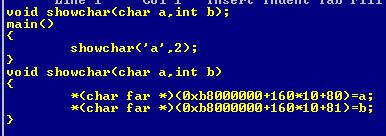
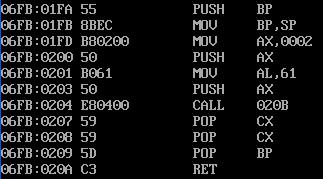
# 一、 学习过程

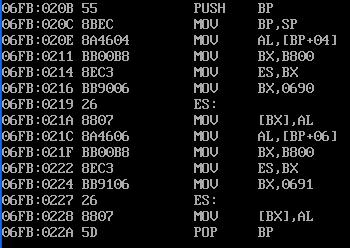
编写程序如下：



编译连接并用debug加载，观察main函数的内容：



Showchar函数的内容：



IMG_259

观察发现，main函数要传递两个参数‘a’和2，在汇编代码中是先将2赋给ax，再将ax入栈，然后将a赋给ax,将ax入栈。在showchar函数中，程序将sp赋给bp，再从bp+4处取出形参a赋给al，再将al中的值赋给b800：690h，然后再从bp+6处取出形参b赋给al，再将al中的值赋给b800：691h。可见main函数给showchar传递参数是把要传递的值赋给ax，再将ax入栈，且如果有多个要传递的值，是由后往前将参数入栈。Showchar函数接收参数是将sp赋给bp，然后由bp+4找到栈中存储的参数a，由bp+6找到栈中存储的的参数b，为什么是bp+4和bp+6呢？因为程序在将两个参数入栈后，call指令将showchar的地址入栈占2个字节，在showchar中将bp入栈又占2个字节，所以要由bp+4找到第一个参数的地址。那么我对此提出三个问题：

（1） main函数将char型数据和int型数据入栈是占2个字节，那么如果是float型或者long int型、double型、long double型等超过2字节的变量类型怎么办？

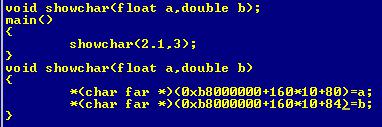
（2） Showchar函数将栈中取出的参数赋给al，为什么2是int型也只赋给一个字节的al？如果是更大的参数怎么办？

（3） 我们注意到这一个指令

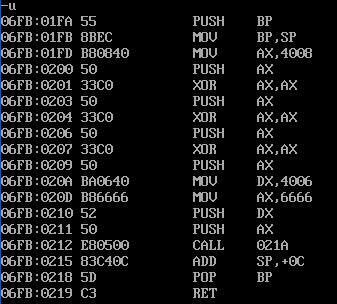
IMG_260

  是把al赋给es：[bx]，是不是所有非ds的段寄存器都要像这样写？

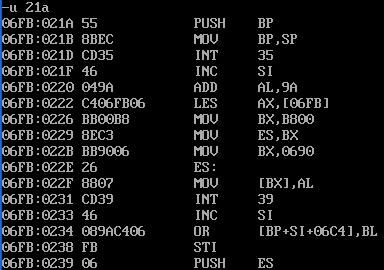
对于第一个问题，我们把程序中的char和int改成float和double看看：

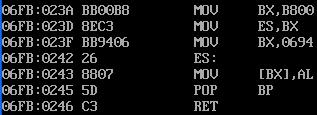


编译连接，用debug查看，main函数为：



Showchar函数为：





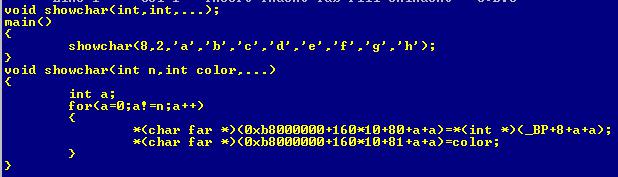
发现main函数的入栈值为：4008、0000、0000、0000、4006、6666.

