第37卷 第15期 计 算 机 工 程 2011年8月
Vol.37 No.15 Computer Engineering August 2011

·开发研究与设计技术· 文章编号：1000—3428(2011)15—0262—03 文献标识码：A 中图分类号：TP39

基于EDK2平台的数据备份与恢复技术

王晓箴[1,2]，于 磊[1]，刘宝旭[1]

(1. 中国科学院高能物理研究所计算中心，北京 100049；2. 中国科学院研究生院，北京 100049)

摘 要：针对文件分配表文件系统数据的安全保障需求，基于符合统一可扩展固件接口(UEFI)规范的EFI开发套件2(EDK2)开源环境，提出一种固件层数据备份与恢复技术。利用EDK2提供协议接口读取硬盘分区数据，将获取的文件及目录信息备份为EFI镜像文件格式，数据恢复时反向写入目标分区。实验结果表明，该技术可解决传统数据备份对操作系统的过度依赖问题，其在固件层实现数据备份，实用性更强。

关键词：可扩展固件接口；数据备份；数据恢复；EFI开发套件2；文件分配表系统

Data Backup and Recovery Technology Based on EDK2 Platform

WANG Xiao-zhen[1,2], YU Lei[1], LIU Bao-xu[1]

(1. Computing Center, Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China;
2. Graduate School of Chinese Academy of Sciences, Beijing 100049, China)

【Abstract】Aiming at the target of protecting File Allocation Table(FAT) file system data security, based on EFI Developer Kit 2(EDK2) open source environment defined by Unified Extensible Firmware Interface(UEFI) specification, this paper proposes a technology of data backup and recovery on firmware level. After using the protocols EDK2 provided for accessing disk partition data. It gets details of files and directories by calculating, and saves them as EFI image file, which will write into destination partition when the recovery module works. Comparing to legacy data backup technology, this technology works on firmware level, is independent of operating system, and this advantage makes it more usable.

【Key words】Extensible Firmware Interface(EFI); data backup; data recovery; EFI Developer Kit 2(EDK2); file distribution list system

DOI: 10.3969/j.issn.1000-3428.2011.15.085

1 概述

计算机存储的信息量正以指数级飞速增长，数据逐渐成为计算机应用的核心，数据安全也被提升到了前所未有的高度。为保障计算机系统在各种环境下的正常运行，保障硬盘数据安全，应采取先进有效的措施对数据进行备份与恢复，防患于未然。

可扩展固件接口(Extensible Firmware Interface, EFI)是在Intel主导下提出的下一代基本输入输出系统(Basic Input Output System, BIOS)技术，旨在定义操作系统与平台固件之间一套完整的接口规范，用以替代传统的BIOS。Intel于1999年发布了首个EFI技术规范，2005年业界成立了统一可扩展固件接口(Unified EFI, UEFI)论坛，共同制订适应于各种平台的接口标准，并于2006年1月发布UEFI2.0版本。随着各大厂商纷纷推出支持UEFI的主板，未来UEFI的应用必将成为主流。UEFI提供高效的嵌入式应用平台，可根据实际需要开发出各种应用。

本文在UEFI EFI开发套件2(EFI Developer Kit 2, EDK2)开发环境的基础上，实现基于固件层的硬盘数据备份与恢复应用，解决传统过程中对操作系统的依赖。

2 UEFI及EDK2介绍

2.1 UEFI启动

UEFI规范定义的启动过程在Boot Manager控制中，从平台最基础的固件初始化开始，接着开始装载EFI镜像文件(EFI Images)，包括EFI驱动(EFI Drivers)和应用程序(EFI Applications)，进一步初始化以丰富平台功能，最后装载的是OS Loader，开始启动OS。在进入OS Loader后，EFI的启动服务将终止[1]。

2.2 EDK2分析

EDK是实现UEFI平台所需基础代码的开源集合，EDK为开发、调试及测试EFI驱动和应用文件等提供了条件。

EDK的编译必须在完整源代码的根目录下进行，为开发带来很多不便。结合用户反馈，文献[2]于2010年推出EDK的升级版EDK2，使开发者可以更方便地编写应用。

利用EDK2进行UEFI应用程序开发时的概念如下：

(1)模块(Module)：是最小规模的独立可编译代码或预编译二进制文件。每个模块中需包含元数据文件INF文件，用来描述模块的行为及属性。

(2)包(Package)：是发布工程的最小单位，是工程的逻辑划分，一个包中可以包含零个或多个模块。包中必须包含包元数据文件DEC文件，平台元数据文件DSC可选。

