Machine Arithmetic, Characters, and Strings

1-27-2012

Opening Discussion

- ACM tutoring is MTWR 3:30-5:00 in HAS 329.
- Minute essay comments:
  - Why learn binary representation of numbers?
  - You are learning a different language.
  - Negative numbers vs. large numbers.
  - Is there a number you can represent as positive but not negative in 2s-complment?
  - Who/what uses hex?
  - Binary is not the same as log base 2. It is just base 2.

More

- Examples of piping.
- Meaning of literal.
- Quiz format: always EC, some different question styles will appear.
- Bad quiz grades.
- Adding binary number in Scala.
- Difference between 16-bit, 32-bit, and 64-bit computers.
- Difference between J and Scala.
- I hope you never become fluent enough in binary and unicode to read it.

Even More

- Not everyone thinks we are going slow.
- Control over output to force fewer digits on Double.
- We will code in vi and test in REPL. The vi code will be run as a script.

Hexadecimal

- Binary is unwieldy for humans because of the large number of digits.
- Hexadecimal (base 16) is commonly used because it converts nicely to binary, but has few digits.
- Four bits is a hex digit. Start at the right and group bits by 4.
- Use letters A-F for numbers 10-15.
- Hex literals start with 0x
- toHexString

Octal

- Octal (base 8) is less common than hex, but not uncommon.
- Group bits into groups of three.
- Octal literals and toOctalString().

The math Object

- For other math functions use methods on the math object.
- For example, use math.sqrt() to take the square root of a number.

Characters

- The Char type represents a single character in Scala.
- The literal for Char has the letter that you want in single quotes.
- The Char is stored in the computer as a 16-bit unsigned integer encoded in Unicode.
- Unicode has the alphabet of every written language in it.
- You can convert to an Int to see the numeric values of characters.

Escape Characters

- Not all characters can be easily entered. For things you can't nicely type, use escape characters.
  - \n – for a new line
  - \t – for a tab
  - \” - to get a double quote
  - \' - to get a single quote
  - \\ - to get a backslash

Strings

- We have seen the String type and that re represent String literals by putting characters in double quotes.
- Escape characters can also go inside of normal strings.
- Strings have many methods. We can see the basics using tab completion. (If we put in some extra parentheses.)

Raw Strings

- There are some situations when using escape characters is a pain.
- For this, use triple double quotes to make a raw string.
- Anything you type between the triple double quotes will go into the string.
- They can span multiple lines even.

Variables

- It is very common to want to represent values with names.
- A variable is a name that we use to represent a value.
- In Scala we can declare variables using val or var.
  - val *name*:*Type* = *expression*
  - var *name:Type* = *expression*
- A val can't change it's value, a var can.
- The colon and type are generally optional.

Tuples

- Another type in Scala is the Tuple type.
- A tuple has comma separated values in parentheses.
- They give us a way to handle a fixed set of associated values.
- Assignment into a tuple does pattern matching.

Scripts

- We have spent most of our time in the REPL entering one statement at a time.
- When we want to do things repeatedly it is nice to put the commands in a file called a script.
- We can use vi to write a script and put Scala commands in the file.
- We can run it by specifying the file name after the command scala.
- Alternately, we can load it into the REPL.

Sequential Execution

- When you put commands into a script, they are normally executed one after the other from top to bottom.
- This is what we call sequential execution and it is the default way things happen.
- Order can be very significant for the instructions in a program.

Standard Input

- Scala provides a whole set of functions you can call to read from standard input.
  - readInt()
  - readDouble()
  - readLine()
  - And many more
- This can make your scripts far more useful as they can be used with different values each time you run them.

Some Problems

- We are still a little limited in that we can only do simple math and formatting, but let's try to do some things with that in scripts.
- We'll start with formatting money.
- Calculating hourly wages?
- Taking averages of grades?
- Suggestions?

Minute Essay

- What questions do you have about the topics we have been working on?