Assembly Language: Raspberry Pi

Chapter 2 Notes

Date: 9/23/16

Starting out

**ARM Chip**

* Def.) Stands for Advanced RISC Machine and is the CPU of the Raspberry Pi
* Receives and decodes an endless stream of binary 1’s and 0’s to process instructions and data.

**Assembly**

* Def.) The compiler for the Raspberry Pi.
* Stands just a level above machine code.
* Called input or source file

**ARM Instructions**

* Keywords and code are written in mnemonics(shorthand txt)
  + These instructions are referred to as operation codes or “opcodes”
  + Not compatible with other languages
* Microprocessors move and manipulate data and is used consistently throughout all languages
  + Ex) SUB and ADD keywords take values and calculates them before returning the result.

**Transformation Process**

* An *assembler* converts the language program into machine code to be processed.
  + Since it is converted to machine code, the program runs many times more efficiently.
  + It is also possible to retain the original source file for upgrading and modifying.
* Biggest advantage of using GCC is that it can also assemble C programs.

**RISC and Instruction Sets**

**Keywords**

* ADD
  + Adds values
* SUB
  + Subtracts values
* MOV
  + Move command that takes information from one place and sets it in another.