**MIPS:**

基础知识：



**1byte = 8bit、1word = 32bit、1half-word = 16 bit**

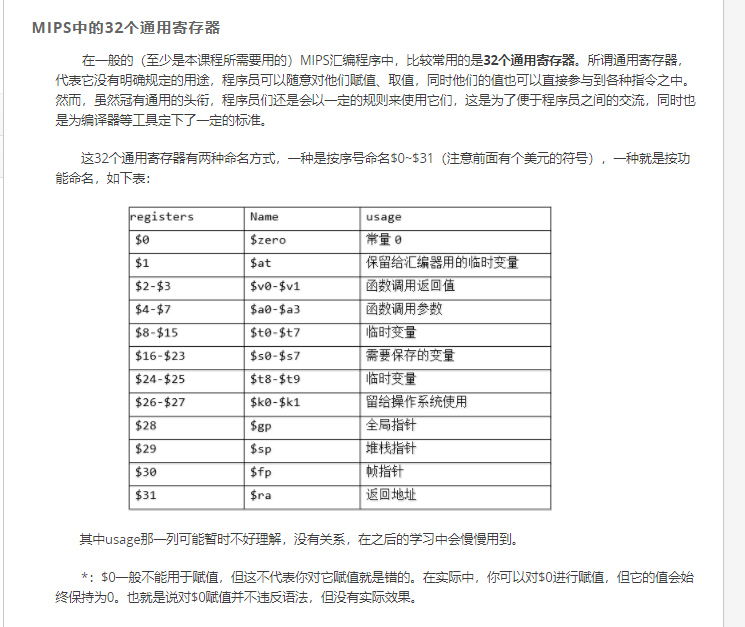


**右边是小端，在低地址**

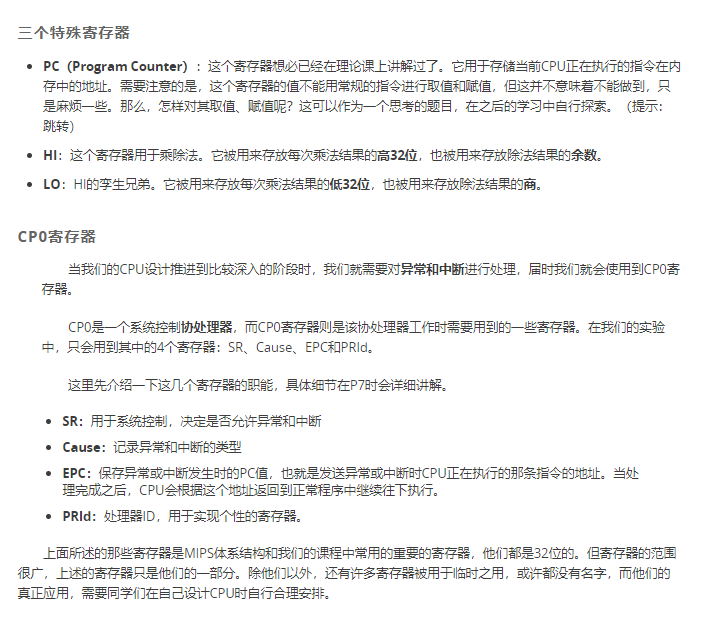
寄存器：



通用寄存器：

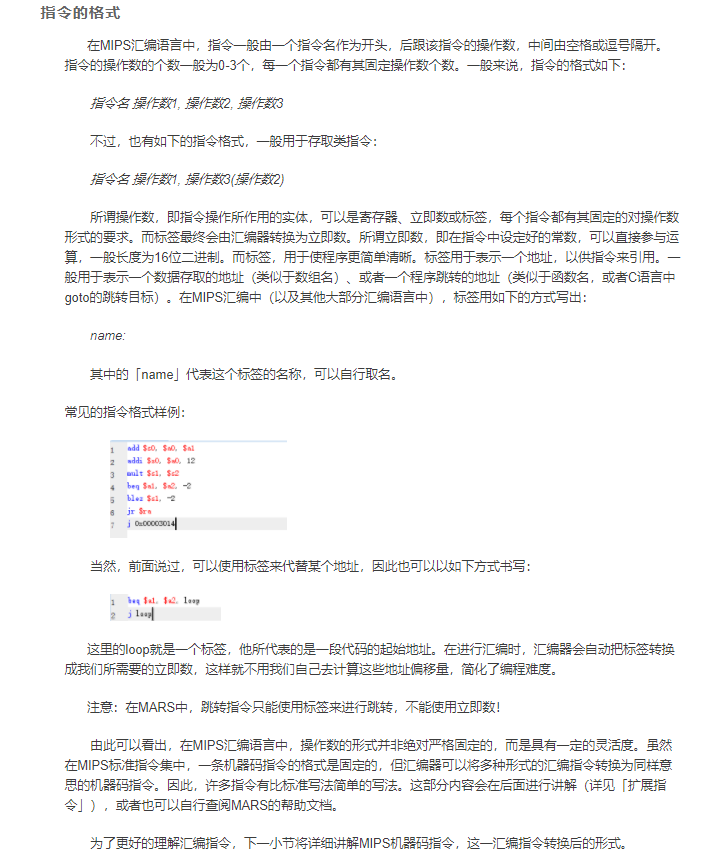


