ISSN 2616-5775 Vol. 6, Issue 1: 52-56, DOI: 10.25236/AJCIS.2023.060108

Research on how to optimize data structures with C++ language

Weilun Guan

Lexington Christian Academy, Boston, 02420, USA

Abstract: *As the most basic programming language in high-level computer languages, C++ language has the advantages of simple attributes, convenient use, no compilation environment restrictions, fewer syntax restrictions, and possible operation in different operating systems, thus making it easy to execute a migration. Since its advent, C language has been popular among programming enthusiasts. With continuous improvement in the long process of development, it has eventually formed a complete theoretical system, which plays a pivotal role in programming languages. This paper focuses on how to optimize data structures with C++ language.*

Keywords: *C++ language; D ata structures; application areas*

1. Introduction

As a branch derived from C language, C++ language inherits the functions and advantages of C language, enabling it to adapt to more complex graph structure data. It has found wide application in computer data structures, not only with a variety of functions and libraries but also with powerful features. It is necessary for programmers to master the knowledge in the field of data structure when learning software programming and development technology. Data structures include linear structures, tree structures, and graph structures, with arrays, linked lists, stacks, trees, and graphs as frequently used data structures. They not only are applied to organize data, but also greatly affect the running speed of code. Since data varies in structure, programs may differ by many orders of magnitude in running speed. A good command of data structure results in that of data processing algorithm[1]. A good data structure is very important for the execution efficiency and data storage efficiency of the software system.

2. Overview of C++ language

2.1 The development of C++ language

Early computer programming languages are computer control instructions, each of which is a set of binary numbers. Different calculations have different computer instruction sets. Program development using binary instruction sets is extremely complex and requires memorizing a large number of binary numbers. In order to facilitate memory, letter combinations are used as a substitute for binary numbers, and string keywords become assembly language instead of the programming language in binary machine code. As a low-level language, assembly language is easier to remember than machine code. However, it cannot avoid the disadvantage of poor readability, since plenty of jump instructions and address values make it difficult for programmers to understand the meaning of the program in a very short period of time. Consequently, programming language has entered the era of high-level language.[2]

The first high-level language was ShortCode developed by UNIVAC in 1952, while the most popular one is FORTRAN designed by Backus (an American scientist) on IBM's computers. However, FORTRAN and Algo160 were mainly applied for scientific and engineering calculations, followed by Pascal and C languages. C language is developed on the basis of other languages. First, Richard Matin developed a high-level language BCPL. Then, Ken Thompson simplified it into a new language, B language, which lacks the concept of type. Dennis Ritchie studied and improved B language by adding structure and type to it, and he named the improved language C language.[3]

C++ language is developed from C language. Strousstrup has added the concept of class to C language through intensive research. The original name of C++ was Cwith Class. In December 1983,

ISSN 2616-5775 Vol. 6, Issue 1: 52-56, DOI: 10.25236/AJCIS.2023.060108

Rick Mascitti suggested that it be renamed C-Plus-Plus, namely C++.

2.2 Features of C++ language

Data structures in C++ language are diverse, including fundamental data types (such as int, float, double, char, enum, etc.), structured data types (such as array, structure, union, etc.), and pointers. They not only provide users with user-defined data types to realize complex data structures, but also define classes to realize object-oriented programming. Efficient programs can be achieved through the combination of classes and pointers.

The control statements of C++ language have various forms and are easy to use. There are several control languages, such as two-way branch, multi-way branch and loop structure, which are convenient for the realization and control of structured modules. Combined with object-oriented programming, it is convenient for program compilation and maintenance.

As an object-oriented programming language, C++ language combines abstraction with reality. Objects communicate with each other by means of message and increase code reuse through inheritance.

Due to the strong portability of C++ language, programs written in C++ can be used on different types of computers without much modification. C++ standard applies to a variety of operating systems.

On the other hand, C++ also has some shortcomings. The C++ language is too complicated, which even makes it difficult for human beings to understand its semantics, which makes the C++ compiler system affected by the complexity of C++ and very difficult to write. Even the compiler that can be used has a lot of problems, most of which are difficult to find. Because of its complexity, the correctness of complex C++ programs is quite difficult to guarantee. Some people also put forward some defects such as primitives that do not support multithreading. However, so many famous people have put forward so many defects, which shows that C++ is widely used and successful.

2.3 Application Areas of C++ language

As a programming language, C++ can be both process-oriented and object-oriented. C++ is a widely used computer programming language, so it is favored by more and more programmers. Compared with java and other languages, it has many advantages, and many large companies' server programs are also developed based on C++. At present, c++ software development mainly focuses on the following areas. The application fields of C++ language are shown in Figure 1.

