

Lab1 实验报告

一、实验目标

本实验的目标是实现 multiboot header，启动 OS 内核，并在 VGA 输出姓名学号

二、实验原理

Multiboot 启动协议是一个标准，允许多个操作系统在同一系统上共存。这个协议规定了加载器和操作系统之间的接口，确保任何遵循 Multiboot 协议的操作系统可以通过任何兼容的引导程序启动。

QEMU 是纯软件实现的虚拟化模拟器，可以模拟多种处理器和系统。GRUB 是一个功能强大的引导加载器，支持多种操作系统和文件系统。在本实验中，使用 QEMU 模拟器模拟系统环境，并使用 GRUB 或直接使用 QEMU 运行程序。

VGA 是一个计算机显示标准。在文本模式下，VGA 文本缓冲区位于 0xB8000，每个字符需要 2 个字节，其中一个字节表示字符的 ASCII 码，另一个字节表示字符的颜色属性。

三、源代码说明

使用变址寄存器 `edi` 存储指向 VGA 文本缓冲区内内存地址的指针，故先用 `movl $0xB8000, %edi` 将 VGA 起始地址存入 `edi`，然后使用形如 `movl $0x2f452f48, (%edi)` 指令每次将要输出的两个字符存入 VGA 地址中，因为占四个字节，故之后使用 `addl $4, %edi` 移动指针位置到空位置。

四、代码布局说明

.S 中 `multiboot_header` 在内存中占据了 12 个字节，因为它由 3 个 4 字节（32 位）的 `long` 型构成，`magic`，`flags`，`checksum` 都是 multiboot 协议的必要参数。对齐方式是 4 字节对齐因为使用了 `.align 4` 指令。

VGA 内存部分从 0xB8000 开始分配。每个字符占用 2 个字节，一个字节表示字符的 ASCII 码，另一个字节表示颜色属性。同时，`.code32` 规定了 32 位代码段，故 `move` 每次可以写入两个字符及其颜色属性。因此，在两个 `movl` 指令中，目标地址相差 4。最后使用 `hlt` 暂停。

五、编译过程说明

首先使用 `make` 指令生成 `bin` 文件，

然后输入 `qemu-system-i386 -kernel multibootHeader.bin` 指令使用 `qemu` 运行编译好的 `bin` 文件

六、实验结果

