- Each of you should have gotten an e-mail with your password for the Linux account. I want you to go ahead and log in with that password. Your username is the one you normally use.
- If you didn't get a password that means you already have a Linux account from a previous class. If you don't remember your password let me know and I can get it reset before next class.

Binary Numbers

- Last time we started talking about binary numbers. You did some reading on them as well.
- I want you to break into groups of 2-3 and convert the following 8-bit binary numbers to decimal.
  - 11010110
  - 01101011
  - 11111111
- How do we represent fractional numbers in binary? What are the meanings of the bits?
- Now convert these numbers in your group.
  - 1101.0101
  - 0110.1100

Decimal to Binary

- Now what about converting decimal to binary? There are two ways to do this.
- Find the largest power of 2 smaller than the value and subtract it out. That position gets a 1. Repeat until you get to 0. Every bit that isn't a 1 is a 0. This builds from the top bit down.
- You can go from the bottom up by noticing that all odd numbers have a 1 in the lowest bit and division by 2 simply shifts the number.
- This second approach works well for fractions as well except you multiply instead of dividing.
- Let's do a few of these together.

Decimal to Binary Exercises

- I want you to convert the following numbers into binary numbers. For fractional values stop at 3 bits.
  - 117
  - 86
  - 49
  - 30,276
  - 29.45
  - 37.16

Octal and Hexadecimal

- Using binary is nice because it relates to machine representation, but it requires a lot of digits. For this reason octal and hexadecimal came into use.
- Because 8 and 16 are powers of 2, it is easy to convert from them to binary and back. All you have to do is group bits together.
- 3 binary bits is one octal character. 4 binary bits is one hex character.
- Convert the following to octal and hex.
  - 11010010 (binary)
  - 137 (decimal)
- Convert these to binary and decimal.
  - 0273 (octal)
  - 0xA8 (hex)

Working in Linux

- Now that we've played with binary some, let's play around in the Linux command-line.
- The first thing we have to do is open up a terminal window. This will let us type commands into the computer.
- You will start off in your home directory.
- Lets use our list of Linux commands to “look” around the directory and see what we have.
- Directories help keep things organized. Each of you should create a directory for this class where you will put things related to it. The directory could be called something like pad1, PAD1, CSCI1320, LewisMakesMeCry, etc.

Getting to Linux from Dorms

- One of the advantage of Linux is that it can be accessed remotely. You don't want to do that with these machines because they are dual-boot and people are likely to reboot them. However, we have a number of Linux only machines that you can safely log into from your dorm room and work on. (Xena01-Xena21)
- To connect to those machines you can use either Putty or a Linux terminal like Cygwin. There are links to both of these on the links page. Putty is simpler and just gives you a connection. Cygwin makes your Windows machine act like a Linux box.

Working with vi

- Now that we have moved around in Linux some, it is time to move to editing text files. I will ask you to use vi as your text editor in this class. It is a nice text editor for programming because it was built for that purpose.
- First we should use vi to edit some of the settings of vi. This will make vi more functional for coding.
- vi has two modes: command mode and edit mode (technically there are several types of edit mode). You start off in command mode. Typing in command mode doesn't actually type, it gives commands. Certain commands cause you to enter an edit mode. Press esc to get back to command mode.

Writing a C Program

- Now I want you all to go into your class directory and use vi to edit a file for a C program. You can call the file whatever you want, but it needs to end in ".c".
- Let's write a program that lets you input two integers and then prints out their sum, difference, product, and quotient.

Math on Computers

- We've talked about the fact that computers use binary numbers for representing everything. Your book even went into some detail on the different types of numbers which we will hit on more next class.
- How would you do some of your standard math operations using binary numbers? Can you do addition in binary? What about multiplication?
- Let's look at these two operations first. Subtraction will wait until next class. How about division?
- What types of operations are really simple in binary?

Minute Essay

- How do you think negative numbers are stored in computer memory given that you only have 1 and 0, no negative signs?
- Remember to let me know if you didn't get a password for a Linux account.