

1. 介绍

实现一个 Linux 工具，将启动程序和内核编译为一个 MIPS 架构支持的操作
系统镜像。

编译程序编译出的可执行文件需要通过一个 Linux 工具（本实验中为
createimage）转换为内核镜像，内核镜像被写在磁盘中，供系统启动时加载使用。
该工具将可执行文件转换为硬件支持的格式，可以被直接加载和运行。

1.1. 需要了解的部分

- ELF文件格式

2. 初始代码

2.1. 文件介绍

- bootblock.s: 程序运行最开始执行的程序，上一阶段完成
- kernel.c: 一个小的内核程序，最终输出一个字符串
- createimage.c: 生成内核镜像的工具，初始提供一个框架，本次任务需
要完成
- createimage: 本次实验不允许使用该文件
- Makefile: 编译配置文件，将createimage的编译选项去掉注释(第12~1
3行)
- ld.script: 链接器脚本文件

2.2. 获取:

课程网站。

2.3. 运行

`createimage` 为提供的可执行文件, 当 `createimage.c` 实现完成后, 将 `Makefile` 中的 `createimage` 项去掉注释。

`make` 命令编译文件

`make clean` 对编译产生的文件进行清除

`sudo dd if=image of=/dev/sdb` 将产生的 `image` 写进 SD 卡中

在 `minicom` 中执行 `loadboot` 运行程序

3. 任务

3.1. 设计和评审

帮助学生发现设计的错误, 及时完成任务。学生需要对这次的作业进行全面考虑, 在实现代码之前有清晰的思路。学生讲解设计思路时可以用不同的形式, 如伪代码、流程图等, 每个组使用 `PPT` 的形式呈现 (不要超过十分钟)。

设计介绍

- 创造内核镜像: 可执行文件(ELF)的特点? 怎样读取可执行文件? 可执行文件 `bootblock` 和 `kernel` 在内核镜像的什么位置存放?

3.2. `createimage` 开发

3.2.1. 要求

实现一个 `Linux` 工具, 将 `bootblock` 和 `kernel` 结合为一个操作系统镜像, 并提供操作系统镜像的一些信息。其中 `bootblock` 存放在镜像的第一个扇区, `kernel` 存放在镜像的第二个扇区。一共需要实现以下函数:

- `read_exec_file()`; 读取ELF格式的一个文件。
- `write_bootblock()`; 将可执行文件 `bootblock` 写入内核镜像“`image`”文件中。

- `write_kernel()`; 将可执行文件`kernel`写入镜像文件“`image`”文件中。
- `count_kernel_sectors()`; 计算`kernel`有多少个扇区
- `record_kernel_sectors()`; 将`kernel`的扇区个数写入`bootblock.s`一个参数中。该函数需要自己在`start code`中进行补充。
- `extend_opt()`; 打印出制作`image`的文件的一些信息。

3.2.2. 注意事项

为了实现一个 `createimage`，需要了解 ELF 文件格式（课程网站上提供 ELF 文件格式介绍）：

- ELF文件头，以及它的`e_phnum`和`e_phoff`等域。
- 可执行程序文件头，以及`p_offset`和`p_filesz`等域。

--extend 选项提供制作内核镜像的文件的一些信息(`bootblock` 和 `kernel`)。

这里请分别输出 `bootblock` 和 `kernel` 文件的一些信息（在 `extend_opt` 中实现）：

- 文件的大小和扇区个数
- `segment`在文件中的偏移
- 在内存中，`segment`的偏移
- 文件的`segment`的大小
- 在内存中`segment`的大小
- 文件制作的 `image`的对齐字节数

`Record_kernel_sectors()`函数将内核的大小写入 `bootblock.s` 的一个变量处（这里设置为 `os_size`）。`Bootblock.s` 根据该参数获得内核的扇区个数，进而从 SD 卡中将完整的内核读取出来。在任务三中，可以手动设置读内核的大小（直接传递数值，例如静态的将 `os_size` 设置为 1），任务四要求通过 `createimage` 来传递内核大小参数，读取内核。

4. 测试

`createimage.c` 可以正确的将 `bootblock` 写入硬盘的第一扇区，将 `kernel` 从第二扇区开始写入。当代码正确时，会打印出“*It's a kernel!*”等字样。

5. 参考资料

- ELF文件格式文档：课程网站