

RTEMS 引导程序的设计与实现

李红卫

(江苏技术师范学院 计算机科学与工程学院, 江苏 常州 213001)

摘 要:在嵌入式系统开发中,引导程序一般由引导记录 BOOT 和装入程序 LOADER 两部分组成,它主要负责初始化硬件设备和引导内核。基于 PC 机的 RTEMS 嵌入式操作系统通常由 GNU GRUB 引导。在分析 GNU GRUB 与 RTEMS 之间的接口和 ELF 可运行程序的文件格式的基础上给出了 BOOT 和 LOADER 的实现算法,并详细介绍了实模式到保护模式的切换过程,且给出了加载 ELF 可运行程序的实现算法。经上机调试,RTEMS 引导程序运行良好。

关键词:Bootloader; ELF 文件格式; RTEMS; BOOT; LOADER

中图分类号:TP316

文献标识码:A

文章编号:1673-629X(2008)07-0153-03

Design and Implementation of RTEMS Bootloader

LI Hong-wei

(School of Computer Science and Engineering, Jiangsu Teachers University of Technology, Changzhou 213001, China)

Abstract: In the embedded system development, the Bootloader generally is composed of the boot record BOOT and the loader program LOADER, it mostly presides over initialization of the hardware device and boot of operation system kernel. Usually, RTEMS based on the PC is bootloaded by GNU GRUB. Proposes the implementation algorithms of BOOT and LOADER after analysis of the interface between GNU GRUB and RTEMS and the format of executable ELF file, and introduces in detail the switch of the real mode to the protected mode, also gives the implementation algorithm of loading executable ELF file.

Key words: Bootloader; ELF file format; RTEMS; BOOT; LOADER

0 引言

RTEMS (Real Time Executive for Multiprocessor Systems) 是一个支持多处理机的嵌入式实时操作系统^[1],最初是由美国军方为导弹控制领域开发的嵌入式强实时操作系统。现在,它成为开源软件,由 OAR 公司提供后继的技术支持,其部分技术指标与商用实时操作系统 VxWorks 不相上下。RTEMS 采用了微内核的设计思想,动态分配内存,采用事件驱动,基于优先级的抢占式任务调度算法,支持多处理器系统以及进程之间的通信和同步等。

Bootloader 引导程序的设计是嵌入式系统软件开发的一个重要环节,它把操作系统和硬件平台衔接起来,通过初始化硬件设备、建立内存空间映射,为最终加载操作系统内核建立正确的环境。在 PC 机环境下,RTEMS 内核的引导通常采用 GNU GRUB 实现,

但在开发嵌入式软件系统时,人们更希望自己的内核能直接启动,而不需要功能强大的 GNU GRUB。为此,笔者基于 PC 机设计了一个引导装入程序 Bootloader,由它引导 RTEMS 的运行。

1 GNU GRUB 与 RTEMS 之间的接口

GNU GRUB 是一个符合多重引导规范的功能强大的操作系统加载程序,它是开源代码的自由软件。对于内核来说,不论它采用什么样的文件格式,若要用 GNU GRUB 引导,需要在内核中设置一个多重引导头,它的位置要求以 4 字节对齐,且必须在内核的前 8kB 以内。多重引导头定义了内核的一些特性参数,比如内核正文段的开始物理地址、数据段结束的物理地址、堆段结束的物理地址等。GNU GRUB 通过该多重引导头识别内核并引导。

GNU GRUB 在加载内核之前,已经完成以下工作:中断被禁止;打开 A20 地址线;CPU 进入保护模式;CS 指向基地址为 0x0,限长为 4G-1 的代码段描述符;DS、SS、ES、FS 和 GS 均指向基地址 0x0,限长为 4G-1 的数据段描述符;内核一般加载到 1MB 以上的

收稿日期:2007-10-18

基金项目:江苏省高校自然科学基金项目(06KJD520052);江苏技术师范学院应用基础研究基金项目(KYY06107)

作者简介:李红卫(1966-),男,山西阳城人,副教授,硕士,CCF 会员,研究方向为嵌入式操作系统。

内存空间。

RTEMS 内核采用 ELF (Executable and Linkable Format) 文件格式, 它被 GNU GRUB 直接识别。当 RTEMS 内核被装入内存后, 在 GNU GRUB 提供的的环境下开始运行。因此, 在设计 Bootloader^[2] 引导程序时, 只要它提供给 RTEMS 的运行环境与 GUN GRUB 提供的环境一致, 就能正确地引导 RTEMS 内核。

