Pisco Med Publishing

Design Optimization and Application Analysis of Embedded C Program

Yi Jiang

Zhejiang Gongshang University, Hangzhou, 310000, China.

Abstract: From MCU (Micro Control Unit) to ARM (Advanced RISC Machine), from smart phone to industrial Internet of Things, and to embedded development technology, applications have changed rapidly in the past 10 years, and embedded system has been widely used. Due to the characteristics of C programming language and its flexibility in language level, more and more programmers have chosen the embedded application development. Therefore, by analyzing the characteristics of the embedded system and C programming language, this paper discusses the application and optimization of embedded system as well as C programming language, aiming to help developers to improve the efficiency of program development and execution.

Keywords: C Language Assembly Language; Mixed Programming

1. Introduction

Embedded system is a special computer system which takes application as the center, takes modern computer technology as the foundation, and can flexibly cut software and hardware modules according to user needs (function, reliability, cost, volume, power consumption, environment, etc.).One of its characteristics is that the software and hardware can be cut, so it needs a language which can control the hardware and is easy to program and transplant. The characteristics of C language are very suitable for embedded development. The most widely used C language should belong to Unix and Linux operating system. At present, Android and IOS kernel, which are widely used in smartphone operating system, are modified based on Linux and UNIX. However, with the development of embedded system, new challenges also make C language development meet rivals. Other languages such as Java, Object-C and python are also widely accepted by embedded programmers. Therefore, when using C language as embedded development application, we need to optimize it as much as possible to adapt to the current rapid development needs[i].

2. Advantages and disadvantages of embedded C language

C language as a high-level language, has the characteristics of high-level language, can realize some upper application development, has high readability, easy for programmers to write code, also has the characteristics of low-level language, facing the bottom, can read and write memory and register through pointer, so it is easy to access hardware. A standard library is defined in C language, which makes the program written by programmer applicable to all the libraries in compiler to the greatest extent, and is easy to compile and transplant. For example, it can't access some registers. It is a process-oriented language, and its expansibility is worse than other object-oriented languages. Therefore, we need to use the characteristics of other languages combined with the design and application characteristics of C language in embedded system to optimize.

3. Embedded C language development and design：

3.1 Development Environment

The development of embedded software has more strict and complex requirements than traditional PC. For example, in the windows operating system on PC, we can download the compiler tools on its platform, and directly compile and develop windows system programs, that is, local compilation. The development and debugging of embedded software is usually carried out in the cross compiling environment. We should first edit and use the compiler to compile on the ordinary PC, and then copy the executable file generated by the compiler to the arm platform to run. This process is cross compiling. Embedded system is usually integrated on a specific hardware platform, located at the bottom of the software, used to coordinate the interaction between hardware and upper software. In addition, most of the embedded applications play a role in aerospace control, which also makes the embedded software development language have higher and more stringent requirements[ii].Compared with low-level assembly language, C language has become a better choice for embedded software development. It not only has the efficiency of low-level programming language, but also has the development efficiency of high-level programming language.

3.2 Compiling Process

Once we have finished the source code work for C, we must compile it. Generally speaking, in the compilation process, C code should be compiled into assembly code through the corresponding assembly compiler. In order to run on the target machine, we must also link it to other files as executable binaries to identify the target. Therefore, assembly plays a bridge role in the compilation process. If we can handle the call relationship between C language and assembly language, we can improve the efficiency of the code.

4. Embedded C program design and application optimization

4.1 Mixed programming of C language and assembly language

In the design of embedded applications, if all the programming tasks are completed by assembly language, although the object code execution efficiency is high, but its workload will be large and difficult to maintain; if all the tasks are completed by C language, although it is concise and clear, but the object code execution efficiency is low, especially in the application with strong real-time performance, it will highlight the shortcomings of C. Therefore, an embedded application program is usually implemented by the mixed programming of assembly language and C language. In addition to the assembly language programming for the initialization part, the main programming tasks are usually completed by C.