(3)全局唯一标识符(Global Unique Identifier, GUID)：用以标识实体的128位值，具备唯一性。

(4)平台配置数据库(Platform Configuration Database, PCD)：使开发者可以从模块外部获取信息，并可以控制模块内部流程。

(5)协议(Protocol)：UEFI中的API函数，用GUID进行命名。

基金项目：国家科技支撑计划基金资助重点项目(2009BAH52B06)；中国科学院知识创新工程重要方向基金资助项目(YYYJ-1013)

作者简介：王晓箴(1985－)，女，博士研究生，主研方向：网络安全，信息安全；于 磊，工程师；刘宝旭，副研究员

收稿日期：2011-02-22 **E-mail：**wangxz@ihep.ac.cn

3　系统设计与实现

3.1　UEFI 模块编写

3.1.1　模块开发

本系统基于 EDK2 进行开发，开发过程包括以下 4 个阶段：

(1)创建包

创建 DEC 文件定义包的接口，包括其余包中引用的目录、GUID 值、Protocol GUID 值、对本包内发布的 PCD 入口点的声明等。

(2)创建模块

编写模块中的 INF 文件，用以定义模块类型、需要的函数库、需要的 Protocol/GUID/PCD、与其他模块的依赖关系等。INF 文件创建完成后，编写源代码，实现数据备份与恢复功能。

(3)创建编译所需的 DSC 文件

DSC 文件用以描述如何对包进行编译，包括需要被编译的模块、不同模块类型所选择的库实例、模块中使用的 PCD 入口点的配置。

(4)调试模块

完成相关文件编写及源代码编写后，对模块进行调试。

3.1.2　编译基本架构

EDK2 的编译基于 Python 语言及 C 语言，可以支持跨平台的编译操作。EDK2 编译系统基本架构的工作流如图 1 所示。

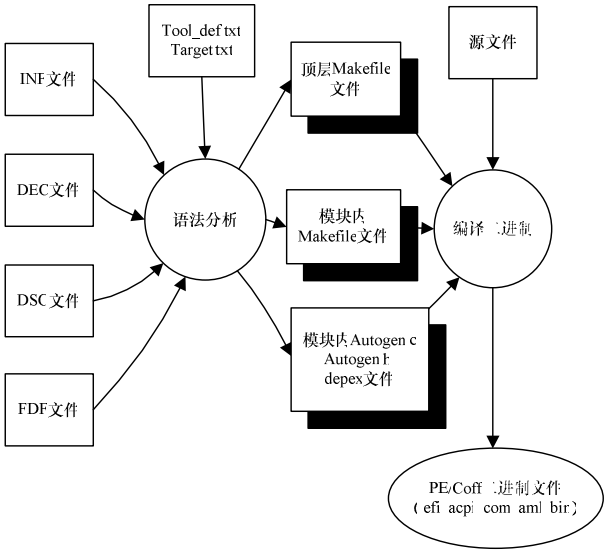


图 1　EDK2 编译过程工作流

在图 1 中，圆形表示过程；矩形表示源或输入；阴影框表示输出或中间量；椭圆表示最终 ROM 镜像。EDK2 编译工具对元数据文件(DSC、DEC、INF 文件)进行语法分析，生成相应的 Makefile 文件及 autogen.c/autogen.h 文件。在 autogen 文件中，EDK2 编译工具生成本模块使用到的 GUID、Protocol、PPI 及 PCD 的定义，并且自动激发本模块入口函数实现中所应用到的所有库实例的构造函数。

3.2　硬盘分区数据获取

UEFI EDK2 提供对文件分配表(File Allocation Table, FAT)文件系统的库支持，本文即研究实现 FAT 分区的数据备份与恢复。

3.2.1　FAT 文件系统

FAT 文件系统是当前 Windows 系统中主流的文件系统之一，包括 FAT12、FAT16 和 FAT32 3 种版本，不仅用于硬盘，而且广泛应用于 U 盘等移动存储设备上。一个 FAT 分区由分区引导扇区(DBR)、保留扇区、FAT、文件目录表(File Directory Table, FDT)和数据区构成[3]。

分区引导扇区是系统格式化分区时产生的，位于分区的第 1 个扇区，包含一个引导程序和一个被称为 BPB(BIOS Parameter Block)的本分区参数记录表，BPB 主要记录 FAT 各个部分的起始扇区、结束扇区、根目录大小、FAT 个数、分配单元大小等重要参数信息。