2 Bootloader 的设计

从操作系统的角度看, Bootloader 实现的功能是将内核正确地装入到内存中, 并将 CPU 控制权交给内核。由于 Bootloader 的实现依赖于 CPU 的体系结构, 因此大多数初始化引导程序分为两部分, 第一部分主要包含依赖于 CPU 体系结构的硬件初始化代码, 称为 BOOT 引导记录^[3], 其大小为 512 字节, 正好放在一个扇区中; 第二部分实现的功能比第一部分更多更复杂, 它为内核准备运行的环境, 将内核装入到指定位置, 并转到内核处运行, 称为 LOADER 装入程序, 它的大小不受限制。Bootloader 的工作流程如图 1 所示。

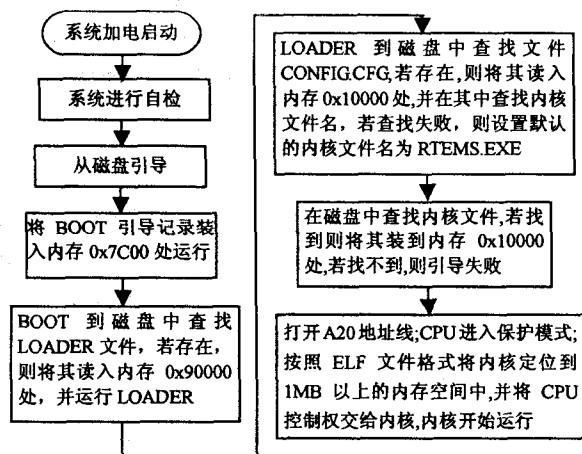


图 1 Bootloader 的工作流程图

2.1 BOOT 的设计

当 PC 机加电后, 系统首先进行自检, 然后寻找启动盘, 若从软盘启动, 计算机将读取软盘的 0 面 0 磁道 1 扇区的 BOOT 内容, 将其装入到内存地址 0x7C00 开始处, 然后 CPU 转到该处运行。

BOOT 的主要功能是在磁盘的目录区中找到 LOADER 文件目录, 并将其程序装入内存, 然后将 CPU 的控制权交给 LOADER 程序, 由 LOADER 负责初始化 RTEMS 工作环境并引导内核运行。BOOT 引导记录的算法描述如下:

- (1) 在显示屏上显示“Booting...”提示信息;
- (2) 软驱复位;
- (3) 在磁盘的目录区中寻找 LOADER 文件;

(4) 若 LOADER 文件存在, 则转(6);

(5) 若 LOADER 文件不存在, 显示提示信息“Boot fail!”, 关闭软驱马达, 程序进入死循环;

(6) 将 LOADER 读入内存 0x90000 处;

(7) CPU 转到 0x90000 处运行, 即将 CPU 的控制权交给 LOADER, 初始化和引导系统的工作就由 LOADER 来完成。

这段程序用汇编语言编写, 它经过汇编生成二进制文件, 使用工具软件将其写到软盘的 0 面 0 道 1 扇区。

2.2 LOADER 的设计

2.2.1 CONFIG.CFG 文件

CONFIG.CFG 文件存放有关内核的配置信息, 其中有一项信息表明将要启动的内核文件名, 其格式是: KERNEL = <kernel_name>, kernel_name 表示将要引导的内核文件名。

2.2.2 ELF 文件格式

ELF 文件格式^[4~6]是由 Unix 系统实验室设计的, 现在已成为一种最常用的可执行文件格式。ELF 目标代码文件可分为三类: 可重定位文件、共享文件和可执行文件。RTEMS 目标程序采用可执行文件格式。文中仅分析可执行文件的 ELF 结构。

可执行的 ELF 文件结构包括 ELF 文件头表、程序头表和节头表。文件头表在可执行文件的开头处, 作为总表描述整个 ELF 可执行文件的结构。它记录了程序头表在文件中的偏移、程序头表的表项数目、程序头表每一表项的字节长度、节头表在文件中的偏移、节头表的表项数目和节头表每个表项的字节长度。在 ELF 文件头后是程序头表, 程序头表中的每一个表项与每个程序段相对应。程序在运行时, 是以段为单位读入内存的。程序段的内容由若干节组成, 节的内容组合在一起连成一片构成程序段的内容。文中所设计的 LOADER 程序, 是根据内核的程序头表内容将内核装入到内存指定的位置。

2.2.3 实模式到保护模式的切换

PC 机启动时, CPU 工作在实模式下, 而 RTEMS 需要在保护模式下运行, 因此, LOADER 程序将内核装入内存后, 需要将 CPU 的运行模式由实模式转换到保护模式下, 才可启动内核的运行。进入保护模式的主要步骤:

- (1) 准备全局描述符表 GDT (Global Descriptor Table)。在保护模式下, 每一个段都有一个相应的描述符来描述, 为了便于组织管理, 将描述符组织成线性表, 形成描述符表。GDT 含有每一个任务都可能或可以访问的段的描述符。