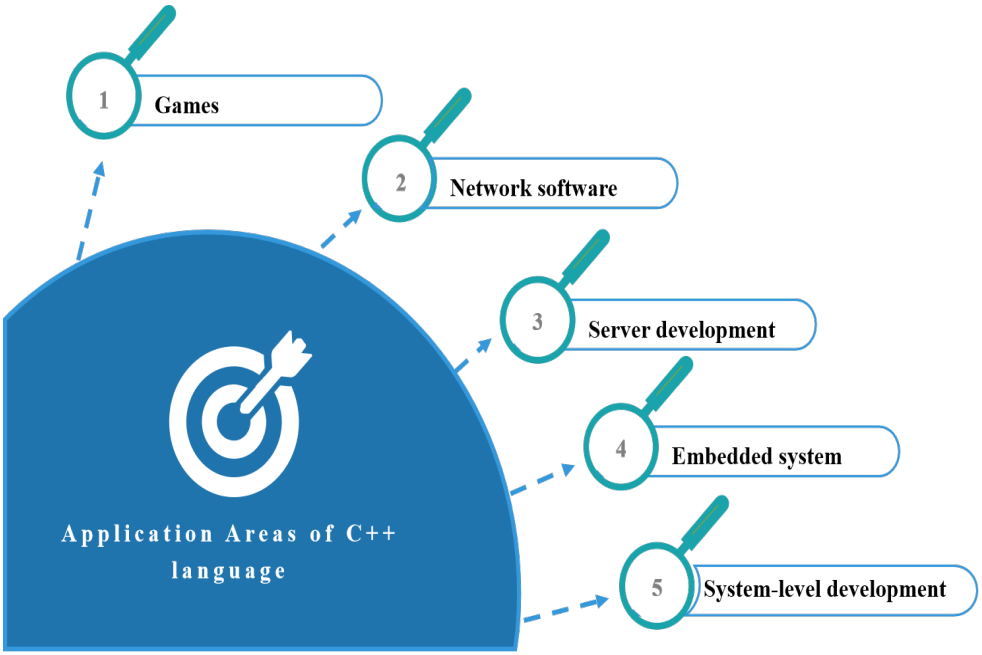

Figure1: Application Areas of C++ language

ISSN 2616-5775 Vol. 6, Issue 1: 52-56, DOI: 10.25236/AJCIS.2023.060108

2.3.1 Games

C++ has found wide application in the field of games in recent years, due to its super efficiency, advanced numerical calculation libraries, generic programming and other advantages. At present, C++ language provides a lot of extensibility, which can be used for small and large game projects. Moreover, it is platform independent, which means that you can simply transfer projects from one operating system to another. C++ is undoubtedly one of the best programming languages for game projects: Wizard 3, Dark Soul, Elder Scrolls 5: Sky, PUBG, Fortress Night, etc. are all game projects created in C++.[4]

2.3.2 Network software

C++ has many mature libraries for network communication, the most representative of which is the cross-platform, heavyweight ACE library, one of the most significant achievements of C++ language. It is applied in many important enterprises, departments and even the military.

2.3.3 Server development

The back-end servers of many Internet companies are developed based on C++, most of which are Linux, Unix and other similar operating systems. It is necessary to be familiar with the Linux operating system and its development thereon, database development, and network programming, all of which cannot do without the support of C++.

2.3.4 Embedded system

Because C++ has high efficiency and maintains compatibility with C language, it can make the underlying platform have high efficiency and great flexibility, so that it has great application in the underlying development. In addition, C++ also has a good performance in software expansion, migration and maintenance.

2.3.5 System-level development

Although C language is the main programming language in this field, C++ language can be used to write drivers due to its compatibility with C language and application in low-level development. Therefore, C++ can be used to develop system-level software and write operating systems.

3. The application of C++ language in optimizing data structures

3.1 Data structures

A data structure is a collection of data elements with structural characteristics. It studies the logical and physical structures of data, as well as their relationship. By defining appropriate operations for a structure, it designs corresponding algorithms. Meanwhile, it is guaranteed that the new structure obtained after these operations still maintains its original structure type. In short, a data structure is a collection of data elements that have one or more specific relationships with each other, or, a collection of data elements with a “structure”. “Structure” refers to the relationship between data elements, which is divided into logical structure and storage structure[5].

3.2 To optimize data structures with C++ language

3.2.1 Application of C++ language in linear structure

Linear structures have various manifestations, such as linear linked lists, stacks, doubly linked lists, queues, etc. All structures have common attributes, with one data element at the beginning and one at the end. Except the first and last elements, each element corresponds to a specific and unique predecessor and successor. These features make the structure of a linear table simpler and easier to understand than other data structures, so linear structures are the most widely used. In such structures, it is necessary for data elements to select reasonable predecessor and successor modes based on the storage mode. However, it does not mean that the storage modes in linear structures are all linear. For instance, data storage of a linked list is not linear. At the operational level, linear structures can be divided into insertion, deletion, and modification. If the storage mode in a linear structure is also linear, it will be possible to greatly simplify the operation and improve work efficiency.

