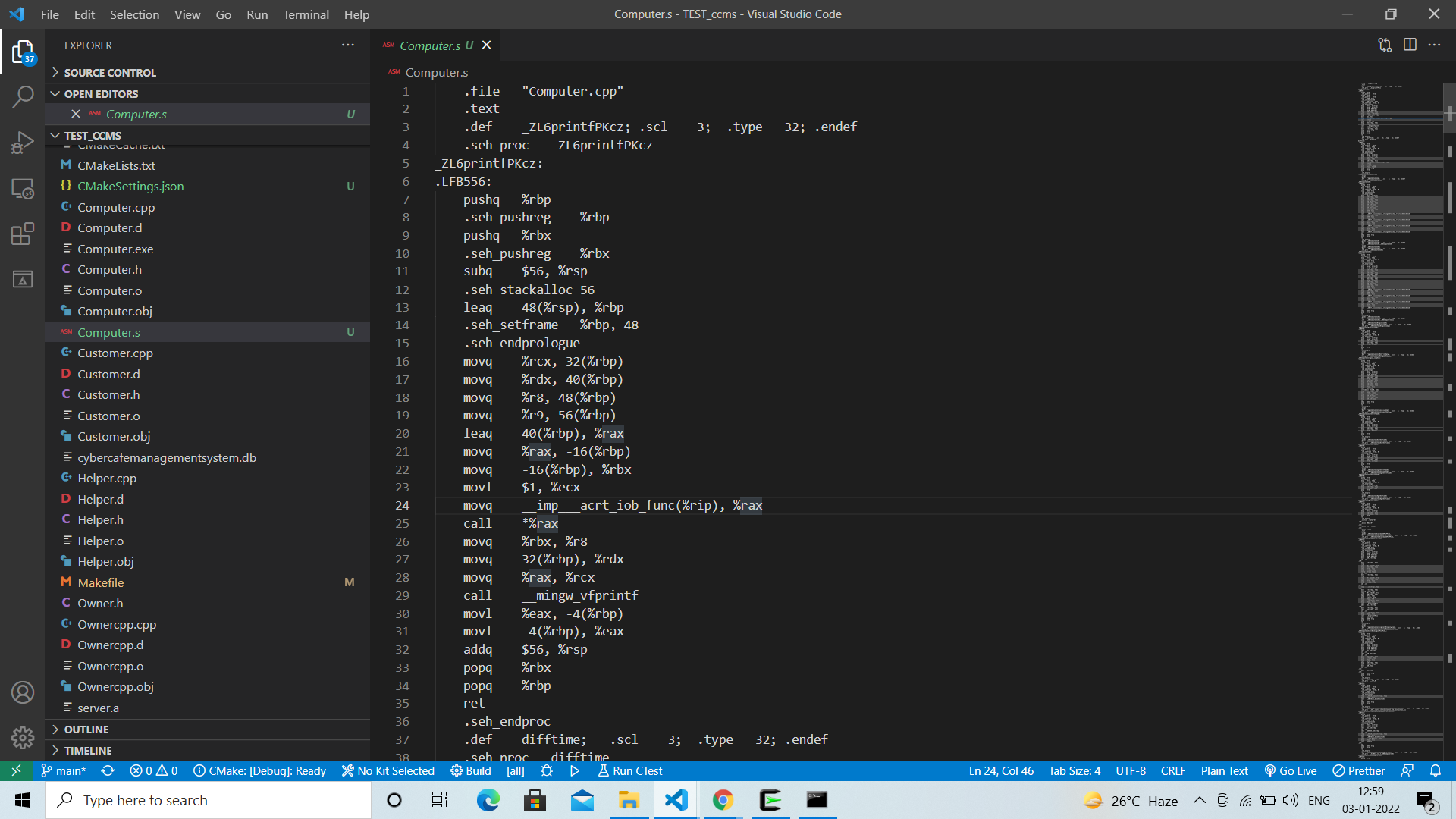
**Assembly Code**

An assembly language is a low-level programming language designed for a specific type of processor. It may be produced by compiling source code from a high-level programming language (such as C/C++) but can also be written from scratch. Assembly code can be converted to machine code using an assembler.

Since most compilers convert source code directly to machine code, software developers often create programs without using assembly language. However, in some cases, assembly code can be used to fine-tune a program. For example, a programmer may write a specific process in assembly language to make sure it functions as efficiently as possible.

* **Commands**

g++ -S filename.cpp



Each indented line in the above code corresponds to a single machine instruction. For example, the **pushq** instruction indicates that the contents of register**%rbp** should be pushed onto the program stack. All information about local variable names or data types has been stripped away.

* **Description**

While assembly languages differ between processor architectures, they often include similar instructions and operators. Below are some examples of instructions :

* MOV - move data from one location to another
* ADD - add two values
* SUB - subtract a value from another value
* PUSH - push data onto a stack
* POP - pop data from a stack
* JMP - jump to another location
* INT - interrupt a process