**实验二：8086汇编与MIPS汇编字符串输出**

**一、实验目的**

1.使用8086汇编输出字符串。

2.使用MIPS汇编输出字符串。

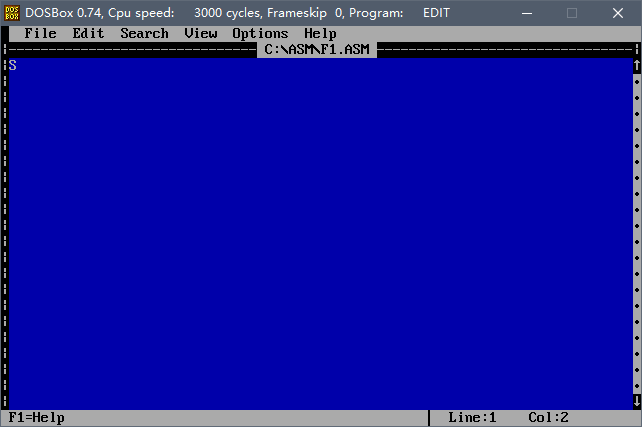
**二、实验过程**

**1.8086汇编**

首先打开DOSBox。进入后输入：

EDIT f1.asm

进入蓝色界面：



输入以下代码：

data segment

string db 'Wang Junya, 22307049', 0dh, 0ah, '$'

data ends

code segment

assume cs:code, ds:data

begin

    MOV AX, data

    MOV DS, AX

    LEA DX, string

    MOV AH, 9

    INT 21H

    MOV AH, 4CH

    INT 21H

code ends

    end begin

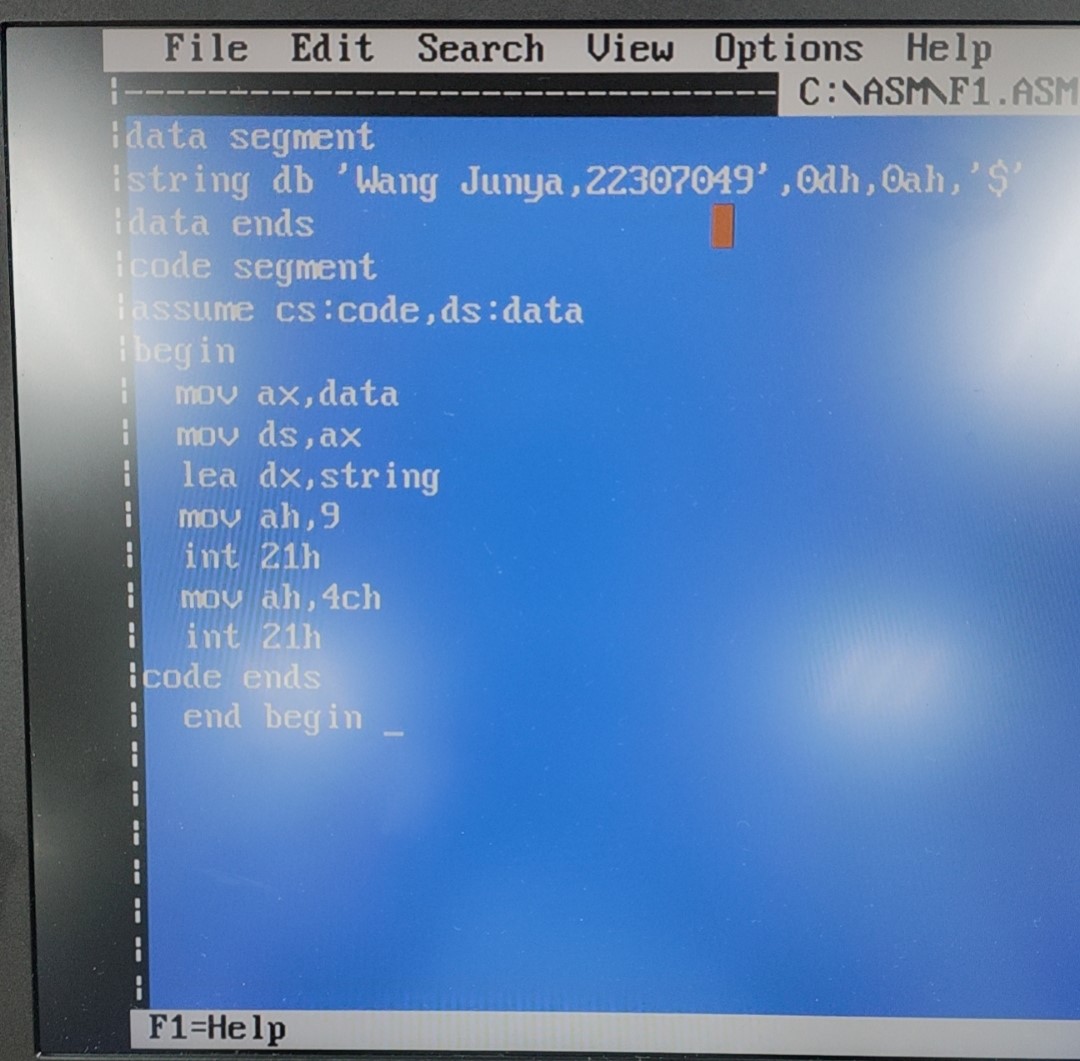
在这段程序中，data段中定义了string的内容。其中0dh, 0ah共同组成了换行的功能，’$’标志了字符串的结束。