In C++, data elements can be processed directly with the powerful functions of the language, but it is relatively difficult to operate and much similar to the way that linear structures are operated with C

ISSN 2616-5775 Vol. 6, Issue 1: 52-56, DOI: 10.25236/AJCIS.2023.060108

language. In addition to these methods, users also use various containers in C++, many of which are essentially different types of linear structures. The flexible use of these containers can greatly reduce the code involved in the operation. There is no need for users to care about whether the C++ implementation code uses linear storage or non-linear storage. This part is transparent to users. If the linear structure used is relatively simple and has high requirements for processing speed, C++ pointers can be used for processing. C++ language has a variety of pointers for fundamental data. If the data structure is designed by the user, C++ language can also support user-defined pointers. The types of many pointers are interchangeable, a function unavailable in many other languages.

3.2.2 Application of C + + language in tree structures

Trees and graphs are representative of nonlinear structures, and tree data structures are quite important. There are many tree structures in human life. In order to make the computer process some data better, people abstract the tree structure. A tree structure consists of many nodes, only one of which is root node. The other nodes do not intersect with each other, called subtrees of the root. Most of the trees used in actual programming are multi-layer trees. Except for the lowest nodes, the rest are subtrees. Such an abstract type is similar to a tree in nature, so it is called a tree structure. The difference between a linear structure and a tree structure is that the nodes in the tree structure only correspond to one predecessor and multiple successors at the same time. It is not sequential in overall structure but continuously spreads from top to bottom. There are many types of tree structures, with different characteristics for different types, such as balanced trees, cross trees, etc. Generally, different methods are used for the storage and construction of different trees. As far as C++ language is concerned, all the conditions for using a tree structure are provided. For example, the connection of a tree is usually multi-point, while the storage space of a computer is sequential, so the tree is usually discontinuous in the computer. That calls for the use of pointers, which are very powerful in C++. In addition to user's own implementation of the tree structure, C++ also provides relevant functions that have the features of tree construction, maintenance, and search. Meanwhile, these functions make C++ more efficient in processing tree structures, which not only reduces operational difficulties of tree structures but also brings great convenience.

3.2.3 Application of C++ language in graph structures

The most complex structure model in data structures is the graph structure, which has only one predecessor and successor for all linear structures. In contrast, each node in a tree structure has only one predecessor and multiple successors, and there is no intersection and connection between adjacent sub-numbers. Besides, the sense of hierarchy is relatively obvious. However, in a graph structure, any two different data elements have certain connections, as a result, the formed graph structure is relatively complicated. According to the distinctiveness of graph structure, it is difficult to establish functions that can be directly operated and processed by graphics in C++ language. Some are only basic graphics processing methods, so graphics processing can only rely on programmers to conduct the analysis and processing based on specific requirements and problems. The processing of graph data structures in C++ has the following advantages. First, the code efficiency is relatively high. Although the operation speed of C++ is slower than that of C language and assembly language, it is faster than that of other languages, which makes C++ suitable for dealing with complex problems. Second, pointers are quite flexible. C++ pointers are difficult for beginners to apply in practice. However, unexpected results may be obtained by means of C++ pointers, especially in designing various algorithms to solve complex graphics problems. Third, the logic is very strict. With strong logic, C++ can write various complex algorithms to handle graphics problems. In a word, C++ language is the most efficient and suitable way to process graph data structures except C language.

4. Conclusion

This paper summarizes the development history and features of C++ language, and analyzes how to apply C++ to optimize data structures, and examines the knowledge system of C++ language and several common problems in programming. The rapid development of computer science and applications has directly promoted the rapid progress of C++ technology and its improvement. The most effective way to learn C++ language is programming, and program debugging is a very important and essential part of application development, which can be modified by compiling hints for some syntax errors in the program. For some hidden errors that cannot be reported by the compiler, researchers can track and debug them according to the debugging tools of the C++ IDE (set breakpoints, single-step execution debugging function). In summary, with more practice and accumulation, various

ISSN 2616-5775 Vol. 6, Issue 1: 52-56, DOI: 10.25236/AJCIS.2023.060108

errors can be avoided in order to master the C++ language programming techniques.

References

[1] Fan Xiang. A framework for exploring the development of computer software programming based on C language technology [J]. Electronic Components and Information Technology, 2021, 5 (12): 182-183. doi: 10.19772/j.cnki.2096-4455.2021.12.080.
[2] Ritchie D M. The development of the C language [J]. ACM Sigplan Notices, 1993, 28(3): 201-208.
[3] Kernighan B W, Ritchie D M. The C programming language [M]. Pearson Educación, 1988.
[4] Information on: https://www.oschina.net/news/170627/gaming-projects-top10-programming-languages-2021
[5] Shi Yuqiang, Yan Dashun. Data Structure and Algorithm: China Agricultural University Press, 2017.02