Apply Digital Simulations to Source Code Understanding in the Embedded System

Cheng cunxue, Yan daibiao
PLA 92117 unit, Beijing, China
yandaibiao@sina.com

Abstract - Embedded system software has a close contact with its hardware, so the paper makes digital simulations of the running environment of analyzed software, thereout we can find exception address and exception path when system go in various states, on the base of which, we can divide the software into comparatively unattached function model, finally we can obtain function structure chart, so that we can study the source code more easily.

Index Terms - Embedded system, Distributed simulation, Software understanding, Function structure chart

1. Introduction

Software reverse engineering and maintenance is often a miscellaneous task, especially it is difficult for us to understand the existent software systems which have only source codes. Despite many people have put up much program understanding technology, such as method of engineering, method of program slicing, method of concept evaluate, method of concept analysis, mode-matching technique, intelligent upstanding and so on[1], but first we must divide the program into many relatively function models on one's own, then we can analyze the program by program understanding technology when we faces a large-scale system which has a complex software architecture level. Basing on the source code, we can get the program flow chart automatically with some arithmetic and syntax of the programming language [2][3][4], thus we can understand the structure of program, but we can't get the function model of the program, also we can't understand meanings of the program segment. Embedded system software has a close contact with its hardware, the function of which can be carried out by constructing the simulation platform of the hardware (such as apparatus and operation panel), simulating the relative in-out data and enabling the system running. This paper makes digital simulation of the running environment of analyzed software, thus we can find exception address and exception path when system go in various states, on the base of which, we can divide the software into comparatively unattached function model, finally we can obtain function structure chart, so that we can study the source code more easily.

2. Idea of Digital Simulations

Embedded system is often a big complex system, which has a complex running environment for its software. It is very difficult to making digital simulation to the embedded system by only one computer, and it is inconvenient to develop and manage the software system. Thus we must distribute the simulation tasks into every net node that depends on many computers, namely distributed simulation.

Because embedded system has many devices, if each device uses a computer, while each node task is relatively homogenous and is simple to implement, but needs more number of computers, which takes spending much. In order to compensate for the lack of hardware platform, we can use the multi-tasking operating systems and use a computer to simulate multiple physical devices, in which simulation module of each device can be regarded as a process, at the same time, the execution of the process can be controlled by the interposes synchronization mechanisms provided by the operating system. Based on different systems, we divide the simulation process and control running of the simulation system.

3. Structure Design of Simulation System

Digital distributed simulation system often get the digitalization of various devices by computer systems, then translate the actual system emulation model into computer models by establishing mathematical model for system simulation, so that can take an evaluation and optimization of systems [5]. Because attaining software function hierarchy diagram is the purpose of digital simulation in the process of understanding software source code, it should be distinguished from the general digital distributed simulation systems in their system design. Design of hardware system should be focus on the construction of simulation platform. Design of software systems should be focus on the conversion of software source code in the embedded system and the software simulation of the corresponding hardware capabilities, such as Control knell, display device, and so on.

Embedded digital simulation systems include hardware and software systems. Hardware system is mainly consists of many computers and one workstation (how many computers depending on the complexity of the system). They form a logical bus on Ethernet, the technology of making nets is relatively simple, its structure is shown in Figure 1. Ethernet frequencies are between 10M-100M depend on different system, and adaptive 10M-100M. In Figure 1, except the working station, each computer makes a software simulation for corresponding hardware features, such as control panels, switches and thumb wheel, and so on, thus the simulation system can complete the switches of different working methods, types and functions by responding the operator's instructions at any time.

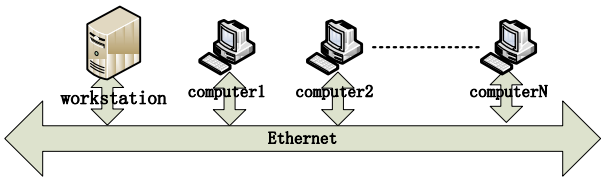

Fig.1 hardware system structure

Software platform can select the operating system which supports the TCP/IP protocol, such as UNIX and Windows NT, which is easy to connecting different types of machines in a network. Software systems focus on developing the simulation support software, including HMI software, source code conversion software, controlling software, network communications software, information convert software, and so on. The system structure is exhibited in Figure 2.

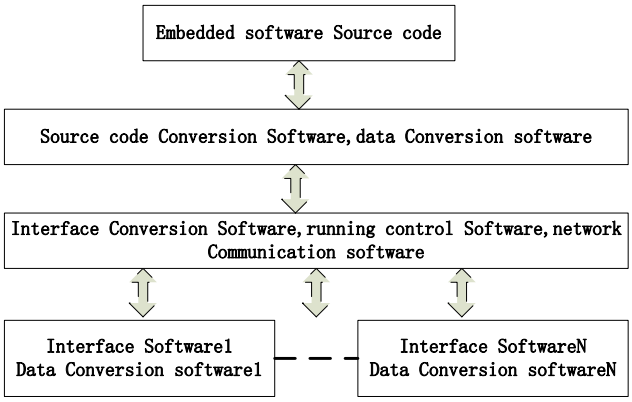

Fig.2 simulation software structures

4. Design of Simulation Software System

Because making use of the source code of embedded software system, software system is relatively simple, only it can run the source code of embedded software systems by some conversion.

A. Task Distribution of Software System

We consider two software mostly: the first is the information conversion software which contacts with external devices, namely instruction translation software of source code; the second is the human-machine interface software, network communications software, and control software. Workstation is the core of the simulation system, so the instruction translation software of source code, control software and network communication software are deployed on the station. Human-machine interface software and information convert software are designed according to the devices and features of each computer, and exchange information according to certain rules of network traffic.