再看showchar函数的内容，查资料发现int 35h的作用是读取中断向量，但是不知道它的具体功能，inc si和les ax，[06fb]；int 39的作用是什么呢？

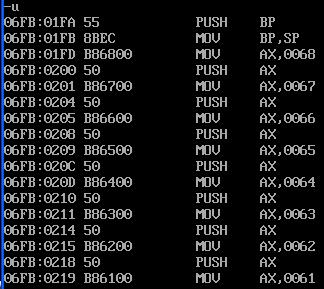
这里我对于c语言的一些语句在汇编里的实现还是有的不理解，但是这不是我们研究的重点，既然暂时弄不懂，就先跳过这个问题。

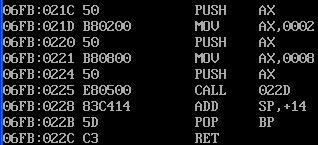
再看第三个问题，发现所有es作段地址的指令都是如上格式，ds作段寄存器的指令都把ds省略了。

再来看下一个程序：

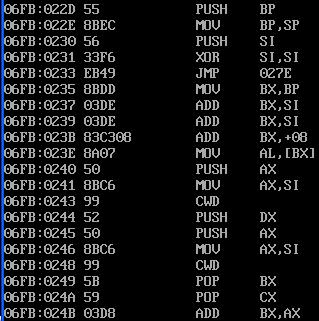


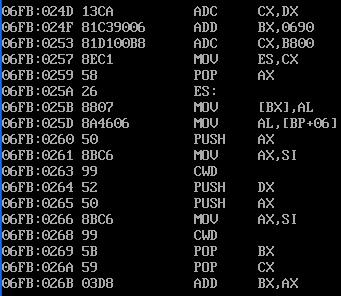
编译连接，用debug加载查看，main函数为：

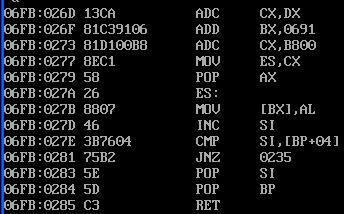




Showchar函数为：





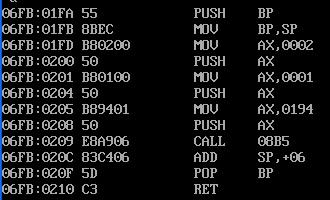


观察C语言的showchar函数可以发现：第一个参数n是要显示的参数数量，第二个参数color是要显示的参数颜色，之后的就是要显示的参数。Showchar函数通过参数n来知道要显示多少个字符。然后通过循环来调用寄存器从栈中提取参数。

但是printf函数的参数是要直接输出的，没有一个参数是告诉它下面有多少个参数。但是printf里面是要输入%c或者%d，那么函数是通过统计%c和%d的数量来判断要输出多少参数的吗？我们写一个printf函数来看看：

IMG_271

编译连接并用debug加载有：

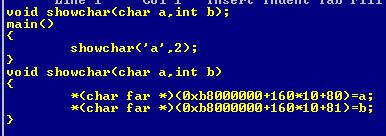


这里是将参数1和2入栈，再入栈194，然后执行printf函数，那么194有什么作用呢？查阅资料知，程序将%c和%d等符号放在偏移地址0194处，结尾加0，通过统计该地址处的%个数来确定要输出的字符数量。所以peintf函数和showchar函数的区别就是showchar函数参数个数已给出而printf函数是要根据%c或%d个数来确定参数个数而已。那么我们要实现简单的printf函数，可以在showchar函数的基础上来改动。

