# Assembler: The Anatomy of a Source File

Let's break down the following source file:

```
        .global _start
_start:
        MOV R0,    #65
        MOV R7,    #1
        SWI 0
```

The starting point of an assembly source file is the following label (on line 2):

**_start:**

Therefore, the first line of the source file:

**.global _start**

is what will define the "_start" label that you see in line 2 as a GLOBAL name, meaning that "_start" is available to the whole program.

Every single assembler source file will consist of a sequence of statements, one per line, in the following format:

<label:>

<instruction(s)> @ comment(s)

Instructions cannot be placed before a label or your program will not execute successfully. You can have multiple sets of labels and multiple sets of instructions in a program, and you will see this during the course of class.

Comments are used to document what is happening in the program. These are very important to include in every program. Make sure to comment at key points in the program. Also, use plain English (not symbols or shorthand) to explain what is going on. The assembler will see the @ symbol and ignore everything after it until the end of the line.

**_start:**

The second line defines where the program should begin its execution. Omitting this label will cause a warning message when linking the program, and this could cause the output of the execution to be completely incorrect. Always, *ALWAYS* use this label on line 2 of the assembler source file so that the assembler knows exactly where to start execution.

The use of the colon ( **:** ) at the end of the second line denotes that this is a label. It is best practice to use *lowercase* letters for label names to identify them in the program.

The next 3 lines of the source file are assembly language commands. Let us look at each of the lines separately.

**MOV R0, #65**

The MOV stands for “move”, R0 denotes the number of the register – a special place on the ARM chip (in this case, register 0), and #65 means the literal number value 65. This line of assembler is saying “***MOVE the value of 65 to Register 0***”.

It is good to note here that whatever value comes after the pound ( **#** ) sign is taken literally.

Also, in assembly language, it is best practice to ensure that all commands and all registers are written in *UPPERCASE* letters. This way, commands and registers are

easy to identify, and stand out from label names, which as stated on page 1 of this document, are in lowercase letters.

**MOV R7, #1**

Again, we see the MOV command, we see a different register, R7, being used, and the literal value 1. This line of assembler is saying "***MOVE the value of 1 to Register 7***".

**SWI 0**

This command is called a "software interrupt". This special command is used to call the Raspbian Operating System. In this particular source program, this command is used to exit the assembler program and return control back to the command line prompt.

The programs in class will get more complex but will follow this basic anatomy.