B. Functions of Software System

Source code conversion software: the language of embedded system software is a machine-oriented, according to the transformation of instruction set, syntax and semantics, each statement of software source code can carry out the according function in the existing simulation environment. Firstly, we must understand the specific features of various directives in embedded system, and the data haves been used, and the data sources, for example constants, registers, storage unit or an external device; second, we must count for the numbers of registers and memory statistics used, and replace it with some kind of data structure; finally, we will achieve its functionality in an emulated environment. For example, for a article addition directive, one operation number comes from constants, another comes from memory, added results are put into left-players, so in the conversion process of directive, first we defines data structures according to features and common of data used, then we defines the variables corresponding to memories and left-players, then read a specifies of constants as a operation number from constants paragraph, then read another operation number from memory variable, finally, we put the added results into registers variable.

Information converts software implements mutual recognition between the information between virtual device information and the information needed in implementation of software source code. On the simulation platform, the information is extracted from the corresponding virtual registers, virtual memory or virtual input/output status of the device, which related with hardware of embedded systems and needed in implementation of software source code. The control virtual machine status information is also get from execution results of the software source code. By information convert software, we can achieve the mutual recognition of the two different types of information. For example, a status of a knob on the control panel of embedded systems will be convert as the information which needed in the implementation process of source code by the information convert software; on the other hand, execution results of software source code would also be convert as the information which can be identified by virtual device information, so as to control the status of virtual devices.

Human machine interface and control software are used to control running of the source code by source code conversion software modules, such as running step, setting breakpoints, running to cursor, displaying the corresponding source code, running recording, and so on. Control software should also have compiled function, which is able to compile the software modified and correct errors in syntax and variables. Each virtual device of simulation platform will produce different interrupt message according to different operations, but control software will handle the interrupt message, and begin to run the corresponding source code segment, then the control software will return.

While other computers will make a "virtual" interface design according to corresponding device emulated, such as the console operator panel, the instrument pointer, then make a program for corresponding conversion software.

4.3 Communication among Software Model

As can be seen from the above, the workstation software can be divided into two process in which HMI software and control processes running in the foreground, also be called the

primary process, while source code conversion software and network communication software running in the background. Other microcomputer carries out its own individual process which can be controlled to start or stop by the primary process of workstation.

When the platform need the status information of a some kind of special equipment (a computer) when it run the software source code statements, system will make a query for the information which is corresponding to the virtual device's status information. Because the virtual device's status is in real time, each state makes a match to a message, when the status is update, the corresponding information is updated. When need to output some kind of executive outcomes when running a source code statements, platform will send the information to the device completed the function, namely a computer, and also send a message to starting the computer, then the computer will response to this message and complete these functions by convert software. In order to ensure reliable data transfer, system will select connection-oriented protocols ~TCP protocol, and at the same time adopt client/server mode between the two processes being communicating.

5. Testing of Simulation System

How to verify that source code program is running correctly when a simulation system has been set up? It is essential important for correctness of translation software of source code, if there is an error in conversion of the source code, the result may be bizarre and unpredictable. Because the majority embedded systems will make a test by using suitable test instruments and test methods in the phase of developing test software system, while each test case has standard input and output data, and then check up the correctness of a software system by contrasting the running result and the standard result, so that we can also use a standard test case to test the correctness of simulation system, if it is not correct, then make debugging by running control software, such as running step and other ways, to check for errors that may appear, until we obtain properly data being the tolerance scope.

By using of the simulation system tested, we can carry out the work as follows:

1) Acquire execution: Run the software source code without taking any action to "virtual device", the execution path of software is running process of the master target software program. Execution path can be expressed in source code text, can also be described by flow chart. Usually we use the latter so that understanding software execution paths more intuitively.

2) Acquire function hierarchy diagram: According to the operating manual, we can implement the function by taking an action on the "virtual device", which is based on the querying specific actions of some features. Tracking and recording the execution address and location of source code in different working condition, so that we can understand software implementation process when it is in different State and performs different functions. Then compares them with the execution path of the master program execution path, locates the connection point with the master program, expands at the master program execution path, and finally we can acquire function hierarchy diagram of the target software.

Using the simulation system, we can acquire the features chart of embedded systems easily. There is a section of source code that corresponds to each feature, and then we make a analysis on a relatively independent source code by the software understanding technology, which not only improves the understanding efficiency of software, but also acquires more specific and in-depth understanding to the software.

6. Conclusion

Using the above-mentioned ideas and methods, we established a simulation platform about computer software source code of a firepower, command and control. By using the simulation platform, we achieved the desired goal good results. In establishing simulation system, we must understand the system's hardware, the contact of software and hardware, and the information of the various devices in different States in hardware, therefore, there are a lot of work to do before establishing simulation system, but it is an inefficient, error-prone and tedious labor if the programmer try to understand the functionality of the program only by repeatedly reading the source code. Using established simulation platform, the programmer analyzes execution paths to various situations and get the target software function hierarchy diagram, which will greatly improve the efficiency of software understanding. Additionally, once simulation system of operating environment of target software has been established, we can make a test by using of the simulation system of target software, and may find subtle errors in the software. In the software maintenance process, if we modified the target software and add new functions, we can make a test to the target software changed by using the simulation system.

References

[1] Li bixin etc. The research and development of software comprehension. Computer Software and Applications. 1999, (8): 897-905.

[2] Chen lijun etc. Software execution path visualization, Chinese Journal of computers.1998, (3): 193-203.

[3] Shan yongming. A method of automatically generating source to the control flow graph. Micro computer system. 1996, (10):45-49.

[4] Ding zhongjun. From the source to the conversion chart method. Micro computer system. 1994, (5):34-39.

[5] Pang junjun etc. Research and Realization of distributed system simulation technology. Fire control and command. 2001,(3):37-40