COAL LAB

Assignment 01

Submitted by: Abubaker

Submitted To: Ma’am Fizza Asif

Roll Number: I211379

Section: H

Report: Assembly Language

History:

- Assembly Language first introduced in 1949 and quickly became widely used in automatic calculators and electronic delay storage. A low-level computer language called Assembly was used to simplify the language of machine code, or the precise instructionsneeded to run a computer.
- Intermetrics created the programming language HAL (High-order Assembly Language) for NASA in the early 1970s. For onboard spaceflight applications at the time, HAL/S is recognized as one of the real-time, high-level programming languages. HAL was used to write 85% of the programs used on the Space Shuttle.

Use of Assembly Language:

Assembly language is still used today to directly manipulate hardware, gain access to specific processor instructions, or solve serious performance problems. Device drivers, low-level embedded systems, and real-time systems are examples of typical uses. Games! Even if it'simpossible today, Rollercoaster Tycoon was entirely assembled. Understanding how a language, library, or piece of code operates is incredibly helpful. For purposes of troubleshooting, it's really helpful.

To What Extend it is in Used:

It is used in all the fields where computations are performed, Like Embedded systems, Supercomputing, Space Science, Networking, Telecom.

Assembler:

It consists of Assembler which is know as translator for conversion of code into machine language. In short it produces binary code in form of 0 and 1.

Example:

GAS, GNU etc.

Structure Of Assembly Language:

A series of statements, called directives, or assembly language instructions like ADD and MOV, make up an assembly language program.

A **directive**, sometimes known as a pseudo-instruction, gives the assembler instructions, whereas an instruction instructs the CPU on what to do. For instance, the CPU executes the ADD and MOV instructions, whereas ORG and END are assembler directives. When the ORG command is used, the assembler places the opcode at memory location 0, and END designates the end of the source code.

The following four fields make up a program language instruction:

[label:] mnemonics [operands] [; comments]

The **label field** enables a line of code to be named in the program. There is a maximum character limit for the label fields.

Together, the fields for **mnemonics** and **operands** carry out the actual work of the program and complete the tasks. Statements like ADD A, C & MOV C, #68 yield opcodes; "A, C" and "C, #68" are operands. ADD and MOV are the mnemonics in these statements. Directives might be present in these two fields. While instructions are converted into machine code for the CPU to execute, directives are solely used by the assembler and do not.

A **semicolon**, which serves as a comment signal, begins the comment area.

Whenever a **label** refers to an instruction, a colon should come after it.