实验三

**1.指针的理解**

\*(char \*)0x2000 =’a’;自己以前的理解为：将2000h 强行转换为char型指针地址，将字符送到指向2000h的指针所指向的单元中。现在理解：第一个\*表示要访问的是一个内存空间。“（char \*）”里面的“\*”指明了数值表示一个内存空间的地址。“Char” 指明了这个地址是储存char型数据的内存空间的地址。

**2.编写程序在屏幕中显示字符‘a’**

main()

{

\*(int far\*)0xb80007d0=0x0261;

}

**3.** c全局变量放在内存中，就是所谓的静态区。局部变量放在堆栈中，如果局部变量多次使用，会把局部变量放在 si,di 寄存器中。对于push bp,mov bp,sp的含义，因为局部变量的访问要用bp来寻址，就要改变bp的值，所以要push bp. 之后又sub sp,06h这是为局部变量开辟的空间，这样就可以在主程序push pop 不会覆盖局部变量区域。最后sp 和bp恢复到进入子程序的值，这样就会正确返回。因为main函数其实是子程序，调用的时候就把main程序的偏移地址压入堆栈。

**4.c语言将函数的返回值存放在哪里？**

经过研究得以下结论：char型的返回值放在AL中，int型的返回值放在AX中。Long int型返回值在AX,DX中,double 尚未研究明白

5.解析程序

#define buffer ((char \*)\*(int far \*)0x200) ；定义buffer为char型地址，地址为0000:0200地址处所放的地址

main()

{

buffer=(char \*)malloc(20);//在内存空间找连续20字节单元为空的内存空间，把首地址放到

// AX中在把AX放到0000:0200处 经debug调试发现0000:0200处放的偏移地址为0298h

// 即申请的内存空间为DS:0298H 开始的20个字节单元

buffer[10]=0; //令ds:0298h+a 处的单元赋值为0

while(buffer[10]!=8) //判断ds:0298h+a 处的单元值是否为8 不是就继续执行，是就跳转

{

buffer[buffer[10]]='a'+buffer[10];

buffer[10]++;//经过8次得到 ds:0298h 开始单元一次存放’a’, ’b’ , ’c’ , ’d’ , ’e’ , ’f’ , ’g’ , ’h’。

}

free(buffer); 释放申请的内存

}