**c语言综合研究七报告**

1. **研究过程展示**

**1>.程序1部分：**

获取字符使用什么函数呢？

getchar（）函数？scanf（）函数？getche（）函数？getch（）函数？

getchar有一个int型的返回值.当程序调用getchar时.程序就等着用户按键.用户输入的字符被存放在键盘缓冲区中.直到用户按回车为止(回车字符也放在缓冲区中).当用户键入回车之后,getchar才开始从stdio流中每次读入一个字符.getchar函数的返回值是用户输入的第一个字符的ASCII码,如出错返回-1,且将用户输入的字符回显到屏幕.如用户在按回车之前输入了不止一个字符,其他字符会保留在键盘缓存区中,等待后续getchar调用读取.也就是说,后续的getchar调用不会等待用户按键,而直接读取缓冲区中的字符,直到缓冲区中的字符读完为后,才等待用户按键.。

scanf（）函数会暂停程序等待输入，并且在输入完后还要按回车才能继续向下执行，所以不符合我们的需求。

getche（）函数不用敲回车就可以将字符输入。

getch（）函数不用敲回车，也不显示，只是将字符输入进行处理。

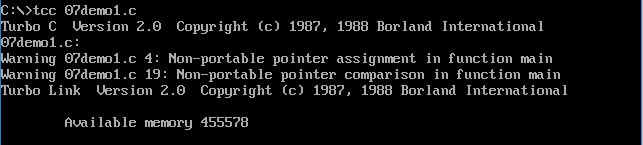
getch与getchar基本功能相同,差别是getch直接从键盘获取键值,不等待用户按回车,只要用户按一个键,getch就立刻返回, getch返回值是用户输入的ASCII码,出错返回-1.输入的字符不会回显在屏幕上.getch函数常用于程序调试中,在调试时,在关键位置显示有关的结果以待查看,然后用getch函数暂停程序运行,当按任意键后程序继续运行.

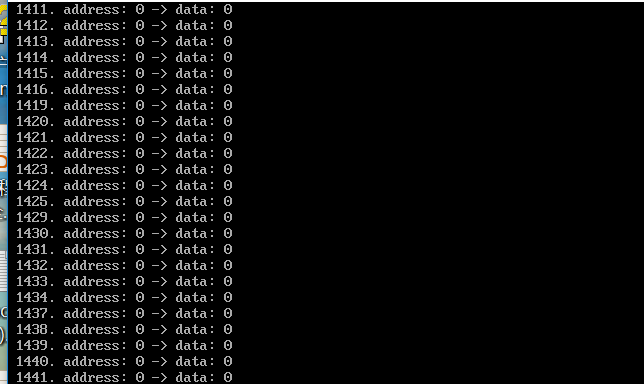
最后，我选择了getch（）函数。

但是只是这样的话，即使没有键盘输入的情况，也会出现阻塞现象。

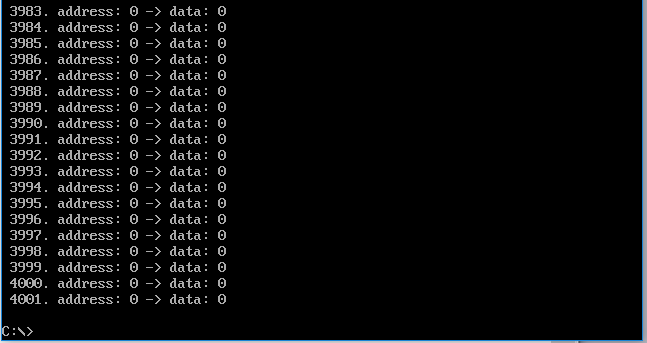
查找c语言库函数发现输入检测函数有kbhit（）函数，如果有输入它会返回一个非0值，没有就返回0