**对$0的赋值没有实际效果**



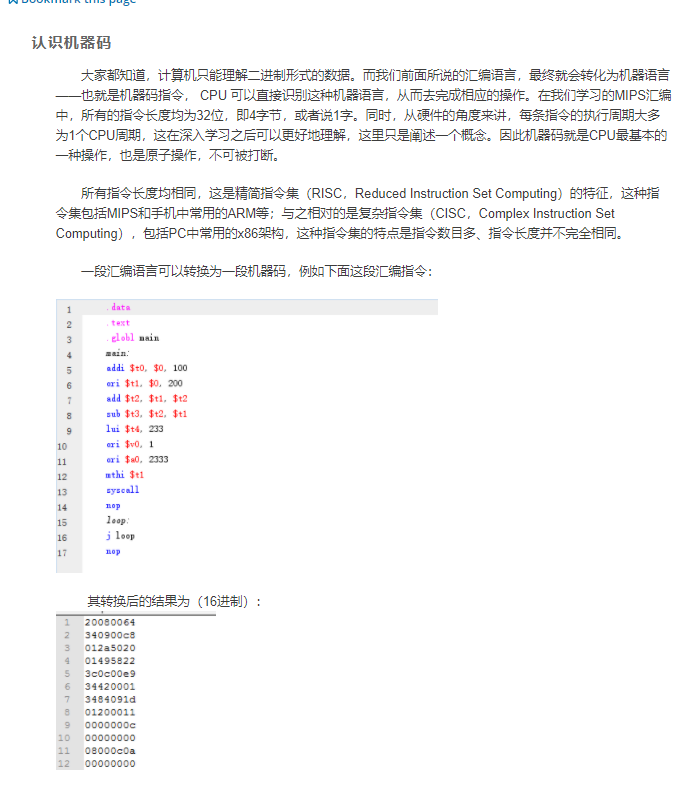
**CP0寄存器，异常和中断！！！！**





**指令的操作数一般为0-3个，操作数可以是寄存器、立即数或标签。**

**标签最终会由汇编器转换成立即数。**



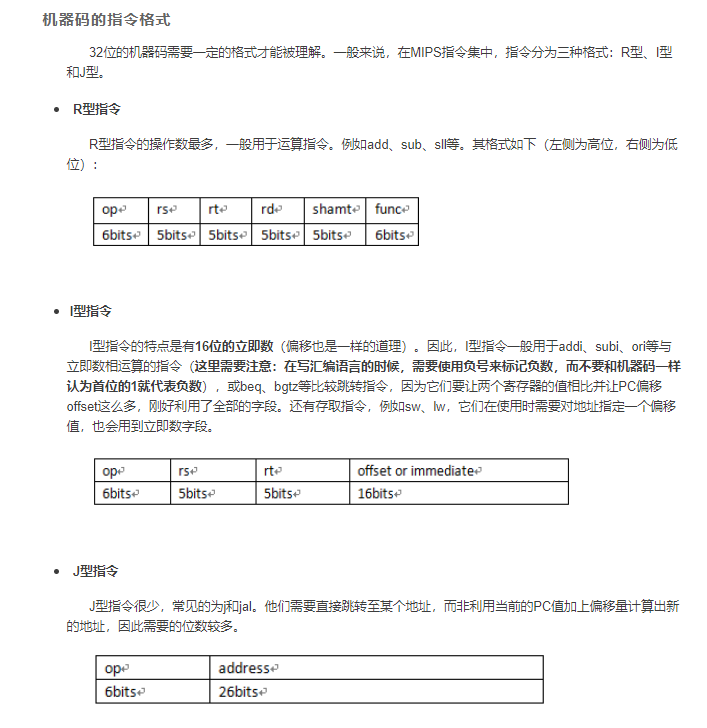
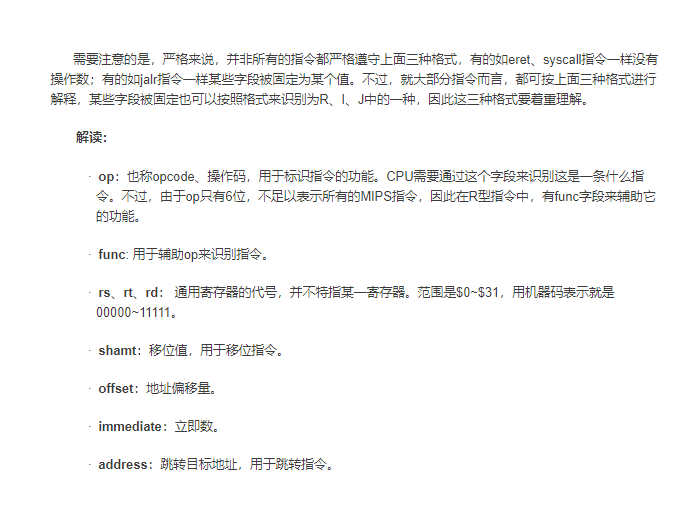
**RISC,Reduced Instruction Set Computing精简指令集**