FAT 表是 DOS/Windows 系统的文件寻址系统。为了数据安全起见，有 2 个完全一样的 FAT。一个是基本 FAT，另一个是 FAT 的备份，两者在磁盘上前后紧排在一起，其大小由本分区的大小及文件分配单元的大小不同而变化。FDT 表紧接在第 2 个 FAT 表后，记录着每个文件(目录)的起始单元、文件的属性等。

3.2.2　硬盘分区数据读取

在 UEFI 构架中固件层读取硬盘设备信息，其思路为利用 EDK2 中提供的对 Windows Nt32 平台块设备读取的库支持，使用 Win32 API 对计算机中的块设备进行读取。同时，在创建的元数据 DSC 文件中定义 PCD 参数，表示块设备属性如 PhysicalDisk、VirtualDisk 等。

在 INF 文件中，定义本模块入口函数为 InitializaWinNtBlockIo。入口函数调用 WinNtBlockIoDriverBindingStart 函数，绑定块设备读取驱动，成功后调用 CreateMapping 函数创建镜像，进行块设备数据读取。OpenDevice 函数用以顺序打开块设备并返回句柄，ReadBlock/WriteBlock 函数均调用 ReadWriteCommon 方法对块设备中的数据进行读写操作。块设备读取模块整体流程图如图 2 所示。

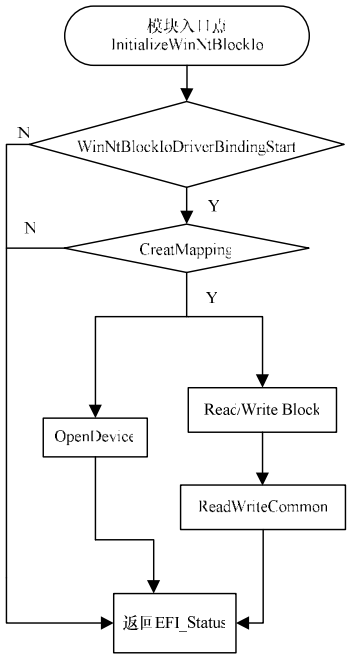


图 2　块设备读取模块流程

通过块设备读取模块，获取所有支持块输入输出协议的设备后，从中筛选出硬盘设备，读取目标分区的 DBR 区、FAT 表、FDT 表数据。通过对 FAT 表、FDT 表数据结构进行的分析，可获取该分区目录与文件位置、大小等详细信息。

3.3　数据备份及恢复

在得到硬盘分区中文件详细信息后，便可开始在 uefi 固

件层进行数据备份，备份文件通过 EFI_FILE_PROTOCOL 接口，保存为 UEFI 特有的 efi 镜像文件，可以使备份文件更稳定，不受操作系统格式的影响。

读写 efi 镜像文件的具体步骤为：通过 uefi 启动服务 LocateHandleBuffer()和 HandleProtocol()，得到所有支持 EFI_SIMPLE_FILE_SYSTEM_PROTOCOL 的句柄，再通过 OpenVolume()打开当前卷设备的根目录，利用 EFI_FILE_PROTOCOL 进行 EFI 镜像文件的创建、读写操作。

数据备份与恢复模块流程[4]如图 3 所示。

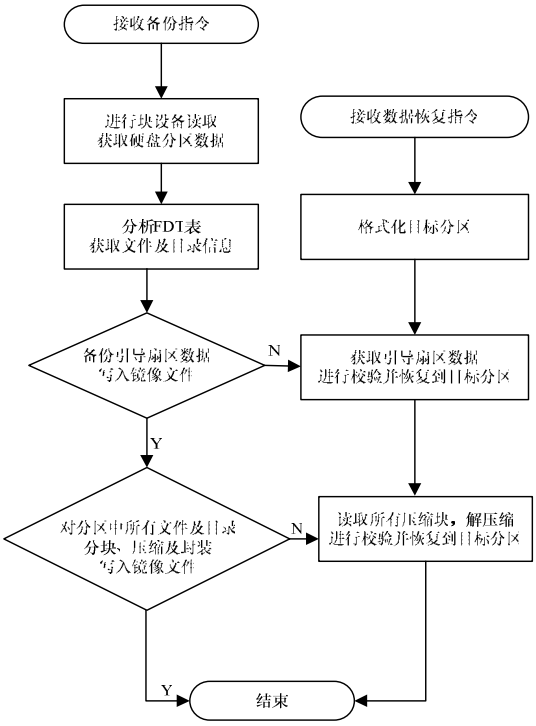


图 3 数据备份与恢复模块流程

4 实验分析

4.1 开发环境

本系统开发环境为：操作系统 Windows XP、编译工具 Microsoft Visual Studio .Net 2005、编码工具 Source Insight 3.5、运行环境 UEFI Shell。UEFI Shell 是简单易用的命令行交互环境，在操作系统(Windows 和 Linux 系统均提供支持)启动前进入[5]。用户可以在 Shell 命令行中进行装载 UEFI 设备驱动、运行 EFI 应用程序、启动操作系统等操作。本文考虑实验成本，选择以模拟环境方式在操作系统中运行 UEFI Shell，进行开发测试。