在code段中，data段的地址首先送入AX再过渡到DS中。将string的地址传入DX。

MOV Ah, 9; INT 21H是调用字符串输出功能，出现一次即输出一次字符串。

MOV Ah, 4CH; INT 21H返回到操作系统，退出程序。

输入效果如下：



之后File-Save，退出蓝色界面。

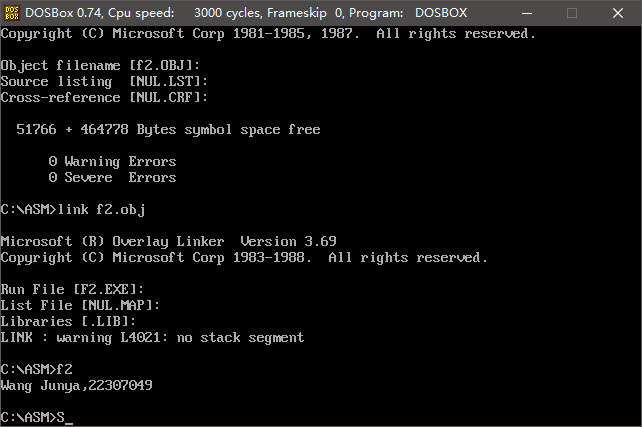
之后依次输入以下代码：

MASM f1.asm

LINK f1.obj

f1

其中，第一行是对f1.asm进行汇编生成.obj文件。第二行进行链接，生成可执行文件.exe。最后一行即是运行f1.exe。可以看到输出了string中的内容，即我的姓名与学号：



**2.MIPS汇编**

首先打开Mars。在这里我将文件命名为namenum。输入以下代码：

#namenum.asm

.data

out\_string:.asciiz"\nWang Junya, 22307049\n\n"

.text

main:

li $v0, 4

la $a0, out\_string

sysca11

li $v0, 4

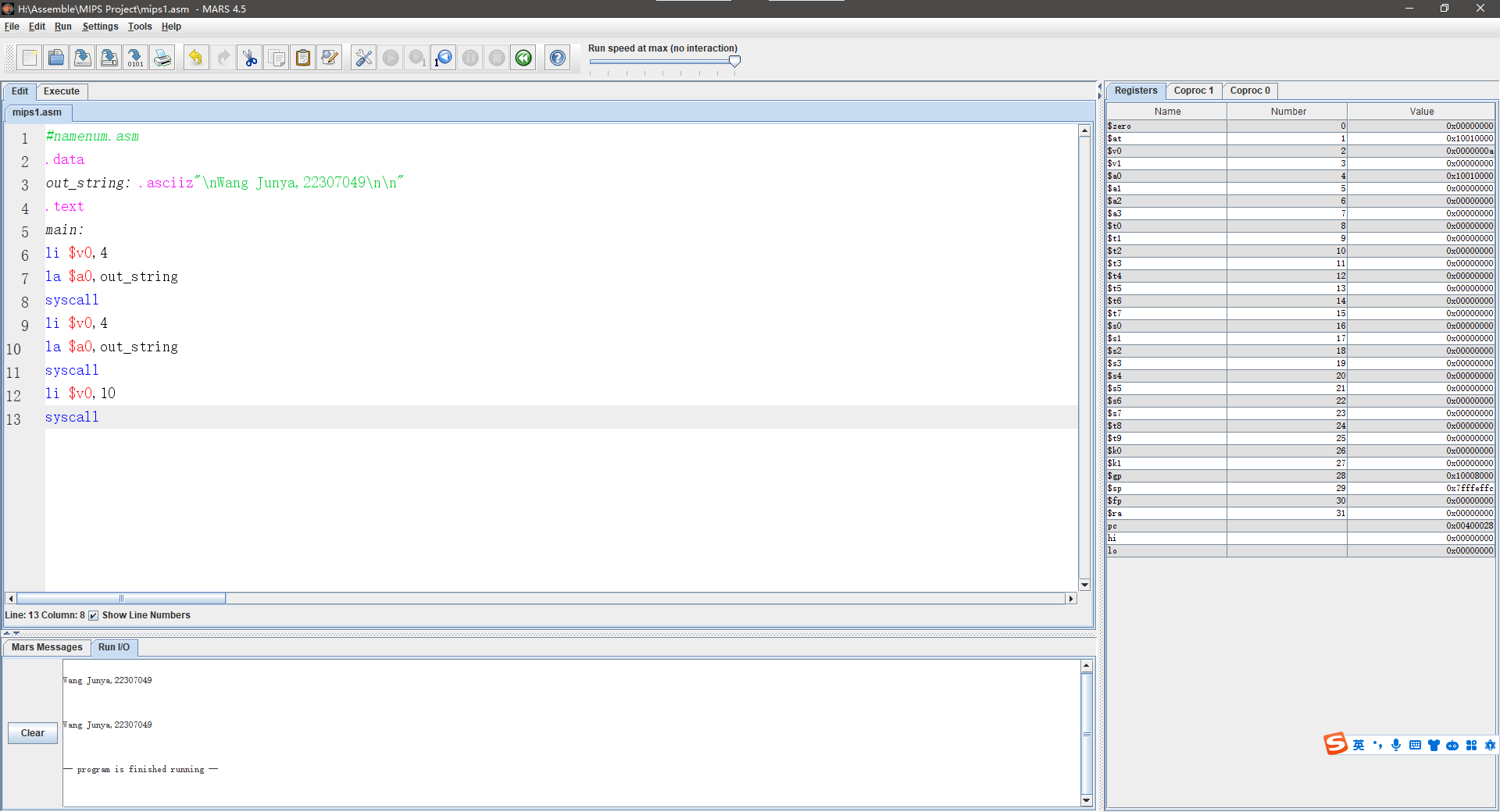
la $a0, out\_string

syscal1

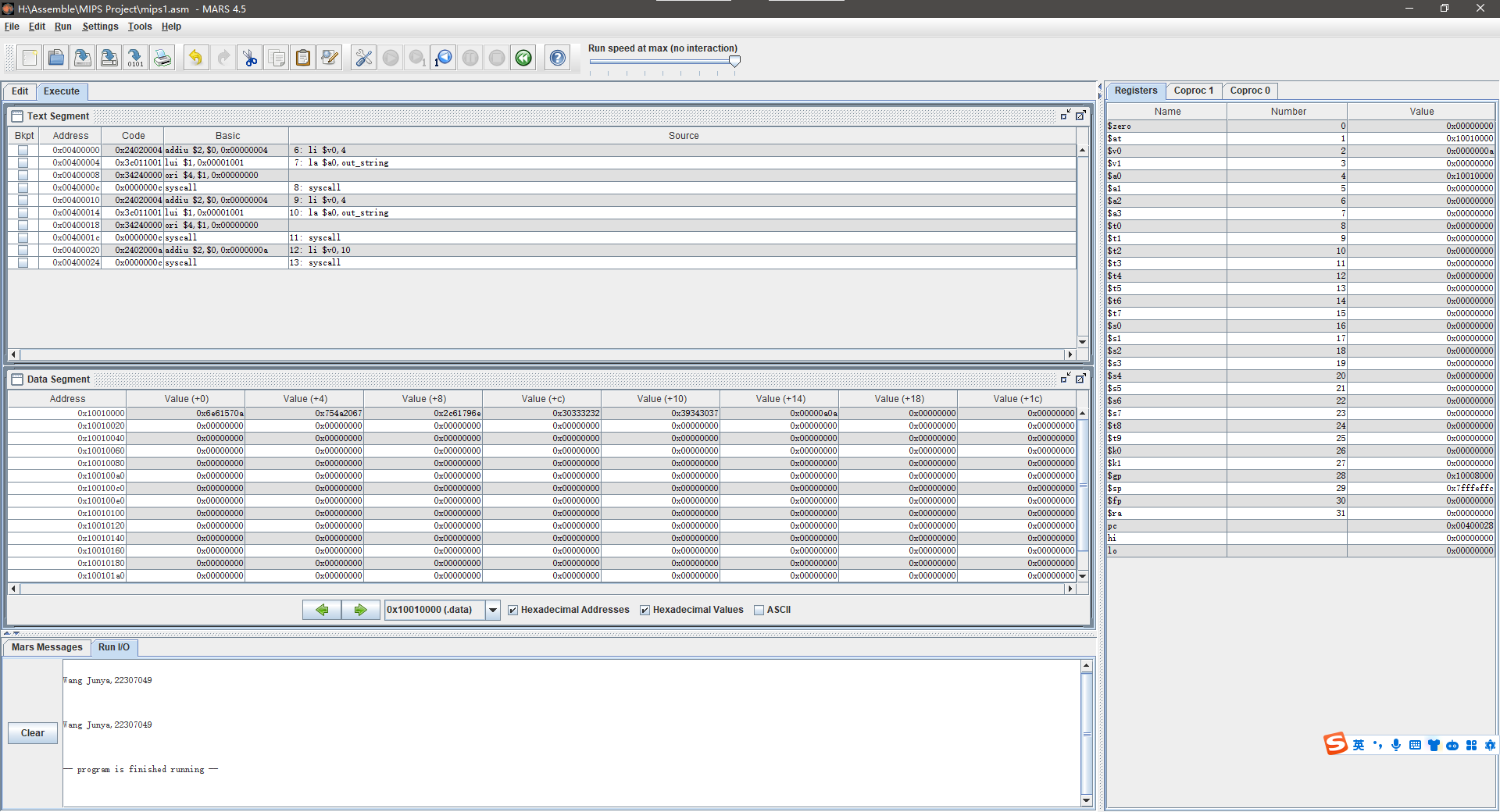
li $v0, 10

syscall

效果如图：



在Run之后，可以看到输出框出现了两次out\_string。



这张图中可以看到实际被翻译出来的语句。