

lab1_实验报告

PB22030892 刘铠瑜

一、实验原理

本实验基于Multiboot协议实现了一个最简操作系统内核，能够在VGA屏幕和串口上输出特定内容。Multiboot协议定义了bootloader与操作系统内核的接口规范，确保兼容性。核心要求如下：

1. Multiboot Header：

- 必须包含三个32位字段：`magic`（固定值 `0x1BADB002`）、`flags`（本实验设为 0）、`checksum`（满足 `magic + flags + checksum = 0`）。
- Header必须位于内核映像的前8192字节内，且起始地址对齐到4字节。

2. VGA输出：

- VGA显存起始地址为 `0xB8000`，每个字符占2字节（ASCII码+属性）。
- 属性字节定义前景色、背景色及闪烁效果，例如 `0x2F` 表示绿底白字。

3. 串口输出：

- 串口端口地址为 `0x3F8`，通过 `outb` 指令直接发送字符ASCII码至该端口。

二、代码运行与编译说明

1. 编译过程

- 工具链：**使用 `gcc` 编译汇编代码，`ld` 链接生成内核文件。
- Makefile：**

```
ASM_FLAGS = -m32 --pipe -Wall -fasm -g -O1 -fno-stack-protector
multibootHeader.bin: multibootHeader.S
    gcc -c ${ASM_FLAGS} multibootHeader.S -o multibootHeader.o
    ld -n -T multibootHeader.ld multibootHeader.o -o multibootHeader.bin
```

执行 `make` 后生成 `multibootHeader.bin`。

2. 运行方法

通过QEMU启动内核：

```
qemu-system-i386 -kernel multibootHeader.bin -serial stdio
```

- `-kernel`：指定内核文件。
- `-serial stdio`：将串口输出重定向到终端。

三、代码与地址空间分析

1. 源代码说明（`multibootHeader.S`）

- **Multiboot Header定义：**

```
.section .multiboot_header
    .long 0x1BADB002    // magic
    .long 0             // flags
    .long -(0x1BADB002) // checksum
```

`checksum` 通过 `-(magic + flags)` 计算，确保校验和为0。

- **VGA输出：**

```
movl $0x2f652f48, 0xb8000    // "He" (0x48=H, 0x65=e, 属性0x2F)
movl $0x2f6C2f6C, 0xb8004    // "ll"
... (后续代码输出完整字符串)
```

每个 `movl` 指令写入4字节（2字符），属性统一为绿底白字。

- **串口输出：**

```
movw $0x3F8, %dx
movb $'H', %al
outb %al, %dx
... (逐个字符输出至串口)
```

通过 `outb` 指令将字符ASCII码写入端口 `0x3F8`。

2. 地址空间分配（`multibootHeader.ld`）

链接脚本指定代码段起始于1MB地址（`. = 1M`），确保Multiboot Header位于内核映像前部：

```

SECTIONS {
    . = 1M;
    .text : {
        *(.multiboot_header) // Header段
        . = ALIGN(8);        // 8字节对齐
        *(.text)             // 代码段
    }
}

```

• 偏移量:

- `.multiboot_header` 位于1MB处 (0x100000) , 占12字节。
- `.text` 代码段从1MB+16字节 (0x100010) 开始, 按8字节对齐。

四、实验结果

The screenshot shows a development environment with a source code editor on the left and a terminal on the right. The source code editor displays assembly code for a multiboot header. The terminal shows the output of a make command and a QEMU virtual machine running, displaying the text "Hello world!PB22030892_LaiKaiyu".

```

15 start:
26 movl $0x2f382f30, 0xb8024 /* 80 */
27 movl $0x2f322f39, 0xb8028 /* 29 */
28 movl $0x2f4C2f5F, 0xb802C /* L_ */
29 movl $0x2f752f69, 0xb8030 /* ui */
30 movl $0x2f612f4B, 0xb8034 /* aK */
31 movl $0x2f792f69, 0xb8038 /* yi */
32 movw $0x2f75, 0xb803C /* u */
33
34
35
36 /* 串口输出Hello world!PB22030892_LaiKaiyu */
37 movw $0x3F8, %dx
38 movb $'H', %al
39 outb %al, %dx
40 movb $'e', %al
41 outb %al, %dx
42 movb $'l', %al
43 outb %al, %dx
44 movb $'l', %al
45 outb %al, %dx

```

```

Microsoft Windows [版本 10.0.22631.4602]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\86152\Desktop\OS\Lab\lab_1\src>wsl
aurora@Aurora:/mnt/c/Users/86152/Desktop/OS/Lab/lab_1/src$ make
make: 'multibootHeader.bin' is up to date.
aurora@Aurora:/mnt/c/Users/86152/Desktop/OS/Lab/lab_1/src$ qemu-system-i386 -kernel multibootHeader.bin -serial stdio
Hello world!PB22030892_LaiKaiyu

```