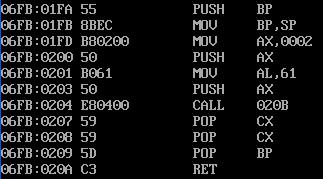
下面是网上找的代码：

# 一、 学习过程

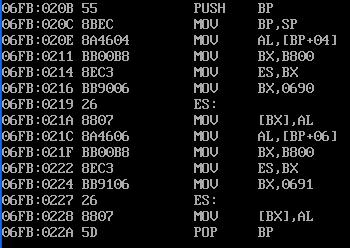
编写程序如下：



编译连接并用debug加载，观察main函数的内容：



Showchar函数的内容：



IMG_259

观察发现，main函数要传递两个参数‘a’和2，在汇编代码中是先将2赋给ax，再将ax入栈，然后将a赋给ax,将ax入栈。在showchar函数中，程序将sp赋给bp，再从bp+4处取出形参a赋给al，再将al中的值赋给b800：690h，然后再从bp+6处取出形参b赋给al，再将al中的值赋给b800：691h。可见main函数给showchar传递参数是把要传递的值赋给ax，再将ax入栈，且如果有多个要传递的值，是由后往前将参数入栈。Showchar函数接收参数是将sp赋给bp，然后由bp+4找到栈中存储的参数a，由bp+6找到栈中存储的的参数b，为什么是bp+4和bp+6呢？因为程序在将两个参数入栈后，call指令将showchar的地址入栈占2个字节，在showchar中将bp入栈又占2个字节，所以要由bp+4找到第一个参数的地址。那么我对此提出三个问题：

（1） main函数将char型数据和int型数据入栈是占2个字节，那么如果是float型或者long int型、double型、long double型等超过2字节的变量类型怎么办？

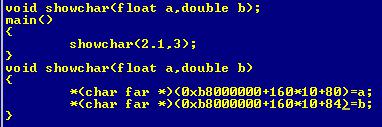
（2） Showchar函数将栈中取出的参数赋给al，为什么2是int型也只赋给一个字节的al？如果是更大的参数怎么办？

（3） 我们注意到这一个指令

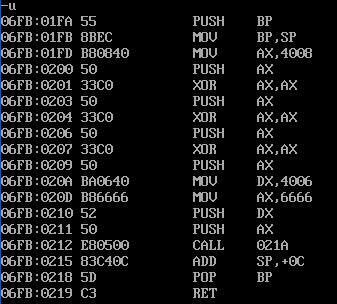
IMG_260

  是把al赋给es：[bx]，是不是所有非ds的段寄存器都要像这样写？

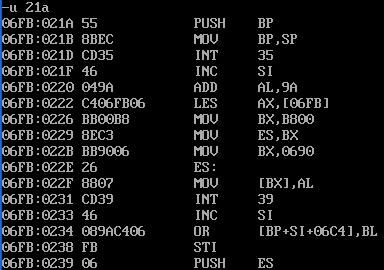
对于第一个问题，我们把程序中的char和int改成float和double看看：

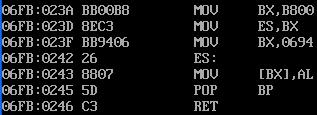


编译连接，用debug查看，main函数为：



Showchar函数为：





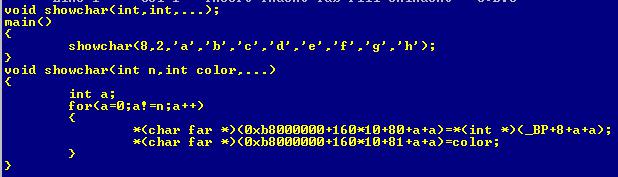
发现main函数的入栈值为：4008、0000、0000、0000、4006、6666.