(2) 用指令 lgdt 将 GDT 表的起始地址装入 gdt 寄存器中。

(3) 打开 A20 地址线。在实模式下 CPU 可访问的存储空间为 1MB, 即访问的最大地址为 0xFFFF, 当 CPU 试图访问超过 1MB 空间时, 系统并不发生异常, 而是回卷, 即实际访问的地址为该地址的低 20 位构成的地址。为了向上兼容, 在设计 PC 机时, IBM 使用 8042 键盘控制器来控制第 20 个地址位, 这就是 A20 地址线, 如果不被打开, 第 20 个地址位将总是零。

(4) 置 cr0 的 PE 位。cr0 是 CPU 的一个控制寄存器, 其中 PE 位是 cr0 的第 0 位, 它标志 CPU 是实模式(0)还是保护模式(1)。

(5) 跳转, 进入保护模式。

源程序如下:

```
lgdt [GdtPtr];加载 GDT
cli
in al, 92h
or al, 00000010b
out 92h, al;打开地址线 A20
mov eax, cr0
or eax, 1
mov cr0, eax;设置 PE 位为 1
jmp dword <CODE_SEL>: <OFFSET_VIRTUAL>;真正进入保护模式
```

2.2.4 LOADER 程序的设计

LOADER 程序负责将内核装入指定位置, 它首先在磁盘中查找 CONFIG. CFG 文件, 若找到, 将其读入内存, 并查找内核文件名。若在磁盘中没找到 CONFIG. CFG 文件或在 CONFIG. CFG 文件中没有发现参数 KERNEL, 则默认的内核文件名为 RTEMS. EXE; 其次, 将内核文件装到内存地址 0x10000 开始的空中; 第三, 关闭中断, 打开 A20 地址线, 对进入保护模式前进行初始化; 第四, 进入保护模式, 将保存在内存地址 0x10000 处的内核程序, 按照 ELF 文件的格式将内核定位到 0x100000 以上的内存空间中, 并将 CPU 的控制权交给内核程序, 引导程序 LOADER 完成其使命, 内核程序开始运行。

LOADER 程序算法描述如下:

(1) 软驱复位;

(2) 在软盘中查找配置文件 CONFIG. CFG, 若没有找到, 则转(4);

(3) 将 CONFIG. CFG 读到内存 0x10000 处, 并查找 KERNEL 参数, 若该参数存在, 则设置内核文件名为 KERNEL 的值, 转(5);

(4) 设置默认内核文件名为 RTEMS. EXE;

(5) 软驱复位;

(6) 在软盘目录表中查找内核文件, 若找到转(8);

(7) 在软盘中没有找到内核文件, 显示信息“Kernel file do not exist!”, 关闭软驱马达, 程序进入死循环;

(8) 将内核文件读到内存 0x10000 处, 并关闭软驱马达;

(9) CPU 进入保护模式;

(10) 调用 ELF 加载程序, 将存储在内存 0x10000 处的 ELF 文件格式的内核移到内存地址 0x100000 开始处;

(11) 程序跳转到 0x100000 处, 内核开始运行, LOADER 装入程序运行结束。

在 LOADER 程序中, 调用 ELF 文件加载程序的作用是将存储在内存中的内核, 按照 ELF 程序头的要求将指定的内核代码移到内存指定位置, 其算法描述如下:

1) 从 ELF 头表中获取程序头数目, 并保存在变量 cx 中;

2) 设置 esi 指向程序头表的第一个程序段;

3) 读取 esi 所指程序头表;

4) 若该程序段是不可装入的段, 转(6);

5) 将该程序段复制到该程序头表要求的内存中去;

6) 使 esi 指向下一个程序段;

7) $cx = cx - 1$;

8) 如果 cx 等于 0, 则装入结束, 否则转(3)。

加载 ELF 程序到指定位置的源程序如下:

```
LOAD ELF: mov cx, word [0x10000 + 0x2C];程序头数目保存在
cx
movzx ecx, cx
mov esi, [0x10000 + 0x1C];程序头表在文件中的偏移量
add esi, 0x10000;程序头表在内存的实际位置
Begin: mov eax, [esi + 0]
cmp eax, 0
jz NoAction;该程序段是不可装入的段, 转 NoAction
mov eax, [esi + 0x4];该段的偏移量
add eax, 0x10000;该段应存放的内存地址
push esi
mov edi, [esi + 0x8];该段的虚拟地址
mov ebx, [esi + 0x10];该段的大小
mov esi, eax;该段在内存中的地址
L1: mov al, [ds:esi]
inc esi
mov byte [es:edi], al;移动 1 个字节
inc edi
```

表 2 约简后的决策表

U	C1	C3	C4	D
1	1	1	0	0
2	1	1	1	0
3	2	1	0	1
4	3	1	0	1
5	3	2	0	1
6	3	2	1	0
7	2	2	1	1
9	1	2	0	1
...

