Monday 15th November 2021

Interpreters and Compilers

Translation

Programs written in a high-level language must be translated into machine code before it can be executed by the processor. Compilers, interpreters, and machine code are examples of translators.

Compilers

A compiler translates the source code into a machine code program (object code) which can then be executed by the processor.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| The translation is only done once, and as a separate process. | *If errors are encountered, the translation continues, and reports the error at the end.* |
| The program is translated into machine code, which is executed quickly. | *The translated program cannot be changed, only the source code.* |
| The program is protected from competitors, who cannot see the original source code. |  |

Interpreters

An interpreter translated the higher-level code line by line into machine code each time the program is run.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| When an error is found, the interpreter stops, and pinpoints the error | *Every line has to be translated every time the program is run.* |
| The code is not platform-specific and can be run on different OS’s as long as there is an interpreter |  |
| The program can be easily edited, as it always exists as source code |  |

Assemblers

Assemblers translate the mnemonics (short keywords) of assembly code into machine code instructions. Assembly language is very similar to machine-code, and one assembly instruction translates to one machine-code instruction.