

基于 VLIW 目标机的 ELF 可执行文件编辑器设计与实现

鲍丽丹^{1,2}, 张铁军¹, 王东辉¹

(1 中国科学院 声学研究所, 北京 100190; 2 中国科学院大学, 北京 100039)

摘 要: 程序测试是嵌入式软件开发的重要保证,但由于缺失源文件代码,需要对可执行文件深入解析以方便测试的进行. 将基于 VLIW 目标机实现对 ELF 可执行文件的解析与编辑,提供理解程序功能的途径,成为顺利执行程序测试的一种有效手段. ELF 文件编辑器将借助指令编码表,二级查找完成指令流的分析,同时利用 LEX 和 YACC 技术建立灵活的命令行模式,准确快速地完成二进制文件编辑,节约软件调试成本.

关键词: ELF 文件; 二进制编辑器; VLIW 结构

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2015)03-0015-04

Design and Implementation of Binary Editor Based on VLIW Machine

BAO Li-dan^{1,2}, ZHANG Tie-jun¹, WANG Dong-hui¹

(1 Digital System Intergration Lab, Institute of Acoustics Chinese Academy of Sciences, Beijing 100190, China;

2 University of Chinese Academy of Sciences, Beijing 100039, China)

Abstract: Procedure testing takes an important part in the software development of embedded system. In order to implement software testing in absent of source program code, executable file is analyzed and translated. This paper completes the design of binary editor, based on SuperV DSP with VLIW architecture. This binary editor performs analysis of program code using instruction template table and build intelligent command-line mode. It provides an effective way for understanding intention of applications, which makes large convenience in procedure testing.

Key words: ELF(Executable and Linkable Format); binary editor; VLIW architecture

1 引言

软件设计与系统开发过程庞大且复杂,测试工作是软件有效运行的重要保证. 开发过程往往由不同的功能单元组成,导致错误出现的环节并不确定. 但是测试过程中往往缺失源文件代码,使得测试者需要对二进制文件进行解析,更好地理解程序功能,最终定位并修改程序错误.

目前 PC 平台上流行的可执行文件格式主要包括 PE 格式和 ELF 格式,他们分别属于 Windows 和 Linux 操作系统. 此外还有 Intel/Microsoft 的 OMF 格式,Unix 系统下的 a.out 格式,以及 MS-DOS 下的 COM 格式等可执行文件格式.

Unix 中最早的可执行文件格式为 a.out,它的设

计简单,无法满足共享库概念的表达. 此后 Unix System Release 3 率先提出并使用了 COFF 格式,该格式解决了此问题. ELF 格式是源于 COFF 可执行文件格式的一个变种,它是 System V Release 4 在 COFF 基础之上改变引入的^[1]. Unix 系统实验室将其作为应用程序二进制接口的一部分发布. ELF 标准定义了一组能够延伸到不同的操作系统中二进制接口,有效减少重新编码和编译程序的开销^[2]. 常见的 Linux 系统也以 ELF 作为其基本可执行文件格式.

ELF(Executable and Linkable Format)文件是一种功能强大、结构较为复杂的可执行文件格式. ELF 文件标准中主要包括可重定位文件、可执行文件、共享库文件、核心存储文件四种格式类型. 可重定位文

件适合链接,可与其他目标文件链接后创建可执行文件或共享目标文件.可执行文件为程序提供进程映像和加载的内存执行.共享库文件包含了上下文中链接的代码和数据,其结构与可执行文件相似.

本文在 ELF 可执行文件结构的基础上,通过分析指令编码实现指令定位与修改.该二进制编辑器的设计实现概括如下:(1)建立指令编码表,并采用二级查找结构,快速解码指令;(2)实现指令格式的中间存储结构,完成对指令流的分析;(3)利用 LEX 和 YACC 技术分析指令关键字,实现命令行读入模式,自动匹配命令行指令.

2 ELF 编辑器原理分析

二进制可执行文件编辑器的实现,首先需要深入了解 ELF 文件格式,完成二进制数据的读取与分析.同时基于目标机器指令集描述进一步解析数据.本节将详细介绍 ELF 文件格式以及目标处理器的体系结构特征,最后分析 ELF 编辑器设计的关键点.