4.2 Determination of parameter passing and calling relationship

Assembly language program and C language program involve parameter transfer when they are called, usually using stack for parameter transfer. In assembly language, BP is used as base register. The calling program first pushes the parameters in the C language program into the stack in turn, and then uses BP plus different offsets to stack the stack in turn when these parameters need to be used. Access the data in the operation. For the procedure or function to be called, it is necessary to explain and establish the relationship between the callee and the callee in advance. The procedure or function to be called should be specified as an external type in advance, so that it can be referenced by an external module. The calling program also needs to specify the name of the external module to be referenced in the program. When interpreting the calling relation, the corresponding assembly language format should be determined according to different storage methods. The small C program pattern corresponds to the near process of the assembler, while the large c program pattern corresponds to the far process of the assembler[iii].

4.3 Design application expansion

From the above analysis, we can see that the compilation of C program needs to go through the process of compilation → link → executable. Therefore, it is inevitable to complete the compilation process again when debugging and controlling the application. C language is a process-oriented language. In the development of some upper level applications such as UI, it is not as easy to expand and maintain as object-oriented language. In order to extend the application of C program, Lua is adopted to meet the current requirements. Lua script language has been successfully applied to online games. By analyzing its characteristics, combining with the characteristics of C language and the requirements of embedded development, we can combine Lua and C into the development of embedded system. With the development of dynamic language, Lua can help programs to complete more functions and simplify the work of programmers. Therefore, the advantages of combining Lua with C are discussed here. Firstly, Lua is a script language, which is a programming language created to solve the traditional development language needs to be encoded, compiled, linked and executed. Unlike the advanced language, Lua does not need to compile itself, but performs "interpretation execution" when executing, because it can save compilation process and reduce development time. Lua script itself is very small, compared with other scripting languages such as python, it runs fast and occupies less memory, which is very suitable for embedded devices with limited memory resources. Secondly, Lua is open source . Lua itself is implemented in C language. The code is very concise and short, so it is easy to transplant to C programs, and it can be compiled on almost all platforms. Therefore, there is no need to worry about the problems of difficult transplantation and heavy hardware consumption. The characteristics of embedded system require good portability of software, so Lua is also accepted by many embedded developers. Finally, Lua has good expansibility, the defined variables have no type restrictions, and adopts the object-oriented rather than process-oriented design structure, which makes up for some shortcomings of C language in design. Through these three advantages, the combination of lua and C is a good choice for embedded application development. Lua's parser can be transplanted to C program, and a set of interfaces can be provided for Lua in the program Call, programmers only need to write a simple Lua script to call these interfaces, through UART serial port or USB transfer Lua file to the embedded device for analysis, to complete a series of control without modifying the C program code. For example, in the factory's mechanical arm operation, we need to change the operation action. We can use the combination of lua and C, and write Lua script to realize a new group of operation action, without having to go through the process of program modification compile run again, which greatly improves the development efficiency1.

5. Conclusion

In general, the code set of the compiler will become the assembly set of the compiler, and the compiler of the program will be compiled. It shall full play to their respective advantages, cross call each other, parameter transfer, and share data information and data structure. Since then, the software developed has become more practical, safer, and the development and programming work has been doubled. Therefore, we need to fully analyze and master the programming and application of C language to improve the applicability of embedded C program. In the process of its active development and utilization, it is necessary to study the actual electronic hardware system, and reasonably combine the embedded C language with assembly language (such as Lua language) in the development and design, and expand some functions, which can realize the functions that cannot be completed by high-level language components, and improve the efficiency of application development and execution.

References

[1] Zepeng Cai.Exploration and Analysis of Embedded C Program Design Optimization and Application [J]. Modern information technology.2018-01-25:35.

[2] Zhu Yuan.Pointer detection research and System implementation of embedded C program [D]. Dalian Institute of Technology.2012-06-13.

[3] Wendong Wang, Li Zhulin, still Jian. Hybrid programming techniques for assembly language and C languages [J]. Computer technology and development. 2006-08-10:19-20.