

Selection and Improvement of Computer Programming Language

Jie Zhang

College of Information Engineering, Sichuan Agricultural University, Sichuan, 625000, China

Keywords: Computer, programming language, selection and improvement, maximum efficiency.

Abstract: Use frequency of computer programming language has become higher along with the development of computer technology. Computer programming language has been improved remarkably in operation and other aspects after several stages of development. This provides great convenience for programmers while bringing problems in selection from multiple programming languages. In the work, development and characteristics of programming language were analyzed according to the requirements of programming language users. Therefore, the selection and improvement of computer programming language were further studied to provide reference for this field.

1 INTRODUCTION

Appropriate selection of programming language directly influences the improvement of programmers' working efficiency and the achievement of maximum efficiency. Major aspects should be considered in the selection of programming language, including characteristics of programming language, job requirements and mastering situation of programmers.

2 PROGRAMMING LANGUAGES

2.1 Current Situation of Programming Language

Computer programming language is the special language used by programmers in program writing. Actually, it is a group of grammar rules defining computer program. Its purpose is to give instructions to computer through standard human-machine interaction rules, thus the computer can be used by humans. Users can determine data of computer and actions to be taken with the help of programming language (Deng Desen, 2011).

After decades of development, lots of programming languages have experienced invention, replacement, modification or integration. People tried to create a universal programming language,

but they have failed for many times. There are many factors leading to long-term coexistence of different programming languages, including different original intention of programming, knowledge of programming languages, and different running costs of programs.

2.2 Development of Programming Languages

2.2.1 Machine language

At the early stage of computer development, binary data consisting of "0" and "1" was widely used in programming due to technical limitations. Binary data could be directly recognized by CPU, which was called machine language. Machine language is still the only language can be directly recognized by hardware. Obviously, direct use of machine language in programming brings huge workload and difficulties for general programmers. Therefore, machine language can't adapt to the requirement of current computer technology with fast development.

2.2.2 Assembly language

After emerging and being applied in the early 1950s, assembly language has made great progress compared with machine language. It expresses machine instructions with mnemonics, which is much clearer than machine language. However,

assembly language is directly matched to machine instruction, leading to weak statement function. Thus, the workload is still heavy in writing large programs. Due to strong dependency on hardware, assembly language affects development of computer programming toward the direction of convenience and speediness.

2.2.3 Advanced language and algorithmic language

As the third-generation language, advanced language and algorithmic language directly use natural language and mathematical language to write program code. They make programming more intuitive and simple. Advanced language is a language for process, also known as procedural language (Shao Yuzhou, 2001). When using this language for programming, users need not understand internal logic of the computer. Programs can be written only through telling execution steps of problem solving to the computer. Thus, compared with the former two languages, it is more convenient and quick to use advanced language in programming.

2.2.4 Non-procedural language

For the third-generation procedural language, programming can be achieved only through telling execution steps to the computer. Then the execution steps can be skipped in the application of non-procedural language. Using non-procedural language for programming, users only need to instruct work objectives and conditions to the computer. Other tedious programming jobs are operated by the computer system, thus making the programming easier and more convenient. Of course, as the fourth-generation language, non-procedural language also has limitations. It can only be applied to certain areas, so its versatility and flexibility are weaker than the third-generation language.

2.2.5 Intelligent language

The fifth-generation language, also known as intelligent language, has incomparable advantages compared with the fourth-generation language. For example, the fourth-generation language has difficulty in abstract problem solving, while intelligent language successfully overcomes the weakness of the fourth-generation language. To some extent, the fifth-generation language is intelligent and highly efficient in these fields.

2.3 C/C++ language and Java language

Currently, C/C++ language and the Java language are the most popular languages that have been widely used.

2.3.1 C/C++ language

With advantages of advanced language and assembly language, C language has been widely applied to various types of computer during the 1970s and 1980. It has been used in program writing of system application and that without relying on computer hardware. In addition, C language has stronger ability of data processing than other programming languages. As the advanced computer programming language based on C language, C++ language has wider application.

2.3.2 Java language

As a pure object-oriented programming language, Java has some core technologies of C++. Meanwhile, Java abandons contents of C/C++ language that do not conform to its requirements. Therefore, Java language has been very popular since it emerged in 1995.

3 SELECTION OF COMPUTER PROGRAMMING LANGUAGE

Programmers should have certain understanding of various programming languages to select a suitable one from them. After understanding the characteristics of various programming languages mentioned above, programmers should make reasonable selection based on the needs of work and their actual situation.

3.1 Selecting suitable computer programming language according to needs of work

The above analysis indicates that computer programming languages have experienced decades of development with many changes. However, various programming languages currently coexist without any kind of programming language dominating in all areas. Thus, the selection of computer programming language should be based on specific characteristics of the work.