再看showchar函数的内容，查资料发现int 35h的作用是读取中断向量，但是不知道它的具体功能，inc si和les ax，[06fb]；int 39的作用是什么呢？

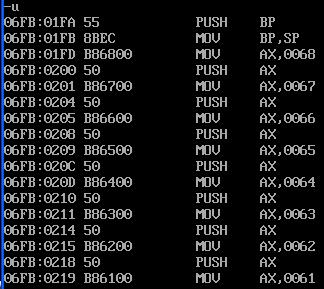
这里我对于c语言的一些语句在汇编里的实现还是有的不理解，但是这不是我们研究的重点，既然暂时弄不懂，就先跳过这个问题。

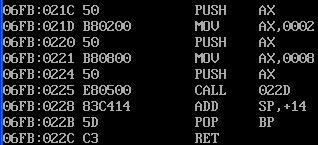
再看第三个问题，发现所有es作段地址的指令都是如上格式，ds作段寄存器的指令都把ds省略了。

再来看下一个程序：

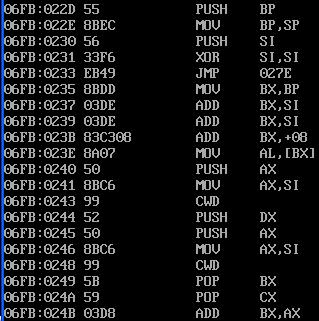


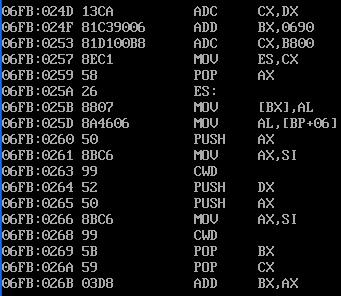
编译连接，用debug加载查看，main函数为：

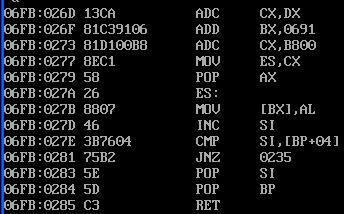




Showchar函数为：





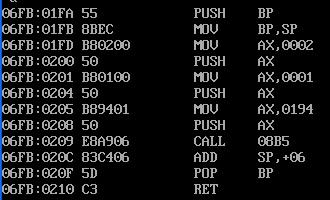


观察C语言的showchar函数可以发现：第一个参数n是要显示的参数数量，第二个参数color是要显示的参数颜色，之后的就是要显示的参数。Showchar函数通过参数n来知道要显示多少个字符。然后通过循环来调用寄存器从栈中提取参数。

但是printf函数的参数是要直接输出的，没有一个参数是告诉它下面有多少个参数。但是printf里面是要输入%c或者%d，那么函数是通过统计%c和%d的数量来判断要输出多少参数的吗？我们写一个printf函数来看看：

IMG_271

编译连接并用debug加载有：



这里是将参数1和2入栈，再入栈194，然后执行printf函数，那么194有什么作用呢？查阅资料知，程序将%c和%d等符号放在偏移地址0194处，结尾加0，通过统计该地址处的%个数来确定要输出的字符数量。所以peintf函数和showchar函数的区别就是showchar函数参数个数已给出而printf函数是要根据%c或%d个数来确定参数个数而已。那么我们要实现简单的printf函数，可以在showchar函数的基础上来改动。

下面是网上找的代码：

void printf(char \*,...);

int pow(int, int);

main()

{

/\*printf("I think this is interesting :%c and %c and %c",0x61,0x62,0x63);\*/

printf("No.%d,%d,%d,this is me %c ing yuan",45,123,8958,'Q');

}

void printf(char \*des, ...)

{

/\*first sure the length of string des\*/

int len=0;

int i=0;

int showp=0; /\*define the point of showing chars\*/

int parap=0; /\*define of parameter position in stack\*/

int intValueLength=1;

int signalNum=0;

/\*calculate length of stirng des \*/

while(des[i]!='/0')

{

len++;

i++;

}

i=0;

while(des[i]!='/0')

{

if(des[i]=='%')

{

/\*check type of value user want to show.\*/

if(des[i+1]=='d')

{

/\*here show integer value\*/

int showIntValue=\*(int \*)(\_BP+6+parap+parap); /\*here, we show understand that define one char point value, we just push the point value into stack, but not the string value\*/

int reValue=showIntValue;

