THE C PROGRAMMING LANGUAGE AND THE CAREER LADDER OF A PROGRAMMER

Artem Puzenko

Educational and Scientific Nuclear and Heat Power Engineering,
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic
Institute"

Language C is a general-purpose programming language that is flexible and easy to use and extremely popular. It is a structured, machine-independent programming language to compile programs for such operating systems as windows, for compiling other application programs such as an Oracle database, Git, a Python interpreter, etc.

This programming language is a compiled language. A compiler is needed to translate a program's source code into executable binary files containing machine code. After compilation, the linker combines the various object files and creates a single executable file to run the application.

It is difficult to find a programming language as easy to use and work with as c language. 'C' is the base language for many programming languages. Therefore, learning C as a primary language will play an important role when learning other programming languages. The "C" language shares the same concepts as data types, operators, control statements, and more. 'C' can be widely used in various applications. In today's market, there are many jobs for a "C" developer.

Upon its introduction, the language was well received because it allowed the rapid creation of compilers for new platforms, and also allowed programmers to be fairly accurate in how their programs were executed. Due to its proximity to low-

level languages, C programs ran more efficiently than those written in many other high-level languages, and only hand-optimized assembly language code could run even faster, because it gave full control over the machine. To date, the development of compilers and the complication of processors has led to the fact that hand-written assembly code (except perhaps for very short programs) has practically no advantage over compiler-generated code, while C continues to be one of the most efficient high-level languages.

Junior is an inexperienced and novice programmer. Many of them have just finished their studies and found their first full-time job. Often, they want the code to simply perform its functions. They equate the current program with a good one. It is difficult to create simple code, and it should not be expected from junior developers - usually their programs are quite sophisticated. Junior can be recognized by the intricate one-line code and overly complex abstraction. With their help, they prove themselves by showing their colleagues how well they know how to write code. And this is wrong, because this approach makes it difficult for other people.

The main requirement for a middle developer is the ability to independently perform the tasks set before him. It is very similar to what was written in the previous paragraph, but there is an important nuance - the word "technical" is missing here. That is, at a new level, you need to understand business requirements and be able to translate them into technical solutions.

Senior writes simple, understandable and maybe even stupid code. And this is one of the most important advantages of a programmer. A senior think about code differently than a junior: programs created by an experienced specialist are easy to maintain and scale.

References:

How the C language works? (2019) C Programming Preview Language. Retrieved from: http:// https://pdfcoffee.com/c-programming-preview-pdf-free.html/

Why learns C? (2022) C Programming Language. Retrieved from: https://topperworld.in/c-programming-language/