2.1 ELF 文件格式

ELF 可执行文件中包含编译后的机器指令代码、数据以及符号表、字符串等信息,并由不同的结构表示^[3].基于其文件格式,ELF 主要以 ELF 文件头,代码、数据等各个节(section),描述各个节信息的节头表(section header file),字符串表等等构成.图 1 中展示的是 ELF 文件的总体布局.



图 1 ELF 文件总体布局

ELF 文件中首先出现的是 ELF 文件头,它标识文件类型并描述文件的基本属性.文件头定义了 ELF 魔数、入口地址、程序头入口和长度、节头表偏移量与长度、节数等基础内容,并描述体系结构和操作系统版本等平台信息^[4].通过它我们可以计算节区头表位置,以及节区表项数目和长度,进一步获取各节区偏移和相关数据.

文件中代码和数据按照不同的属性,以“节”形式存储.而节的相关信息在节头表中描述,标识出每

个节的节名、长度、偏移量、读写权限等.节头表决定了当前文件的所有节属性和结构.程序员可通过计算节头表给出的偏移量来定位并访问相应的节属性.ELF 文件中另一个常见的结构是字符串表,它也是以节的形式存储着各个变量名,甚至是节表中用到的字符串.

ELF 文件系统已定义了一套规范的变量体系,文件各个部分的数据结构都详细定义,具体可参见 ELF 标准说明文档^[2,5].

2.2 目标体系结构特点

通过上节 ELF 文件格式分析,编辑器获取到文件的代码或数据等节区,并解析二进制代码段,将其映射为目标机器汇编代码.而反汇编过程中目标机器的指令集等体系结构特征至关重要.

本文目标机器为 SuperV DSP 处理器,它是一款面向嵌入式应用的处理器内核,以 VLIW 为基本架构,并结合单指令多数据流技术(SIMD),具有运算能力强、功耗低等特点^[6].在处理器结构上,它包含 VLIW 结构内核、系统协处理器以及 SIMD 协处理器,采用四发射超长指令字结构,结合 SIMD 技术,支持 8 位、16 位和 32 位数据.目标体系结构指令编码为 32 位,其指令码由操作码、寄存器、立即数、偏移量以及串并行信息等组成,如图 2 所示.

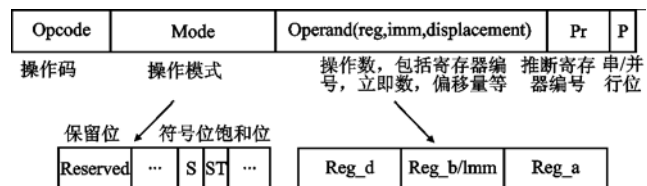


图 2 VLIW 目标机指令结构

操作码即指令序列号,标识操作类型和操作性质.操作模式包含各种不同的信息,依据操作类型不同而改变,以区分同类指令的不同操作.操作数规定了指令数字运算的量,指出数据来源.目标处理器支持推断操作,并由 Pr 字段定义了对应的推断寄存器编号.P 代表了并行发射中串/并行信息.

2.3 ELF 编辑器设计关键问题

基于目标机器的 ELF 编辑器,目标在于帮助快速理解二进制程序涵义,有效编辑可执行文件.为了正确获取二进制文件中的代码与数据信息,ELF 编辑器分析文件的代码节和数据节等.同时依据目标机器指令集等机器特征,完成二进制代码段到汇编代码的映射.对于不能直接读取源代码的程序,提供理解程序功能的途径,从而便于软件测试.同时针对

需要编辑的可执行文件,编辑器可直接采用指令形式修改文件代码,无需再次编译链接过程为可执行文件,节约软件调试成本。

编辑器设计的关键问题总结如下:首先 ELF 文件结构复杂,为了获取文件各个段节数据,需要对 ELF 文件结构进行合理解析读取。其次指令由二进制指令码转化为目标机器汇编代码,需要指令模板的支持。同时为了能够建立正确有效地命令行模式,编辑器需要自动识别命令行中的汇编指令和关键字,转化为指令数据结构。