In selecting appropriate computer programming language, programmers should consider contents,

requirements and complexity of current work, as well as the requirements for hardware and system. Otherwise, it is easy to cause serious errors affecting work efficiency, thus hindering the achievement of maximum efficiency.

Different computer programming languages should be selected according to specific requirements. For example, when the program has strict performance requirements, or it requires to be closely integrated with the system, C language should be selected in programming. This is because C language has the characteristics of advanced language and assembly language. C/C++ language should be selected in programming if there are high requirements for performance and ability of program, and the ability of operating system should be fully played. Further, system environment should be considered in the selection. VC is suitable for Windows environment, while gcc is suitable for Unix-like environment. C language can be used for the programming with lower requirement for combination of operating system. If cross-platform programming is required, programmers can choose cross-platform library such as fox in the use of C language. When programmers write commonly-used programs, they can select Java language. Java language can be used to write programs with requirements for cross-platform and broad supports. Lisp will be a preferred choice in writing flexible and fuzzy programs.

In addition, many computer programming languages are developed for specific purposes. Thus, different work contents should also be finished through different languages. For example, Perl or Ruby is applicable to writing text processing program. Vba can be the proper programming language in writing office programs. In writing server-side programs, programmers should choose PHP, CGI, ASP, and JSP.

3.2 Selecting computer programming language based on programmers' knowledge about programming languages

Among different languages, programmers should select languages suitable for them. In order to find the most suitable programming language, the selection should be based on the requirement of programs and characteristics of languages. In addition, their own actual situations should be considered to select the language that can be well used and grasped.

For example, PHP is the best choice in writing server-side programs. However, if the programmer is not good at PHP language, he can select CGI, ASP and JSP language to exert his strengths. For another example, if the programmer without mastering other languages is required to dynamically explain the execution language, what should he do? If he is a C programmer, pike can be used; and he can choose beanshell if he is a Java programmer.

Therefore, in the selection of computer programming languages, programmers should consider work contents and requirements; in addition, they should select corresponding programming language based on their knowledge of the language.

4 STRATEGIES OF IMPROVING PROGRAMMING LANGUAGES

Based on the above analysis, there are various computer programming languages with different functions and effects, bringing inconvenience for programmers. Therefore, developers and programmers should work together to improve computer programming languages.

4.1 Integrating programming languages with similar function

In many programming languages, some share same or similar function, though with different emphasis. Their strengths should be integrated to develop a new programming language with the advantages. For example, languages of PHP, CGI, ASP and JSP have their own advantages and disadvantages in writing server-side program. Thus, the advantages of these languages can be integrated to develop a new language. The waste of resources can be minimized, with reduction of the workload of programmers to learn many programming languages (Zhang Yiqing, Li Huagui, 2003).

4.2 Enlarging range of object-oriented languages

An important feature of programming language development is that the operation becomes simpler and more convenient. The feature is reflected from machine language to assembly language, and from advanced language to non-procedural language and intelligent language. Compared with procedural languages, object-oriented languages greatly reduce

the workload of programmers. Meanwhile, programming efficiency has been greatly improved, significantly increasing the productivity of IT industry. However, the application range of object-oriented languages is relatively small due to inefficient versatility and flexibility, thus greatly limiting their performance. In reality, the investment in object-oriented languages should be increased, with expanding their application scope, thus making them better serve programmers.

4.3 Improving computer programming language with multi-party cooperation

Currently, the development of various computer programming languages is not satisfying. One of the important reasons is the lack of multi-party cooperation. The advent of any new technologies cannot only rely on individual power. Different programming languages are usually developed by various companies. They do not cooperate with each other due to factors such as commercial purpose. In addition, companies developing programming language lack effective communication with programmers. This development mode without efficient cooperation inevitably leads to waste and repetition of work. Meanwhile, it is not conducive to overcoming weaknesses and developing more efficient and reasonable programming languages. In order to make up for the deficiencies, cooperation with profit share should be strengthened in developing programming languages; developers should communicate with programmers to collect wisdom and efforts, making programming language more suitable for the actual requirements.

5 CONCLUSIONS

There is no universal computer programming language after decades of development. Different languages have their own characteristics and advantages in application. Programmers should reasonably select programming language according to contents, characteristics and specific requirements of the work, as well as their knowledge on the languages. Meanwhile, problems should be timely recognized in the use of language, reducing malignant competition. Long-term mechanism of win-win cooperation should be established to improve the development of computer programming languages with multiple supports. Thus, the

languages can be more suitable for actual needs and maximize work efficiency of programmers.

REFERENCES

- Deng Desen. Selection of Computer Programming Language. *Science & Wealth*, 2011 (5): 272-273.
- Shao Yuzhou. Issues on Programming Language Teaching. *Journal of Beijing Institute of Economic Management*, 2001 (04): 58-60.
- Zhang Yiqing, Li Huagui. Optimization of C/C++ Code in Embedded Real-time Programming. *Microcomputer Information*, 2003 (01): 48-49.