学号：221220113

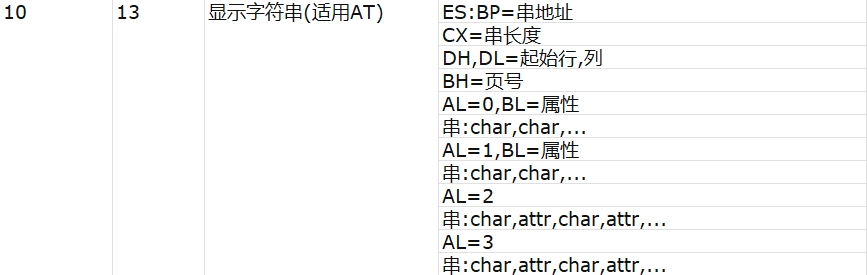
姓名：仲瑞泉

邮箱：1448853658@qq.com

Lab1.1

这一小节的实验流程：BIOS自检，TODO：显示HELLPOWORLD。

实模式下可以利用BIOS中断**INT $0x10(%ah = 0x13)**显示字符串。用到的参数如下：



在中断前，设置好参数：

message地址 -> %bp

message长度­ -> %cx

中断功能0x13 -> %ah

仅显示字符，光标跟随移动 0x01 -> %al

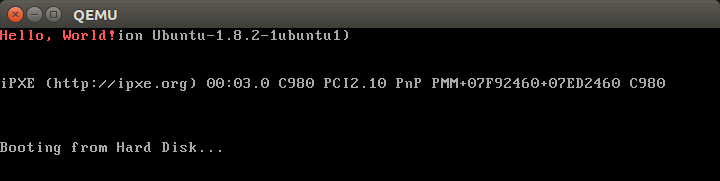
视频页码0x0 -> %bh

显示属性0x0c -> %bl

(关于显示属性，没有找到详细资料，尝试更改为0x0d,0x00，发现与字符颜色有关)

显示位置0行0列 0x0000 -> %ex

设置好参数后，通过INT $0x10即可显示字符串，1.1节实验完成。



Lab1.2

这一小节的实验流程：BIOS自检，TODO：引导CPU进入保护模式，并在保护模式下显示HELLPOWORLD。

(1)要进入保护模式，主要需要分配好GDT，以进行保护模式下的寻址。

根据GDT初始化的具体格式：



（其中type含义见参考资料lab1.pdf）

代码段如下拼凑：

Base:0x0

Limit:0xffffffff

Type:可执行、可读,代码段，1010

数据段如下拼凑：

Base:0x0

Limit:0xffffffff

Type:可读写,数据段，0010

还有一个段用于显存映射，下面再进行解析。

到此，GDT分配完成，其余细节比较简单，跟着注释做即可。

（2）要在保护模式下显示HELLOWORLD，需要进行显存映射。

一般显存起点在0xb8000，要在保护模式下对0xb8000处读写，可以为显存分配一个段基址为0xb8000的段，将其索引存入段选择子 %gs (意义上和graphic segment可以搭个边)，通过 %gs : OFFSET 进行索引。

其GDT初始化如下拼凑：

Base:0xb8000

Limit:0xffffffff

Type:可读写,数据段，0010

另外，VGA显存中，某字符的具体位置如下：

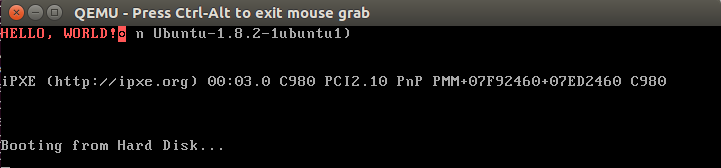
offset = (row \* 80 + col) \* CHARACTER\_SIZE

其中CHARACTER\_SIZE=2，即一个字符占两个字节（颜色码+字符码）。

例如，HELLOWORLD!总计占据一行，假设在第0行，则第i个字符的偏移量为2\*i。

对于HELLO,WORLD!\n\0,共13个字符，利用eax中的ah作为颜色码0x0c，利用al存放ascii码，依次放入显存即可。

1.2节实验完成。



Lab1.3

这一小节的实验流程：BIOS自检，TODO：进入保护模式，加载引导程序到指定位置，执行引导程序。

在进入保护模式上与lab1.2并无区别。

为了加载引导程序到指定位置，可以利用给出的接口readSect(void\* dst, int offset),其作用为将offset扇区的内容读入dst位置处。

在app/Makefile中可以找到规定的程序入口地址0x8c00。

也知道第一扇区中装载的就是引导程序app.s，于是将第一扇区内容读入0x8c00，然后跳转0x8c00执行程序显示HELLOWORLD。

1.3节实验完成。

