实现键盘输入

继续使用10h号中断来完成一些内容

熟悉使用16h号中断—键盘控制中断的功能，包括获取键盘输入状态，以及如何处理。

分成了几个部分来完成。

1.清空屏幕（在原先的基础上做了一部分改动）

2.显示一个字符串

3.键盘输入（设置了一些控制字符的汇编处理）

1：清空屏幕（只介绍新添加的内容）

movb $0x07, %ah #功能号07，这里调用07和06结果一样，上卷下卷都是一样的。

movb $0, %al #上卷全部行，即清屏

movb $0, %ch #左上角行

movb $0, %cl #左上角列

movb $24, %dh #右下角行

movb $79, %dl #右下角列

movb $0x07, %bh #卷入行的属性

int $0x10

movb $0x02, %ah #这时候光标的位置是在中间的位置，需要重置一下位置

movw $0x00, %dx #光标的行与列都是0和0

movb $0x00, %bh #页号是0

int $0x10

ret

一开始时候清屏结束以后，光标停留在中间的位置。调用10h中断的02h号功能来重置光标。

设置dx=0，表示光标的行号与列号为0，设置bh=0表示页号为0。重置结束以后可以在屏幕的左上角开始显示光标。

另外有关清屏，调用06h与07h号功能获取一样的结果。

2：显示字符串

movw $BootMessage, %ax

movw %ax, %bp #ES:BP = 串地址

movw $16, %cx #CX = 串长度

movw $0x1301, %ax #AH = 13, AL = 01h,这里面是调用了10h号中断的13号功能

movw $0x00c, %bx #页号为0(BH=0) 黑底红字(BL=0Ch,高亮)

movb $0, %dl #表示起始行号

int $0x10 #调用10h号中断

call Fill

ret

BootMessage: .ascii "hello, BLCU OS,My name is ChenPeng"

详细参见实验0.1

3：键盘输入Typing:

movb $0x00, %ah #ah = 00h，从键盘读取字符，读取的内容保存到al中

int $0x16 #调用BIOS16h中断，是键盘控制中断

cmpb $0x08, %al #比较输入的字符是不是Backspace键

jz Back

cmpb $0x0D, %al #比较输入的字符是不是Enter键

jz Enter

# cmpb $0x24, %al #比较输入的字符是不是Up键

# jz Back

# cmpb $0x28, %al #比较输入的字符是不是Down键

# jz Back

# cmpb $0x25, %al #比较输入的字符是不是Left键

# jz Left

# cmpb $0x27, %al #比较输入的字符是不是Right键

# jz Back

movb $0x0e, %ah #ah = 0eh，在光标处显示字符。光标随着字符而移动

int $0x10 #调用BIOS10h中断，显示字符

call Typing

ret

有关键盘的BIOS中断服务程序是16h号中断。INT $0x15h来进入。

首先先从键盘读取一个字符。功能号是00H，读取的数据（ASCII码）保存到了AL寄存器中。

之后判别一下寄存器中保存的输入是不是特殊控制字符的ASCII码

Backspace ：跳转到Back

Enter：跳转到Enter

若不符合以上情况。先将AH置为0x0e，表示准备调用10h中断的0Eh号服务：在光标处显示字符。

再通过INT $0x10h，调用显示中断。

最终再call Typing 实现循环输入。

在Back中：

movb $0x0e, %ah #ah = 0eh，在光标处显示字符。光标随着字符而移动

movb $0x08, %al #先后退一个字符

int $0x10

movb $0x20, %al #输出空格

int $0x10

movb $0x08, %al #再后退一个字符

int $0x10

call Typing

首先后退一个字符。让光标停留在准备删除的字符前面

在输出一个空格，让空格来覆盖准备删除的字符，此处光标停留在空格处。

为了能够实现连续删除， 需要将光标停留在空格前面。

最后再call Typing 实现循环输入。

在Enter中：

movb $0x0e,%ah #ah = 0eh，在光标处显示字符。光标随着字符而移动

movb $0x0a,%al #换行，这时候光标在垂直位置上保持当前值。

int $0x10

movb $0x0d,%al #归位

int $0x10

call Typing

首先先将0x0a存入到al中，实现换行。但是这个时候换行后的光标在垂直位置上没有改变。

在将0x0d存入到al中，实现归位，将光标停留在这一行的最初始的位置。

最后再call Typing 实现循环输入。

这里只是实现了Backspace与Enter，但是其他一些键盘控制没有实现，up,down,left,right等等。

这部分准备下次实现。

功能号：02H和12H

功能：读取键盘标志   
入口参数：AH ＝02H—普通键盘的移位标志   
             ＝12H—扩展键盘的移位标志   
   
出口参数：AL＝键盘标志(02H和12H都有效)，  
          其各位之值为1时的含义如下：   
          位7—INS开状态  
          位3—ALT键按下   
          位6—CAPS LOCK开状态  
          位2—CTRL键按下   
          位5—NUM LOCK开状态  
          位1—左SHIFT键按下   
          位4—SCROLL LOCK开状态 位0—右SHIFT键按下

          AH＝扩展键盘的标志(12H有效)，  
          其各位之值为1时的含义如下：  
          位7—SysReq键按下  
          位3—右ALT键按下   
          位6—CAPS LOCK键按下  
          位2—右CTRL键按下   
          位5—NUM LOCK键按下  
          位1—左ALT键按下   
          位4—SCROLL键按下  
          位0—左CTRL键按下