2.1. The process of translating a program written in a High-Level Language (HLL) into an executable file ready for execution involves several steps:

- Preprocessing: This involves handling preprocessor directives such as #include and #define.
- Compilation: This step translates the preprocessed source code into assembly language code.
- Assembly: This step translates the assembly language code into machine code or object code, which is then linked with libraries to create an executable file.

The programs used in these steps are:

- Compiler: The compiler performs the compilation step. It takes the preprocessed source code and generates an assembly language file.
- Linker: The linker performs the linking step. It takes the object code and links it with libraries to create an executable file.
- The Operating System (OS) uses the program loader to load the executable file into memory and run it.

2.2. A MIPS assembly language statement typically has the following elements:

- Instruction mnemonic: Specifies the instruction to be executed.
- Destination register: Specifies where the result of the instruction is stored.
- Source operands: Specifies the values to be used in the instruction.
- Comment: Provides additional information for the programmer and is ignored by the assembler.
- Optional elements of a MIPS assembly language statement include labels and immediate values.