3 ELF 编辑器设计实现与应用

本文将结合 ELF 文件结构,实现 SuperV DSP 处理器体系结构下的二进制可执行文件编辑器。编辑器的实现主要包括以下工作:根据目标机器指令集对指令模板进行定义,并针对 VLIW 结构特点设计指令包;解析 ELF 文件结构特点,准确读取数据代码段;建立参数机制,自动并有效地读取编辑命令行及替换指令。图 3 中解析了编辑器的整体构架,它的三个主要功能模块分别提供对应的支持。图 4 中列出了编辑器的完整工作流程。

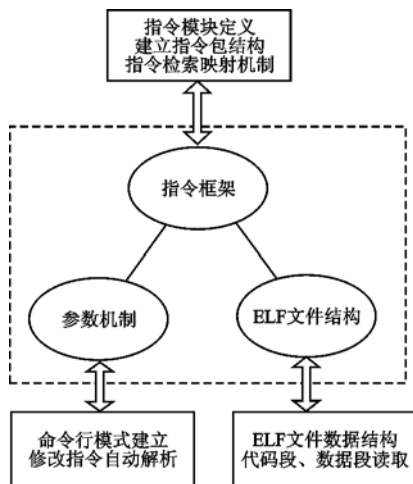


图3 编辑器功能模块图

3.1 目标机器指令模板定义

指令模板详细描述了目标机器指令集,为指令解析提供指令格式等信息。在反汇编过程中指令模板定义文件依据目标体系结构,描述所有指令的数据成员,包括操作码、源或目的操作数、标识位等等。

本文中指令模板的定义将以指令类型分类,相同指令操作码视为同类型,使用操作码识别。同指令类型的不同操作由指令模式的不同而区分开来,编

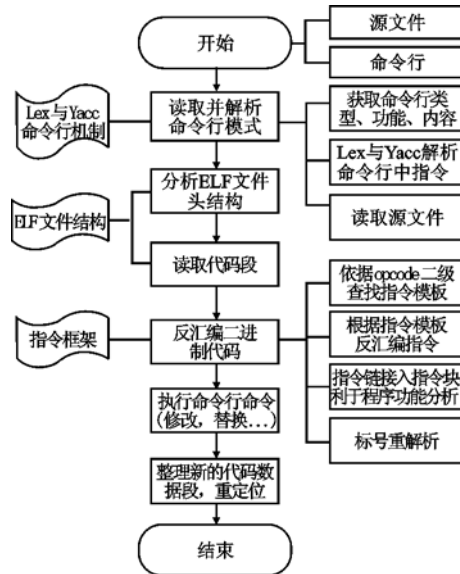


图4 编辑器工作流程

辑器整理指令模式以及指令操作码,作为指令掩码,便于定位指令模板,以及后续获取指令码中操作数、并行位信息。

指令码转换为可读性较好的汇编指令,需要寻找到正确的指令模板,本文采用二级查找机制,快速有效定位指令模板。整个反汇编指令模板映射过程总结如下:

(1) 由 ELF 文件中代码段得到指令码,提取相应指令操作码,确定指令类型,得到同类型指令模板范围;

(2) 基于第一步得到的查找区间,比较指令掩码,获取正确地指令模板,完成第二级查找;

(3) 依据指令模板映射指令,建立目标机器汇编指令数据结构(表示指令操作名、源、目的操作数类型及数据,并行位,推断寄存器等信息)

指令模板以及汇编指令的数据结构如下所示。

```
typedef struct INST_CODE_FORM {
    struct INST_CODE_FORM * next;
    UInt8 opcode;
    short id;
    char * name;
    UInt32special_opd;
    UInt8form_type;
    unionCode_format code_format;
} Inst_Code_Form;
classOper_asm {
    Inst_Code * inst_code;
    union Code_format opd;
    char * opcode_name;
    I_Operand ** pred;
    I_Operand ** dest;
    I_Operand ** src;
    UInt8serial_bit;
}
```

```
char *symbol_name;
}
```

3.2 参数机制设计与 ELF 文件结构解析

命令行参数模式给予编辑器更灵活的功能设置,帮助实现 ELF 可执行文件的反汇编功能,二进制代码的编辑修改,数据替换等一系列操作.本文依据编辑器需要的参数及功能,分别设置表 1 所示的选项,完成命令的有效读取.

