

Project1 Bootloader 设计文档

中国科学院大学

[姓名] 李国峰

[日期] 2021 年 1 月 24 日

1. Bootblock 设计

- (1) Bootblock 主要完成的功能
在计算机启动阶段，需要 BIOS 将操作系统引导程序加载到内存中，然后通过跳转指令从 BIOS 跳转到引导程序执行。
- (2) Bootblock 如何调用 SD 卡读取函数
根据 MIPS64 寄存器的相关定义，要调用函数，需要先将函数所需的参数放到 \$a0~\$a7 寄存器中，再将需要调用的函数 printstr 放到一个临时寄存器中，再使用 jal 指令跳转到函数运行。
- (3) Bootblock 如何跳转至 kernel 入口
使用无条件跳转指令 j 直接跳转到 kernel 所在地址执行。
- (4) 任何在设计、开发和调试 Bootblock 时遇到的问题和解决方法
对伪指令的了解有所欠缺，不了解 MIPS32 和 MIPS64 在汇编指令格式上的区别，因此一直使用的是 32 位的指令导致出错。有了伪指令，就可以在一定程度上简化汇编指令的使用。

2. Createimage 设计

- (1) Bootblock 编译后的二进制文件、Kernel 编译后的二进制文件，以及写入 SD 卡的 image 文件这三者之间的关系
image 文件是由 Bootblock 和 kernel 编译后的二进制文件去掉 ELF 头和文件头的代码段和数据段经过链接形成的。
- (2) 如何获得 Bootblock 和 Kernel 二进制文件中可执行代码的位置和大小？
Bootblock 和 kernel 二进制文件开头都是 ELF 头和程序头，ELF 头结构体中的 e_phoff 表示程序头里 ELF 头的偏移量，可以凭借次找到程序头的位置，程序头结构体中的 p_offset 和 p_filesz 分别表示可执行代码段的偏移量和大小。
- (3) 如何让 Bootblock 获取到 Kernel 的大小，以便进行读取
Kernel 是内核，二进制文件大小可能很大，为了方便读取，可以根据 kernel 的可执行文件的大小计算所占扇区个数，然后一个扇区一个扇区进行读取写入。

3. 关键函数功能

(1) 汇编语言中函数的调用过程

```
# 2) call BIOS read kernel in SD card
ld  $a0, kernel
li  $a1, 0x200
li  $a2, 0x200000
ld  $t2, read_sd_card
jal $t2
```

如上所述，先放置好参数和函数至相应位置，再使用跳转指令跳转到指令执行。

(2) 字符回显

```
while(1)
{
    if((*flag) & 0x01 == 1)
        printch(*input);
}
return;
```

只要串口状态寄存器表示有字符输入，则读取串口数据寄存器并回显。

参考文献

[1] 中国科学院大学操作系统实验手册 Project1-MIPS (2020 秋季学期)

■