



中国科学技术大学

University of Science and Technology of China

011174.01: Operating System 操作系统原理与设计

Project 1: Startup

陈香兰(xlanchen@ustc.edu.cn)

高效智能计算实验室, CS, USTC @ 合肥

嵌入式系统实验室, CS, USTC @ 苏州



温馨提示:



为了您和他人的工作学习，
请在课堂上**关机或静音**。

不要在课堂上接打电话。

Steps



中国科学技术大学
University of Science and Technology of China

- **【非必须】** 建立Ubuntu和主机的共享文件夹
- **【必须】** 了解必要的multiboot协议相关内容
- **【必须】** 安装QEMU，了解QEMU对multiboot的支持情况
可以使用其他虚拟机或模拟器
- **【必须】** 遵守multiboot协议，写multiboot header
- **【必须】** 直接在VGA的显存中写“helloworld”
- **【可选】** 使用串口输出“HELLOWORLD”
- **【必须】** 了解并掌握必要的汇编
- **【必须】** 了解Makefile和链接描述文件

1、共享文件夹



中国科学技术大学
University of Science and Technology of China

- **【原来就有】** 从Ubuntu中浏览主机中的盘符
(仅子系统方式适用，其他方式“具体问题具体分析”)
 - 以C盘为例，在Ubuntu中输入“mount”命令查看C盘挂载情况（在/mnt/c上）
- **【动手】** 建立Ubuntu用户目录下“workspace”到主机工作目录“workspace”的符号链接
 - 在Windows中选择一个文件夹位置（如，我的桌面），建立“workspace”子文件夹
 - 在Ubuntu用户目录下使用“ln -s”命令建立符号链接“workspace”，如
ln -s /mnt/c/Users/*你的Window用户名*/Desktop/workspace workspace
（完成后，使用“ls -la”查看是否成功）

2、Multiboot & QEMU



中国科学技术大学
University of Science and Technology of China

- 【浏览】 [Multiboot Specification version 0.6.96](#)
- 【动手】 QEMU: <https://www.qemu.org/>
 - 安装qemu: `sudo apt-get install qemu`
- 【知道】 Multiboot specification (i386) (version 0.6.96)
 - Multiboot-compliant bootloader sets the system to well-defined state, most notably:
 - **Transfer CPU to protected mode** to allow you access all the memory
 - Enable A20 line
 - Global descriptor table and Interrupt descriptor table are undefined
 - Multiboot-compliant OS kernels:
 - Can (and should) be in **ELF** format
 - **Must set only 12 bytes** to correctly boot



The layout of Multiboot header

Offset	Type	Field Name	Note
0	u32	magic	required, should 0x1BADB002
4	u32	flags	required, can be 0
8	u32	checksum	required, magic + checksum + flags = 0
12	u32	header_addr	if flags[16] is set
16	u32	load_addr	if flags[16] is set
20	u32	load_end_addr	if flags[16] is set
24	u32	bss_end_addr	if flags[16] is set
28	u32	entry_addr	if flags[16] is set
32	u32	mode_type	if flags[2] is set
36	u32	width	if flags[2] is set
40	u32	height	if flags[2] is set
44	u32	depth	if flags[2] is set

3、VGA输出简介



• 【非完整版，仅本实验用】

- 本实验中字符界面规格：25行80列
- VGA显存的起始地址：0xB8000
- 每个字符需要2个字节：一个用于存放字符的ASCII码，一个用于存放该字符的显示属性

Attribute								Character							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

• 显示属性

Attribute							
7	6	5	4	3	2	1	0
Blink	Background color			Foreground color			

• 直接写VGA显存可以输出信息

例如：`movl $0x2f4b2f4f, 0xB8000` 可以输出“OK”（绿底白字）

4、串口编程简介



中国科学技术大学
University of Science and Technology of China

- 【非完整版，仅本实验用】
- UART Universal Asynchronous Receiver/Transmitter
 - QEMU中，即使MyOS不对uart进行初始化，也可以正确输出
 - 端口地址：0x3F8
 - 输出字符的汇编代码为：

```
/*在没有初始化波特率等情况下，直接写UART，QEMU上不出错*/  
movb $0x46, %al /* 输出字母F */  
movw $0x3F8, %dx  
outb %al, %dx /* put char */  
nop  
nop
```


5、必要的汇编



- 【非完整版，仅本实验用】
- Multiboot header特有的数据段
 - 使用 “.section *SECNAME*”
- 32位代码段
 - “.text” “.code32”
- 定义一个标号，例如 “*XXLabel:*”
- “.globl *XXLabel*”
- “.align 4”
- “.long *XXX*”
- 可能用到的指令助记符：nop, mov[lwb], xor[lwb], out[lwb], in[lwb], and[lwb], jz, hlt

- 【非完整版，仅本实验用】

- Makefile的内容：

```
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
```

```
clean:
```

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

- 源代码有3个文件

- Makefile （提供，可根据需要修改）
- multibootHeader.S （自己编写）
- multibootHeader.ld （提供，可根据需要修改）

- 链接描述文件 “multibootHeader.ld” 的内容

```
OUTPUT_FORMAT("elf32-i386", "elf32-i386", "elf32-i386")
```

```
OUTPUT_ARCH(i386)
```

```
ENTRY(start)
```

```
SECTIONS {
```

```
    . = 1M;
```

```
    .text : {
```

```
        *(.multiboot_header)
```

```
        . = ALIGN(8);
```

```
        *(.text)
```

```
    }
```

```
}
```

部分汇编代码（仅供参考）



中国科学技术大学
University of Science and Technology of China

`.globl start` #一般都用start

/*此处，按照multiboot协议要求定义必要的参数*/

/*格式：XXX_ITEM_NAME=value*/

/*value可以是十六进制的（0x开头）、十进制的等等*/

/*为便于阅读，可以适当增加一些注释和空行*/

/*此处开始，按协议标准来定义必须的multiboot header*/

`.section ".xxx"` #先起一个section的名字

/*使用.long和前面定义的参数构建头结构，每次32位，格式为：`.long XXX_ITEM_NAME`*/

`.text` #进入代码段

`.code32` #32位代码

`start:` #这个跟第一行的声明要一致

/*下面屏幕输出OK或其他字符序列*/

/*使用指令`movl $0x12345678, 0xB8000`*/ #0x1234和0x5678各自输出1个字符

/*根据需要输出多个字符，也可以使用其他mov指令*/

/*可以根据需要使用nop指令隔开不同功能片段，也可以适当使用空行*/

/*根据需要初始化串口*/

/*根据需要串口输出你的字符序列，详见前面串口编程简介*/

/*实验结束，让计算机停机，方法：使用hlt指令，或者死循环*/



- 为支持QEMU的虚拟机界面，需要安装Xserver
 - 请自行了解一下Window10中ubuntu的原理、WSL、图形界面相关的内容

Q & A