编译一下，出现两个警告，无错误。

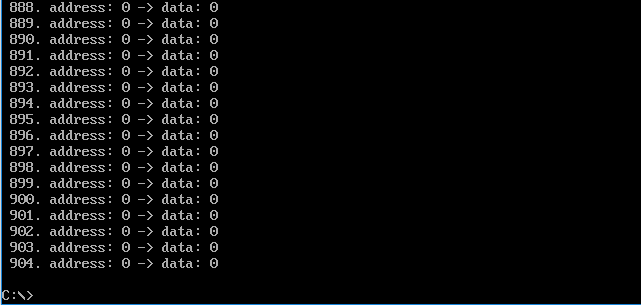




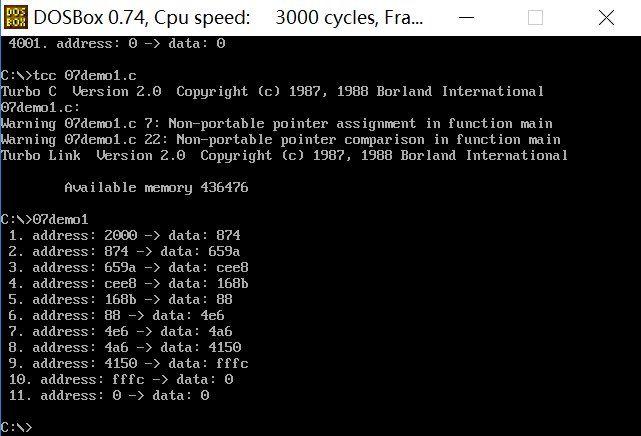
一直运行，直到 count =4001 会退出。



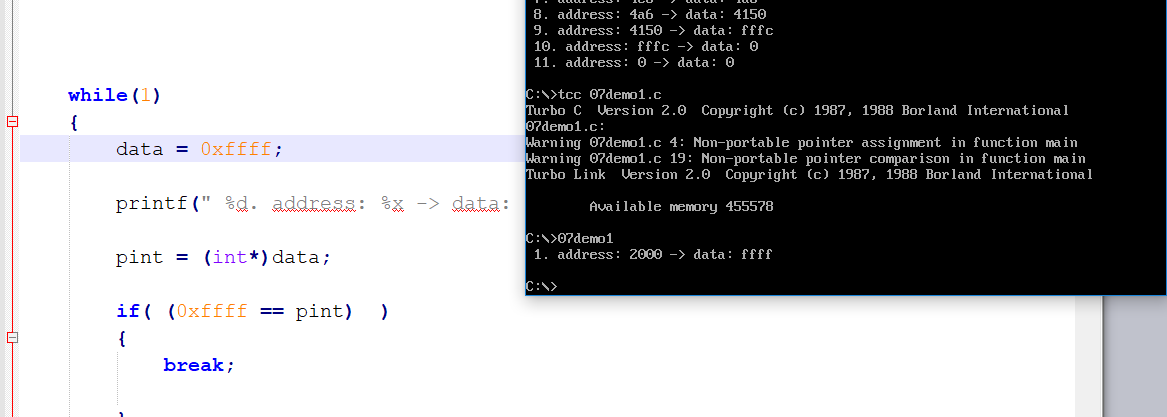
中途按下q：



将count 的值改为10：



将 “data = \*pint; ” 改为 “data = 0xffff;”,观察结果。



程序部分：

void main()

{

int\* pint = 0x2000;

int data = 0;

int count = 1;

while(1)

{

data = \*pint;

printf(" %d. address: %x -> data: %x\n",count,pint,data);

pint = (int\*)data;

if( (0xffff == pint) )

{

break;

}

if( kbhit() )

{

if('q' == getch())

{

break;

}

}

if(count > 4000)

{

return;

}

count++;

}

}

1. **已思考研究并已解决问题汇总**

遇到多个功能相似的函数，如何抉择？

如果数目不多的话，最好每个都亲自试一遍，看一下最后的效果。

1. **已思考研究并未解决问题汇总**
2. 实现自己的kbhit（）函数，之前学汇编时接触的键盘中断，或许可以尝试c语言实现。
3. 两个警告信息这里有没有关注的必要，虽然最后功能从表面上来看是实现了。
4. **研究感想（心得体会）**

最近在鼓捣stm32单片机，这又让人有的联想了 。流行的stm32单片机编程方法，主要有三种：库函数版、寄存器版、位运算版，但是归根结底，程序打交道的是最后的stm32单片机，是硬件，归根结底是在操控寄存器。寄存器就在那儿，不离不弃。所以这个研究中只是用了kbhit（）函数，这就像是用了库函数版，之前汇编键盘中断等就是寄存器版，当然c语言中也是可以直接操纵寄存器的。对于这个问题，自己还要好好理解。