

Assembly Language 2

[30/12/2022]

mnemonic: having some commands that can be called to perform some functions, instead of having to code those functions in binary.

Advantages of assembly languages

- makes it easy to interact with operating system, processors
- it shows how data is being represented on the memory
- allows processor to access and execute instruction
- makes instruction easier for computer

Disadvantages of assembly languages:

One needs to know the architecture of the machine, so one has to study hardware before

Depended on the instruction set and the architecture of the processor, but 32bit system is the most general.

Different environments:

- IBM
- Linux
- Microsoft Assembler

We'll use Microsoft Assembler, or NASM, Dosbox

Syntax of Assembly

Three sectors of assembly:

- Data Section
- BSS Section
- Text Section
- **Data section:** declare constants, filenames, buffer size. to be used in the program, whatever is declared does not change at runtime.

```
section.data  
or  
.data
```

- **BSS Section:** where variables about the processors,

```
section.bss  
.bss
```

- **Text Section:** where the actual code is, it must start with the declaration of the global.

```
.start
```

Comments are used for proper documentation of the code. In assembly the comment is denoted by `;`

Statements in assembly language:

Consists of three types of statements:

- Executable
 - Assembler Directives
 - Macro
- **Executable:** statements that executes into action that performs an operation in a program. generates an executable code
 - **Directives:** Tells the assembler more information about the program/instruction the assembler is running. Not generate machine language instruction.
 - **Macro:** Is a function, consists of both executable and directives, they are a text substitution mechanism. Used for abstraction of repeated operations
-

Format of an assembly language instruction

A statement per line and should be in the format:

- label
- mnemonic or opcode
- operand
- Comment

```
[LABEL][MNEMONIC/OPCODE][OPERAND][COMMENT]
```

What is label: It is used to identify a line of code, helps to be used in a macro

```
ADD Ax, Bx, 2; Add constant 2 to the content in Bx and store it in Ax
```

```
DEC Ax; decrease the content of Ax by 1
```

What are identifiers: Variables defined by programmers to hold some memory locations to be used to store a value. Can be variable, constants, code labels, procedure.

Rules for variable names:

- Can contain 1-27 characters
- Not case sensitive
- First character must be a letter or underscore
- cannot be the same as reserved words such as register name
- number from 1-9

1h means 1 in hexadecimal

Characters are enclosed by a **single quote** " while *strings* are enclosed by *double* " "

≡ Example

Simple statement that adds 1000h & 2000h and subtracts 800h

```
move Ax, 1000h
```

```
add Ax, 2000h
```

```
sub Ax, 800h
```

or

```
add Ax, 1000h, 2000h
```

```
sub Ax, 800h
```