其中,以规则 7 为例,其表示:

if 辅助线与圆有两个交点且输入条件中有“切线”标志,同时辅助线类型为连接

then 此输入条件就是作辅助线的依据条件

所开发的几何专家系统规则库经过约简后,不但约掉了规则中的冗余属性,同时约掉了冗余决策规则,使规则库变得精炼、简洁,易于维护,从而大大提高了系统的运行效率。最终,使初中几何专家系统的规则库的规则数目减少为 2700 多条,运行效率提高了 12% 左右。

3 结束语

知识获取在构建整个专家系统的过程中所占的地位举足轻重。探讨了初中几何专家系统领域内的知识

获取及实现的一般方法,解决了知识获取中的难点,然后利用基于粗糙集的约简理论来消除和减少规则库的冗余,使得平面几何系统规则库中的规则精炼、简洁,易于维护,同时大大提高了系统的效率。系统运行结果证明,此分析方法是有效的,有利于问题域的求解与实现。

参考文献:

- [1] 田盛丰. 人工智能原理与应用——专家系统、机器学习、面向对象的方法[M]. 北京:北京理工大学出版社,1998:12-59.
- [2] 蔡自清, Durkin J, 龚 涛. 高级专家系统:原理、设计及应用[M]. 北京:科学出版社,2005:33-66.
- [3] 刘 东. 知识管理的基本过程与知识的分类管理模式[J]. 南京政治学院学报,2002,18(6):44-47.
- [4] 陈 平, 褚 华. 软件设计师教程[M]. 北京:清华大学出版社,2004:222-237.
- [5] 亿珍珍, 赵 克, 许 威. 基于粗集的知识库冗余性化简研究[J]. 计算机工程与设计,2004,25(10):1731-1733.
- [6] 王国胤. Rough 集理论与知识获取[M]. 西安:西安交通大学出版社,2001:134-139.
- [7] 曾黄磷. 粗集理论及其应用——关于数据推理的新方法[M]. 重庆:重庆大学出版社,1995:55-125.
- [8] 张文修, 吴伟志, 梁吉业, 等. 粗糙集理论与方法[M]. 北京:科学出版社,2003:26-40.

(上接第 155 页)

```
dec ebx;计数器减一
jnz L1;循环
pop esi
NoAction:add esi,0x20;指向下一个程序段
dec ecx
jnz Begin
LoadEnd;
```

3 结束语

Bootloader 的设计使内核的启动脱离 GNU GRUB 的束缚,真正让软件开发人员了解和控制程序的运行,虽然笔者是以软盘为例来实现的,但在实际应用中可在虚拟机环境中使用。首先做一个软盘映像文件;其次,使用二进制编辑软件将该软盘映像文件的前 512 字节内容用 BOOT 的内容替换;第三,将该软盘映像文件作为一个磁盘挂在文件系统中,或利用磁盘工具,将 LOADER、CONFIG.CFG 和内核文件复制到映像文件中,再卸载该映像文件,然后在 Bochs、QEMU 或 VMware 等虚拟机软件中启动。通过虚拟机可方便地对内核进行调试,不需要每修改一次内核程序,就启动一次机器。通过设计 Bootloader 可以简化系统的启动

过程,使系统能更快地投入运行。当然,Bootloader 的设计还不太完善,因为在装入内核代码时,Intel 80x86 CPU 工作在实模式下,只能访问 1MB 的空间,而内核被装到 0x10000 至 0x90000 存储空间中,要求内核的长度不能大于 512kB。如何打破这一限制将是今后进一步研究解决的问题。

参考文献:

- [1] Straumann T. Open Source Real Time Operating System Overview[C]//8th International Conference on Accelerator & Large Experimental Physics Control Systems. San Jose, California;[s. n.],2001.
- [2] 陈海军,申卫昌,史 颖. 嵌入式系统引导程序详探[J]. 计算机技术与发展,2006,16(1):123-128.
- [3] 徐亚鹏,谢凯年. 用 U-Boot 构建 IXP2350 目标系统的引导程序[J]. 计算机技术与发展,2007,17(5):10-14.
- [4] 于 渊. 自己动手写操作系统[M]. 北京:电子工业出版社,2006.
- [5] 何先波,唐宁九,吕 方,等. ELF 文件格式及应用[J]. 计算机应用研究,2001,18(11):144-145.
- [6] 倪继利. Linux 内核分析及编程[M]. 北京:电子工业出版社,2005.