UEFI Firmware Storage System Analysis

Weiming Wu[1,a], Hui Wu[1,a], Jiongjiang Lan[1], Wenxin Lai[1], Qing Su[1]

[1]Guangdong University of Technology, Guangzhou 510000, China;

[a]xiaohui_hubei@163.com

Keywords: UEFI, BIOS, Firmware Storage System.

Abstract. The analysis of UEFI firmware storage system is rare in available papers about UEFI.We introduce the Logical and physical structure of UEFI firmware. Then we analyse the elements -- firmware volume, firmware file, firmware segment -- that composed the UEFI firmware and give related code define and pay additional attention to the typical firmware segment. In the experiment, we analyse a typical UEFI BIOS firmware with binary viewer and introduce the nested firmware segment.We also create a program prototype witch can resolve and extract modules included in UEFI firmware to verify the analysis correctness.

1. Introduction

BIOS(Basic Input and Output System) is the foundation of computer work. It is responsible for initializing hardware, hardware function detection and guide the operating system.Many design of the first generation BIOS can not meet the needs of today, such as 16 the addressing mode is low efficiency and can't support more than 2 TB disk volume, extensibility is poor and security is imperfect.UEFI which is the next generation of BIOS technology has the following advantages:the operating system pre boot environment, startup and runtime service, Independent drive with CPU,secure startup and update, etc.UEFI is a open specification, so any manufacturer can be free to modify and extend UEFI under UEFI sepcification[1].

2. UEFI Firmware Overview

UEFI boot process is divided into two stages: the initialization of the firmware and the loading of the operating system.Respectively corresponds to the PI UEFI (Platform Initialization UEFI) specification and UEFI specification.PI UEFI specification defines the initialization process of the UEFI platform, as well as the implementation of various types of protocols for the UEFI driver to provide the implementation environment.In this paper, the definition of the UEFI firmware storage system is defined by the PI UEFI specification.

UEFI firmware is a storage device that is used to store the UEFI code. It is generally the ROM chip on the motherboard, but it can also be stored on the hard disk, flash memory and other storage devices.The code stored in the UEFI firmware is the code that is executed first after the computer is powered on.These code implement the simlilar function as traditional BIOS, such as the power of self inspection, initialization of external devices, etc.In addition, it has a lot of its own characteristics,Such as safe start, fast start, support for large capacity external storage, easy to extend the interface, etc[2,3].UEFI firmware can be regarded as a micro operating system, to provide the boot service and run time service for real operating system, and manage all kinds of hardware, but also easy to expand to support the new hardware and various computer architecture in future.

Research on the UEFI firmware storage system is helpful to understand the structure of the UEFI firmware module.No need to know the case of the firmware code, you can directly through the tool to increase, delete and modify the firmware module, thereby affecting the computer behavior, which is important in computer security and other aspects.

3. UEFI Firmware Storage System

The logical structure of the UEFI firmware file system is a generalized table.Each node in the

table can be a generalized table node or an atomic node, and its type includes a firmware volume , a firmware file , or a firmware segment .Figure 1 is a schematic diagram of the logical structure of the UEFI firmware file system[4,5].

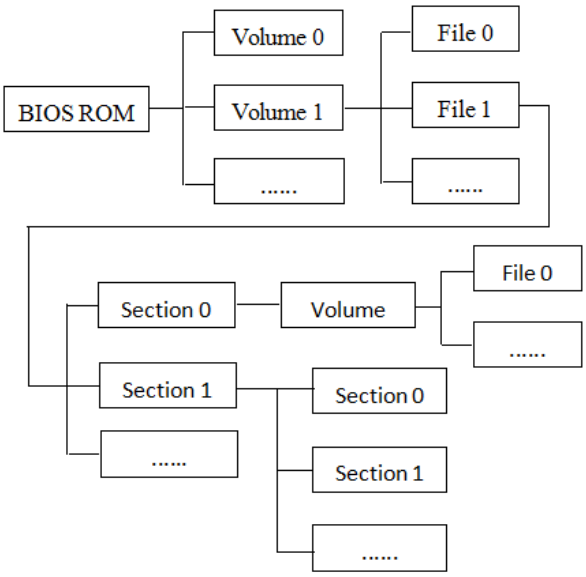

Fig.1 Logical structure of UEFI framwire filesystem

It can be seen from the figure that the UEFI firmware (ROM BIOS) is composed of a set of firmware volumes, each of which is composed of a set of firmware files, each of which is composed of the most basic unit firmware segment.There are many types of firmware segment, according to the different types of firmware to make different interpretations of the data.For example, the firmware segment type is a compression type, which indicates that the data section needs to be extracted; and the firmware segment type is the type of firmware volume, which indicates that the data section is a new firmware volume.