4.2 编译调试

EDK2 源代码中 Conf 目录中包含 build_rule.txt、target.txt、tools_def.txt 3 个文件，对其进行改写，定制编译环境，指定所需编译工具等。在整体环境配置完毕后，再对模块参数进行配置：将模块的 INF 文件定义添加至所在包的 DSC 文件中，并对模块中用到的库函数及 PCD 进行定义。

在配置完成后，在 Windows 命令行下，对编写的数据备份及恢复模块进行编译。EDK2 编译命令的格式为：

Build –p Package.dsc –m Module.inf –a ARCH –b TARGET –t TOOLTAG

4.3 运行测试

在测试用机的硬盘中创建一个目标分区，文件系统为 FAT16，分区大小 1 GB，存储 10 个文件大小共计 3.2 MB，盘符标识 J。

在模块编译通过后，在 Windows 命令行下输入 build run，便可进入 UEFI Shell 模拟运行环境。其运行界面如图 4 所示。

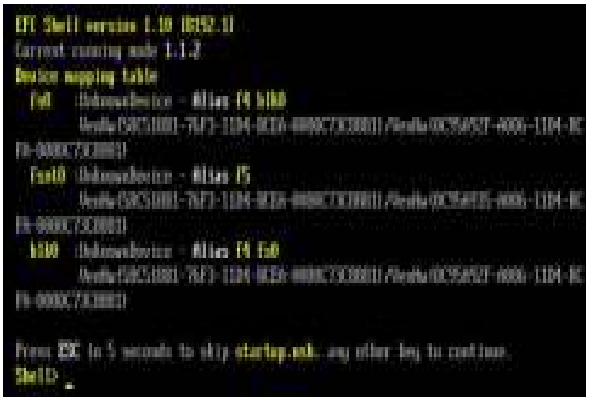


图 4 UEFI Shell 命令行截图

在 Shell 中输入命令，进入数据备份模块所在目录，运行 efi 应用程序，生成镜像文件 backup.efi 即为分区 J 的备份文件。在格式化分区 J 后，再次进入 UEFI Shell 运行环境。进入数据恢复模块所在目录，运行 efi 应用程序。程序运行耗时 38 s，运行效率在可接受范围内。在程序运行结束后，检查分区 J，已成功恢复所有 10 个文件，分区运行情况正常。

5 结束语

在固件层实现对硬盘数据的备份及恢复，相比目前主流的 Ghost 等数据恢复软件，更加灵活，可集成在 UEFI 固件中，运行于操作系统前，无需受制于操作系统(包括 Windows、DOS、Linux 等)，无需预安装软件；备份文件保存为 EFI 镜像文件格式，可在文件级别上实现对硬盘分区数据的备份，更具安全性、更易存储、实用价值极高。今后将在 UEFI 固件层实现应用范围更广泛的 NTFS 文件系统的数据备份及恢复。

参考文献

[1] Unified EFI. Unified Extensible Firmware Interface Specification v2.3[EB/OL]. (2009-05-22). http://www.uefi.org/specs/.

[2] Intel Corporation. EDK II Module Writer’s Guide v0.7[EB/OL]. (2010-03-31). http://cdnetworks-kr-1.dl.sourceforge.net/project/edk2/General%20Documentation/EDKII%20Module%20Writer_s%20Guide_0_7.pdf.

[3] 戴士剑, 涂彦晖. 数据恢复技术[M]. 2 版. 北京: 电子工业出版社, 2007.

[4] 任 欣, 李 涛, 胡晓勤. 远程文件备份与恢复系统的设计与实现[J]. 计算机工程, 2009, 35(10): 112-114.

[5] Intel Corporation. EFI Shell Developer’s Guide v0.91[EB/OL]. (2005-06-27). http://www.mirrorservice.org/sites/download.sourceforge.net/pub/sourceforge/e/project/ef/efi-shell/documents/EFI_Shell_Developer_Guide_Ver0-91.pdf.

编辑 陆燕菲