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

char \*p1="this is a c program.";

char \*p2="welcome to c!";

char \*p3="hello world!";

char \*p4="wo shi shao wen cai!";

char \*p5="liu ning shi sha bi!";

char \*p6="zhang sheng shi dou bi!";

char \*p7="test success!";

main()

{

showstr(p1,p2,p3,p4,p5,p6,p7,0);

}

/\* 函数实现原理

buffer存放p1,通过buffer[]打印数据

然后p2传给buffer,通过buffer[]打印数据

直到buffer==0时结束 \*/

void showstr(char \*buffer,...)

{

unsigned int i,j,k;

i=0;/\* 在栈中找各字符串的首地址 \*/

j=0;/\* 把字符串一个一个打印 \*/

k=0;/\* 换行作用 \*/

for(i=2;buffer;i=i+2)

{

for(j=0;buffer[j];j++)

{

\*(char far\*)(0xb8000000+160\*10+80+j\*2+k)=buffer[j];

}

k=k+160;

buffer=(char \*)(\*(int \*)(\_BP+4+i));

}

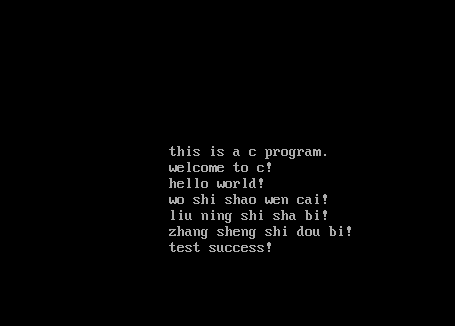
}

/\* for(;;)只需要两个分号，我之前是这样for(i=0;i<10;i++;)因为多加一个分号导致我的错误，搞了好久。

还有子函数 showstr(char \*buffer,...) 一定要有一个形参，如果是这样 showstr(char \*,...)和

showstr(...)则会报错 \*/

打印结果：



对于声明的研究

子函数和主函数在同一文件下，如果子函数未声明则编译器会自己创建一个返回值为int型，形参为不定型的函数声明。 如果你的函数定义返回类型为int型 则会编译通过，否则就会发生错误。函数前面没有类型是默认为int型。

当子函数与主函数不在同一文件下，如果没有声明是不会发生错误的因为链接过程只是根据函数名链接的，但是没有声明编译环境还是会创建一个默认的函数声明。主函数调用子函数 实参压栈是根据实参的类型压栈，而函数返回时会按照自己的类型返回 ，返回后取值是按声明的类型取值给相应变量。比如你定义的函数为

long f(long) 但是会发现 主函数调用 a=f(1)是将1按两个字节压入，然后f函数里参数是按long型在运算，返回long型数据，但是到主函数里就只取低两个字节给变量a。

综上 声明是必须的，如果你传递的参数不全是int型 就很有可能出错

如果你有返回值且不是int型 也很可能出错。

如果你的函数既没有形参 也没有返回类型 那么就不需要声明。正确使用是不会出错的。