/\* \*(int far \*)(0xb8000000+160\*10+80+showp+showp)=showIntValue;

\*(int far \*)(0xb8000000+160\*10+81+showp+showp)=2;\*/

i+=2;

parap++;

intValueLength=1;

/\*here we calculate the length of integer value we want to show ,and then we can sure the next value show position\*/

while(reValue/10!=0)

{

intValueLength++;

reValue/=10;

}

/\*first calculate the length of unmber and show every sigal positon number of Integer \*/

signalNum = showIntValue/pow(10,--intValueLength);

\*(char far \*)(0xb8000000+160\*10+80+showp+showp)=signalNum+48; /\*show the highest signal number\*/

\*(char far \*)(0xb8000000+160\*10+81+showp+showp)=2;

showp++;

while(intValueLength!=0)

{

showIntValue=showIntValue-signalNum\*pow(10,intValueLength);

signalNum= showIntValue/pow(10,--intValueLength);

\*(char far \*)(0xb8000000+160\*10+80+showp+showp)=signalNum+48; /\*show the highest signal number\*/

\*(char far \*)(0xb8000000+160\*10+81+showp+showp)=2;

showp++;

}

/\*showp+=intValueLength;\*/

}

else if (des[i+1]=='c')

{

/\*here show charactor value\*/

\*(char far\*)(0xb8000000+160\*10+80+showp+showp)=\*(int \*)(\_BP+6+parap+parap); /\*value of \_BP and distance address of CALL order\*/

\*(char far\*)(0xb8000000+160\*10+81+showp+showp)=2;

parap++;

showp++;

i+=2;

}

else /\*direct show char value in string des\*/

{

\*(char far \*)(0xb8000000+160\*10+80+showp+showp)=\*(int \*)(\*(int \*)(\_BP+4)+i);

i++;

showp++;

}

}

else /\*also direct to show char in des\*/

{

\*(char far \*)(0xb8000000+160\*10+80+showp+showp)=\*(int \*)(\*(int \*)(\_BP+4)+i);

i++;

showp++;

}

}

}

int pow(int index,int power)

{

int finalValue=1;

if(power==0);

else

{

while(power!=0)

{

finalValue=finalValue\*index;

power--;

}

}

return finalValue;

}

二、 解决的问题

（1） 使用es+偏移地址时，查看指令，段寄存器会独自占一条指令。

（2） Main函数是如何给showchar传递参数的？showchar是如何接受参数的？

答：main函数将参数入栈，showchar用bp寄存器在栈中提取参数。

（3） showchar函数是如何知道要显示多少个字符的？printf是如何知道有多少个参数的？

答：showchar函数是通过第一个参数n知道要显示字符数量的，printf是通过第一个字符串中%c和%d的数量来知道要显示字符数量的。

三、 未解决的问题

（1） main函数将char型数据和int型数据入栈是占2个字节，那么如果是float型或者long int型、double型、long double型等超过2字节的变量类型怎么办？

（2） Showchar函数将栈中取出的参数赋给al，为什么2是int型也只赋给一个字节的al？如果是更大的参数怎么办？

二、 解决的问题

（1） 使用es+偏移地址时，查看指令，段寄存器会独自占一条指令。

（2） Main函数是如何给showchar传递参数的？showchar是如何接受参数的？

答：main函数将参数入栈，showchar用bp寄存器在栈中提取参数。

（3） showchar函数是如何知道要显示多少个字符的？printf是如何知道有多少个参数的？

答：showchar函数是通过第一个参数n知道要显示字符数量的，printf是通过第一个字符串中%c和%d的数量来知道要显示字符数量的。

三、 未解决的问题

（1） main函数将char型数据和int型数据入栈是占2个字节，那么如果是float型或者long int型、double型、long double型等超过2字节的变量类型怎么办？

（2） Showchar函数将栈中取出的参数赋给al，为什么2是int型也只赋给一个字节的al？如果是更大的参数怎么办？