**所有指令长度相同**

**CISC，Complex Instruction Set Computing 复杂指令集**

**指令数目多、指令长度不完全相同**

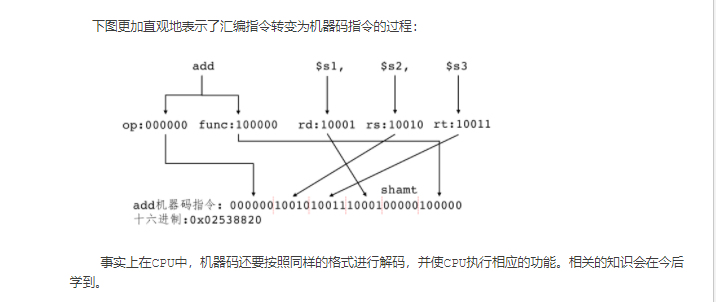
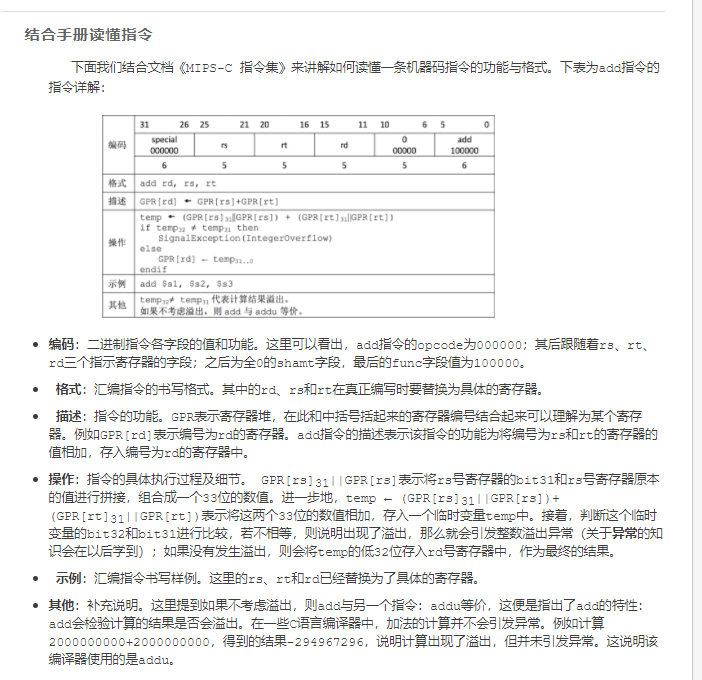
三种指令：



