

## 任务三：Bootloader 开发

### 1. 介绍

实现一个操作系统启动程序。

在 MIPS 架构下，启动程序被放在了启动设备（U 盘，硬盘等）的第一个扇区，系统启动时该扇区被 `loadboot` 自动加载在内存地址 `0xa0800000` 处，然后从它的入口地址 `0xa0800028` 开始执行。启动程序需要将剩余的操作系统程序加载的内存中，即内核。

#### 1.1. 需要了解的部分

- MIPS架构以及汇编语言
- 计算机启动操作系统过程

### 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，需要实现读取内核的代码
- `kernel.c`: 一个小的内核程序，最终输出一个字符串
- `createimage.c`: 在任务四中实现，这里先忽略
- `createimage`: 生成的内核镜像的Linux工具，用来将`bootblock`生成一个`image`文件
- `Makefile`: 编译配置文件
- `ld.script`: 链接器脚本文件

## 2.2. 获取：

课程网站。

## 2.3. 运行

*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（*/dev/sdb* 为插入的 SD 卡盘符，根据需要修改）

在 *minicom* 中执行 *loadboot* 运行程序

可以采用反汇编进行调试，调试命令为 *mipsel-linux-objdump -d*

# 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（和任务二的设计评审一起不要超过十分钟）。

### 设计介绍

- 将内核从硬盘移到内存：怎样从磁盘读内核？将内核放在内存什么位置？

## 3.2. bootblock 开发

### 3.2.1. 要求

bootblock 为内核启动程序，需要完成以下任务：

- 将内核从硬盘加载到内存中

### 3.2.2. 注意事项

PMON 中的读盘函数的地址为 0x8007b1a8, 有三个参数，分别为：

- 第一个参数为读取的目的地址，即读取的数据在内存存放的位置
- 第二个参数为SD卡内部的偏移量，从该处开始读取
- 第三个参数为要读取的字节数

内核读取完成后需要跳转到内核的开始处执行（在 bootblock.s 处设置），此时内核的第一条执行指令地址为内核读取到内存的位置+内核中 main 函数在内核文件中的偏移地址。例如你将内核读取在内存的 0xa0800200 处，而 kernel.c 中 main 函数的偏移地址为 0x6c, 那么在 bootblock.s 读完内核之后就需要跳转到 0xa080026c 处执行。

## 4. 测试

完成功能，最终会输出“*It's a kernel!*”字符串。

**Figure 1** | **Flowchart illustrating the study design.**