



Topic: 1.3.7 High- and low-level languages and their translators

Software is created by using a computer language to set out a set of instructions that the computer will follow. There are many computer languages but they do share some common features.

CONVERTORS (translators):

Every high level language source's code should be converted to low level language so that computer's microprocessor can understand and execute that.

For that reason every language comes with its own interpreter, which is either a compiler or a interpreter.

INTERPRETER:

- Is a program which only works with a source code and do not produce an independent executable file.
- Interpreter converts each source code's line and executes it.
- Interpreter works within emulator and doesn't work directly over two microprocessor. This saves computer from crashing when the program performs any illegal operation.
- It help's programmers for the error correction while writing programs and testing them.

COMPILER :

- It reads the whole program and generates an executable file (exe), which can be easily distributed without the fear of source code being copied.
- Debugging is tough with compilers as they convert the whole program and gives out all the errors in that program compiler runs directly over the microprocessor, thus can crash the computer.

ASSEMBLY LANGUAGE Assembler:

- Every new architecture has its own assembly language
- Any source code that is written for that particular architecture should be translated to machine language through a translator made in assembly for that particular architecture called Assembler.





Topic: 1.3.7 High- and low-level languages and their translators

Difference between Compilers & Interpreters:

- Interpreter reads one line of source code at a time, converts it to machine understanding form and execute it.
- Compiler takes the source code as a whole, converts it to machine language and generates an object (machine) file that can be executed on users wish without compilation.

Distribution:

- Interpreter doesn't work without the source code. So, intellectual property rights infringement is possible, i.e. anyone having source code can copy and further distribute it under their own name.
- Compiler on other the other hand generates an executable file that when distributed does not require source code and can be executed independently.

Shell:

- Interpreter doesn't work directly over the microprocessor but in a shell (emulator) that mimics the microprocessor and doesn't hang the computer when a program malfunctions.
- Compiler works directly with the microprocessor and can hang the pc.

Debugging:

- Interpreter stops at the line where error occurs and compiler continues and generates many errors.