UEFI firmware file system storage structure using sequential storage, the advantage is to save storage space, the disadvantage is to increase the search time, but due to the small size of the UEFI firmware, the impact on the search efficiency is not obvious.

3.1 Firmware Devices

The firmware device is a physical storage device that is used to permanently store firmware code and data.Generally this device is a Flash component, but it may also be other types of non-volatile storage devices.EEPROM is a common firmware device that stores compiled BIOS binaries on the motherboard.Due to the limitation of the memory capacity, a complete set of BIOS codes may be divided into several parts, which are stored in different firmware devices.In the course of the experiment, we found that some of the BIOS code for computing devices are scattered on the main board of the two EEPROM.The BIOS code must be extracted from the two EEPROM content, combined into a complete set of BIOS code, can be correctly identified and used.

3.2 Firmware Volume

The firmware volume is a kind of logic on the firmware, it is the storage of data and the code of the basic storage warehouse, a firmware device is generally composed of a number of firmware volume.Each firmware volume has its own file system, and the basic component is the firmware file[6~8].Figure 2 is the firmware volume structure.

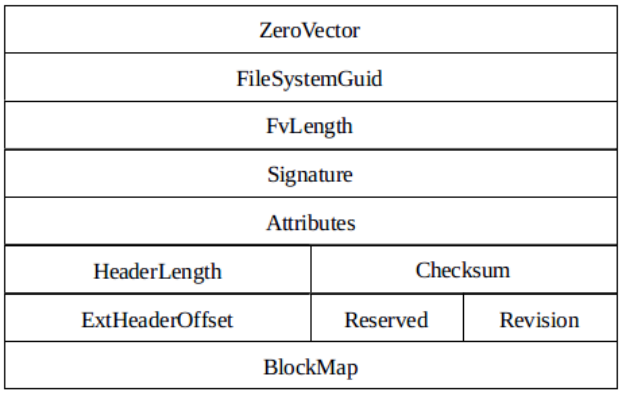

Fig.2 Firmware volume structrue

3.2.1 Firmware File System

Firmware file system mainly describe the organizational form of firmware volume file and free space, each kind of firmware file system are composed of a guid (globally unique identifier) said file system driver through the guid can identify the latest firmware volume.

The firmware file system in the PI UEFI architecture describes the binary format of the file stored in the firmware file.All files in the firmware file system are stored directly in the root directory, and are stored in a sequential order.If you want to get the file that exists in the current firmware volume, you must traverse the firmware volume from scratch until the end of the volume[9].

3.2.2 Firmware File

The firmware file is used to store code or data in the firmware volume. Each firmware file has four basic properties[10,11]:

1) Name: each firmware file has a name that is composed of GUID GUID, the UEFI value must be unique in the firmware volume of the firmware file.

2) Type: each firmware file has its own type, represented by a 8 bit integer value.

3) Alignment: each firmware file must be aligned to 8 bytes.

4) Size: the size of the data in the firmware file, in bytes.

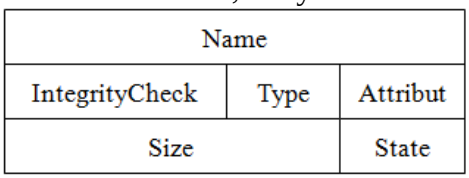

Fig.3 Firmware file structrue

3.2.3 Firmware Section

The firmware is a component of the firmware file, each of which describes a part of the firmware file data.The firmware segment is divided into the following two categories according to whether the data can be directly included[12]:

Leaf Section: only contains data.

container Section: it does not contain data directly, it is only used as a container containing other containers or leaf segments.A container segment may have a plurality of sub segments, each of which may be a leaf segment or a vessel segment.

