

Lab1-实验报告

PB18051098 徐碧涵

原理说明

本实验通过 qemu 启动支持 multiboot 启动协议的内核，利用 vga 输出 ‘hello world, PB18051098xubihan’。

源代码说明

multibootHeader.s 文件源代码说明

```
|section.multiboot_header:                                /*遵循multiboot启动协议编写multiboot_header*/
    /* magic */
    .long 0x1BADB002
    /* flags */
    .long 0
    /* checksum */
    .long -(0x1BADB002 + 0);
.globl start                                              /*声明全局变量start*/

#output hello word,PB18051098xubihan by vga
#every output 2 characters ,the address +4
/*根据vga代码要求将所要输出字符通过mov移到vga显存中*/
start:                                                    /*确定代码入口*/
.text.multiboot_entry:
    movl $0x2f652f68,0xB8000
    movl $0x2f6c2f6c,0xB8004
    movl $0x2f002f6f,0xB8008
    movl $0x2f6f2f77,0xB800c
    movl $0x2f6c2f72,0xB8010
    movl $0x2f2c2f64,0xB8014
    movl $0x2f422f50,0xB8018
    movl $0x2f382f31,0xB801c
    movl $0x2f352f30,0xB8020
    movl $0x2f302f31,0xB8024
    movl $0x2f382f39,0xB8028
    movl $0x2f752f78,0xB802c
    movl $0x2f692f62,0xB8030
    movl $0x2f612f68,0xB8034
    movl $0x2f002f6e,0xB8038

hlt
```

multibootHeader.ld 文件源代码说明

```
OUTPUT_FORMAT("elf32-i386","elf32-i386","elf32-i386") /*以elf32-i386格式输出文件*/
OUTPUT_ARCH(i386) /*建立x86架构下*/
ENTRY(start) /*代码入口*/

SECTIONS{
    . = 1M; /*表示在物理内存1M位置开始放代码*/
    .text : {
        *(.multiboot_header)
        . = ALIGN(8); /*按8个字节对齐存放代码段*/
        *(.text)
    }
}
```

makefile 文件代码说明

通过编译将.s 文件生成目标文件.o 文件再按.ld 文件对代码布局以及地址空间的要求生成.bin 文件。

```
/*通过make功能将.s文件按照.ld文件代码布局的要求在内存空间生成.bin文件*/
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
```

```
/*用于清除之前生成的.o,.bin文件*/
clean:
```

```
    rm -rf./multibootHeader.bin./multibootHeader.o
```

代码布局（地址空间）说明

代码在内存空间中的地址以及布局由 multibootHeader.ld 文件所确定，代码将从物理内存 1M 的位置开始存放，并按 8 个字节对齐，比如 multiboot_header 占据从 1M 地址开始的 12 个字节，然后其后代码从第 16 个字节之后开始存放。根据设置的代码入口地址 start 设置进程入口地址。

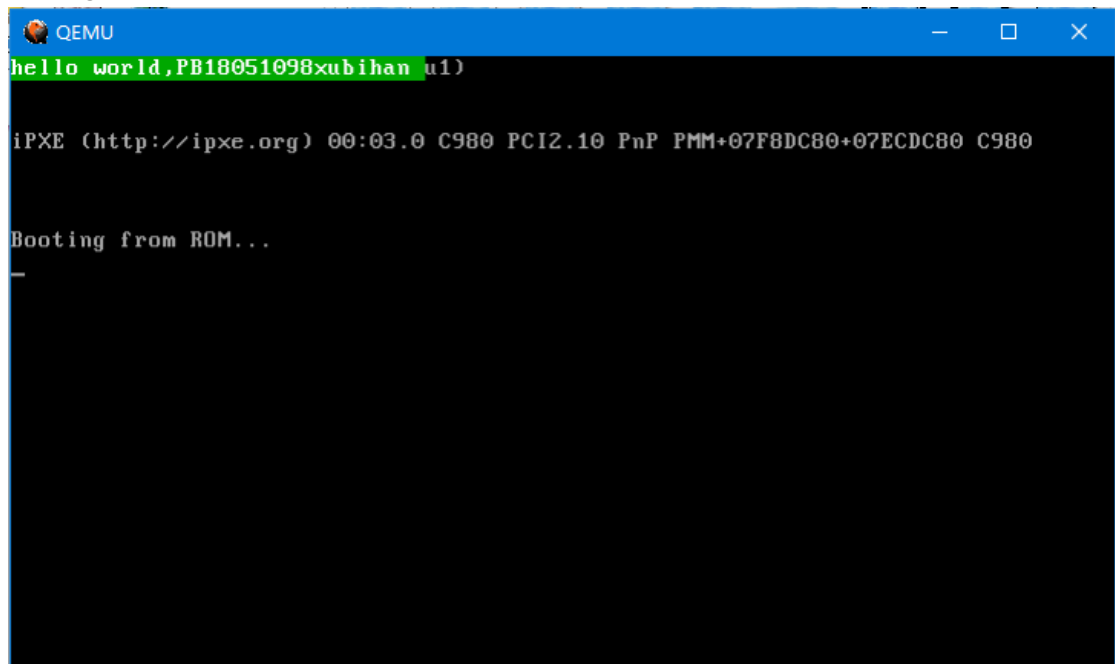
编译过程说明

通过 make -f makefile 命令将 multibootHeader.s 文件按照 multibootHeader.ld 文件代码布局以及地址空间的要求在内存空间生成.bin 文件。

运行和运行结果说明

通过 qemu-system-i386 -kernel multibootHeader.bin -serial stdio 命令通过 qemu 启动内核

从而利用 vga 输出 ‘hello world, PB18051098xubihan’。运行结果如下。

A screenshot of a QEMU terminal window. The title bar is blue with the QEMU logo and the text 'QEMU'. The terminal has a black background with white text. The first line is 'hello world, PB18051098xubihan u1)' in green. The second line is 'iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+07F8DC80+07ECDC80 C980'. The third line is 'Booting from ROM...'. The fourth line is a single hyphen '-'.

```
QEMU
hello world, PB18051098xubihan u1)
iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+07F8DC80+07ECDC80 C980
Booting from ROM...
-
```

遇到的问题及解决方案

一开始通过 vga 输出结果总是伴随背景闪烁，在代码段末尾添加 hlt 即可解决。.long 字段是按高地址到低地址，一开始我并没有弄清楚导致输出字符颠倒，将原先.long 内两个代表字符的数顺序调换即可。