表 1 命令行模式

- input	输入文件参数
- output	输出文件参数
- s	打印符号表参数
- h	打印段表头
- f	打印文件汇编代码
- find	mode 分为 up 和 down,同时显示前向或后向指令
	num 找到前后文的指令数
	addr 寻找的起始指令点地址
- edit	mode 编辑文件的模式
	addr 要编辑的地址位置
	Inst 要增加或替换的指令

针对汇编指令的特性,编辑器借助 lex 与 Yacc 技术完成命令行中的指令解析.lex 与 Yacc 分别是生成词汇分析器与剖析器的工具.通过 lex 发现命令行输入流中特定的标记,并把这些指令标记输入到特定的符号表中.符号表存储着数据类型、变量等各种信息.随后 Yacc 作为语法分析器使用特定的语法规则解释由 Lex 识别得到的标记,最终生成指令的语法结构.ELF 编辑器得到指令语法并映射为目标机器指令结构,提供后续可执行文件代码段的编辑.lex 与 yacc 如图 5 所示.

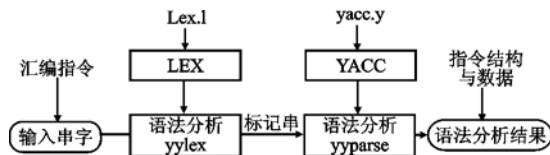


图 5 lex 与 yacc

编辑器另一个重要模块——ELF 文件结构的解析,包括对文件头的分析,代码段和数据段的读取.参照 ELF 文件结构标准,建立相关数据结构,准确识别各段节偏移,获取数据.具体文件结构如 2.1 节中所示.

3.3 ELF 编辑器应用

编辑器基于目标体系结构,完成对可执行程序

的编辑,实现相应的软件测试工作.图 6 所示为编辑器对程序分析实例.通过编辑器对可执行文件的解析,软件调试者正确理解程序功能,寻找程序或者相关软件错误点,便于错误的发现与调试.

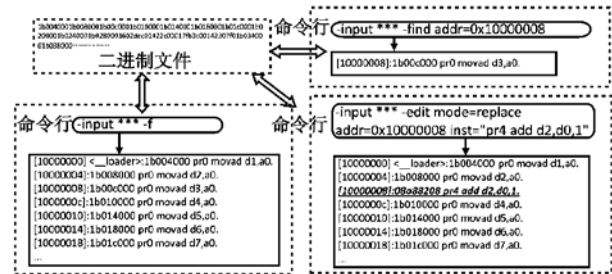


图 6 命令行模式下二进制编辑器运行实例

4 结束语

基于 VLIW 目标机器的 ELF 编辑器,提供灵活的命令行机制帮助快速理解二进制程序涵义、查找定位指令并且正确修改指定语句,从而有效提升程序调试的效率.本文的设计实现基于指令模板,通过分析具体的目标机器指令集,提取指令编码特性,采用反汇编与汇编算法,完成对目标机器可执行机器代码的解析与修改.该文件修改器自动提取相应指令编码格式文件,从而应对不同处理器的编码格式.灵活的命令行指示下,编辑器实现反汇编生成汇编文件,修改 elf 文件,查找特定地址,查找特定指令等功能.实验结果表明,二进制文件修改器能够正确地解析相关指令文件,准确定位指令代码,成功修改可执行文件.

参考文献:

- [1] Mahlke S A, Lin D C, Chen W Y, et al. Effective compiler support for predicated execution using the hyperblock[C] // PROC. of the 25th International Symposium on Microarchitecture. Latvia, IEEE, 1992:45-54.
- [2] Bernstein D, Cohen D, Lavon Y, et al. Performance evaluation of instruction scheduling on the IBM RISC system/6000[C] // Proceedings of the 25th Annual International Symposium on Microarchitecture. Barcelona, IEEE, 1992: 226-235.
- [3] 俞甲子, 矢凡, 潘爱民. 程序员的自我修养—链接、装载与库[M]. 北京: 电子工业出版社, 2009.

(下转第 22 页)