The relationship between container segment and leaf segment can be described by tree structure, which can be regarded as the non leaf node in the tree, and the leaf is the leaf node in the tree.As shown in Figure 4.

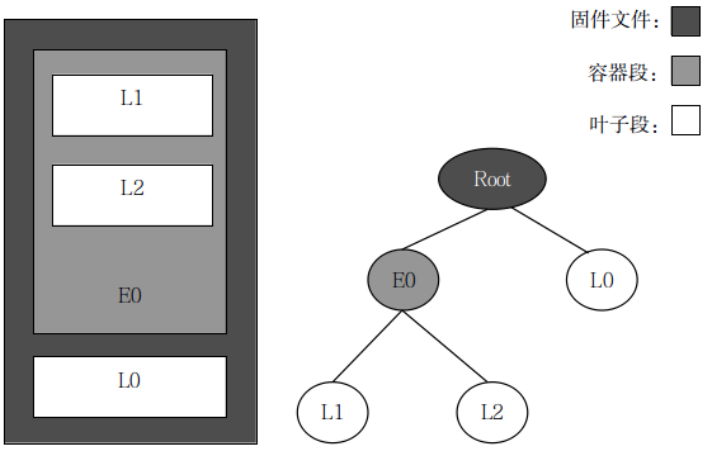

Fig.4 A example of firmware segment

4. Experimental Results

In order to verify the above file system format, using C++ to prepare an analytical program for parsing BIOS UEFI firmware file system structure.Show as Figure 5.

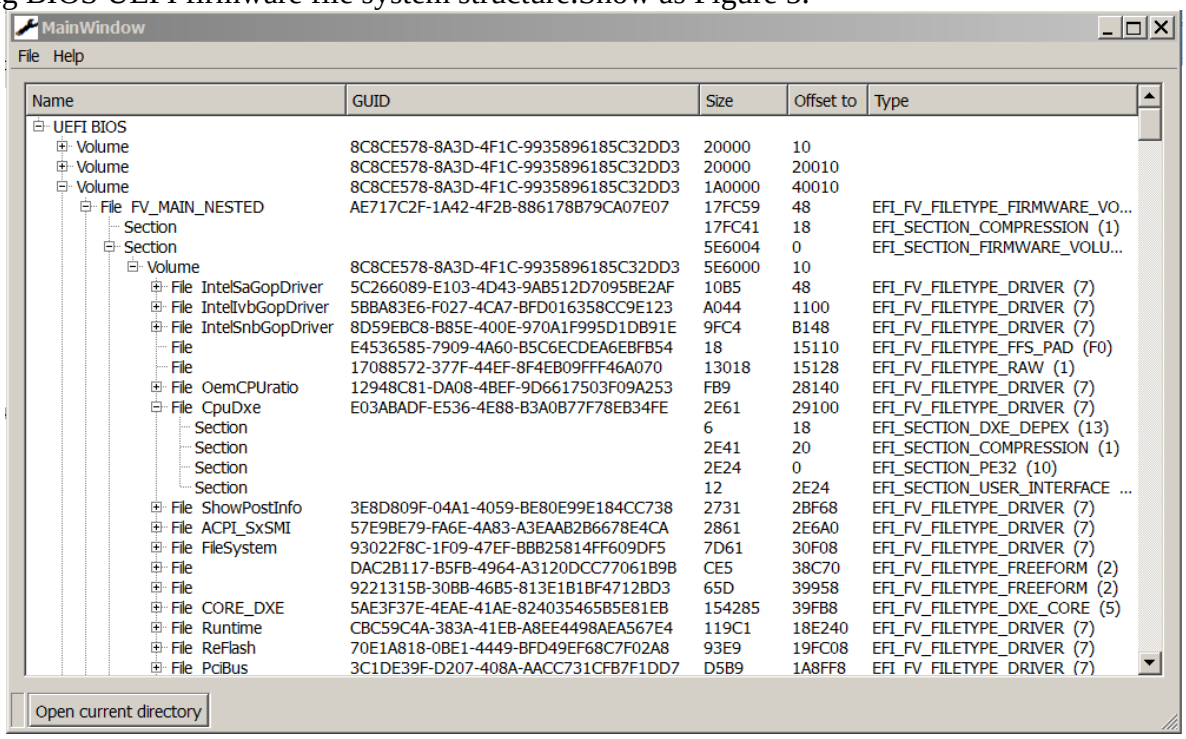

Fig.5 Analyse firmware file system by program

The analytical procedure algorithm is described as follows:

(1) The UEFI firmware is composed of a set of firmware volumes, with a capital letter B that represents a single UEFI firmware.The firmware volume can be expressed as: B = V1, {V0, V2,...} (the index starts from zero, the same below). Parsing starts when traversing all the firmware volumes in the UEFI firmware Vi (I = 0, 1, 2,...), the parse of each firmware volume is transferred to (2).

(2) Each firmware volume is composed of a set of firmware files, which are represented by a V with a capital letter.The contains the firmware file can be expressed as: v = {F0, F1, F2,...}. Traversing the firmware volume of all the firmware file fi (I = 0, 1, 2,...), analysis of each firmware file go to (3).

(3) Each firmware file is composed of a set of firmware segments, and a single firmware file is represented by a capital letter F.The firmware section can be expressed as: F = {S0, S1, S2,....}.Traverse all the firmware segments in the current firmware file Si (i=0, 1, 2,...), analyzing the data contained in each firmware segment based on its type.

5. Conclusion

In this paper, the logical structure and storage structure of UEFI firmware are analyzed, and then the firmware, firmware file, firmware segment structure are analyzed and the code definition is given.At the same time, the paper introduces the principle of compression decompression algorithm used in UEFI firmware.Finally, a prototype of the program is implemented, which can be used to parse and extract the specified firmware module, and the result is good.

Reference

[1]. Vincent Zimmer, Michael Rothman, Suresh Mar-isetty. Beyond BIOS:Developing with the Unified Extensible Firmware Interface.Intel Press, 2010, p.350-365.

[2]. Jian Pen, Gang Xie. File system driver for extensible firmware interface. Computer Engineering. Vol 34(2008) No. 9, p.83-85.

[3]. Mei Chen, Chenliu Zhou, Rongsheng Xu.Analysis of Firmware Volume Structure based on Framework.Computer engineering and Applications. Vol (2007) No. 15, p.86-88.

[4]. Deng Pan, Guangming Liu. Analysis of the EFI A rchitecture and the EFI Driver DevelopmenComputer Engineering and Science. Vol 28(2006) No. 2, p.115-117.

[5]. Jie Zhou,Zhiyong Xie,Han Yu. Research on Domestic Computer BIOS Based on UEFI. Computer Engineering. Vol 37(2011) No.1, p.355-358.

[6]. Zhijian Gao, Chunlei Jiang. LZ77 Compresses Algorithms and Derives Algorithms to Probe into.Journal of Xichang College Natural Science Edition. Vol 19(2005) No. 1, p.88-91.

[7]. Zhensheng Yu, Rong Li, Yi Lu. Application of LZMA Compressed algorithms in the firmware updating process.Electronic Measurement Technology. Vol 29(2006) No. 5, p.135-137.

[8]. Zhaoqing He. New Data Compression Algorithm Based on Huffman Coding. Science Technology and Engineering. Vol 8(2008) No. 16, p.31-35.

[9]. Haibin Huang, Jing Jin. Solution of multi file system based on EFI system. Information Technology. Vol 8(2010) No. 6, p.122-126.

[10].Jian Yan. Application of huffman algorithm in data compression. Computers and modernization. Vol 8(1996) No.4, p.15-19.

[11]. Jia Liu, Xiaochen Xin, Ganggang Shen. Research and development of UEFI Flash update. Computer Engineering and Design. Vol 32(2011) No. 1, p.114-117.

[12]. Xiaozhen Wang, Lei Yu, Baoxu Liu. Data Backup and Recovery Technology Based on EDK2 Platform.Computer Engineering. Vol 37(2011) No.15, p.262-264.