Text

Programming Languages

Section 21.4

Completed

< back to module

Programming languages are often separated into the categories of high-level and low-level languages. A low-level programming language is closer to how the processor "thinks", which means that low-level languages are quite difficult to learn. A high-level programming language is just the opposite: the code you write in a high-level language is more abstract from the machine code that it produces. High-level languages are more easily human-readable since much of the complexity is removed and later added back in automatically by the compiler or interpreter.

So why would we use low-level programming languages when we have high-level ones? High-level programming languages remove a lot of the complexity, but that complexity still exists - it's just hidden from the programmer. This means there is a cost in efficiency. A programmer using a low-level language has more control over the details of how a processor does things which a programmer using a high-level language could never achieve.

Both low-level and high-level programming language can be useful, depending on the situation. If you want to write and debug a program quickly, and you don't particularly care about the low-level performance details, use a high-level language. If on the other hand your program needs to maximise performance (such as an operating system does, like Windows or Linux), then you should likely choose a low-level language.

In CyberStart Essentials, we're going to focus on two programming languages: Python and C.

Python

Python is a high-level programming language. While Python can be compiled, it's usually used as an interpreted language. It's considered very easy to use and very powerful as a programming language, which is one of the reasons it's very popular in the cyber security community. Many security tools are written in this language.

C

C isn't strictly a low-level programming language, but it's close. Many people consider C to be the mother of all programming languages because a lot of other languages (particularly high-level languages) are written in C. (For example, the Python interpreter is written in C.) C is an important language to understand, even if you aren't very good at writing in it, because it allows us to explore aspects of programming that high-level languages hide from us, such as memory management.

Completed

< back to module

Copyright 2020 SANS. Version 1.8.1