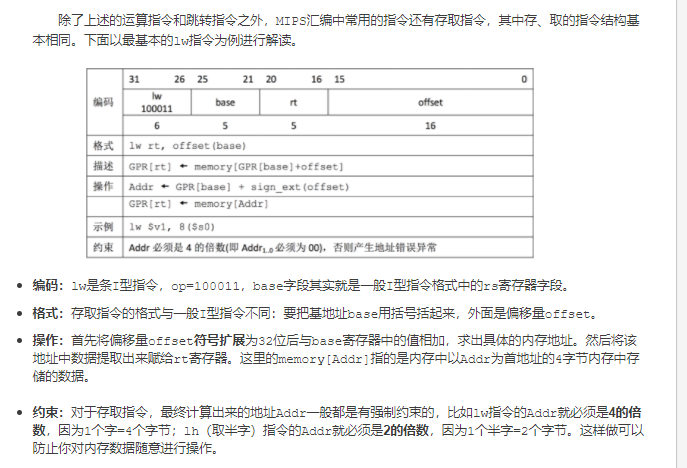
**I型指令有16位的立即数，注意用-表示负数。**

**并非所有指令都有操作数，比如syscall**

使用MARS手册：





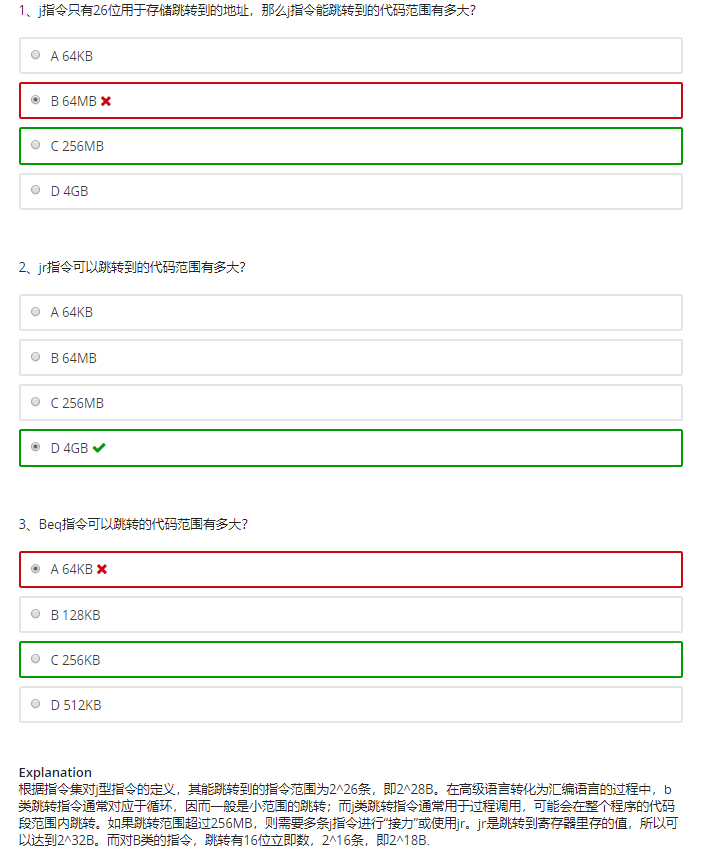


**mips指令都是32位，beq指令中低16位是地址offset（偏量）。 但是要将其转换成32位的地址就必须补长。 因为这个offset有负有正，所以正要补零，负要补一。所以过程是 offset先左移两位（因为一个指令32位占4个地址）然后符号位扩展， 就是真正的32位offset， 如果是负就往回跳，如果是正就往后跳。**

**offset应该是理解成指令数的偏移量。**

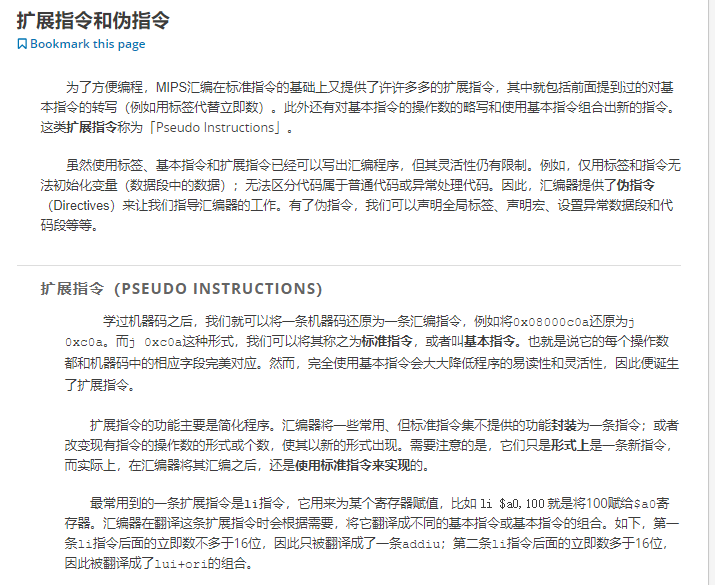




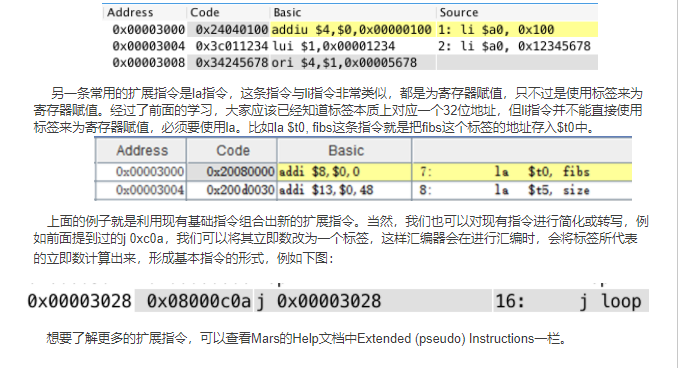


**注意offset是指指令数，代码范围还要乘4**

扩展指令和伪指令：



**扩展指令是一些常用的标准指令的封装**

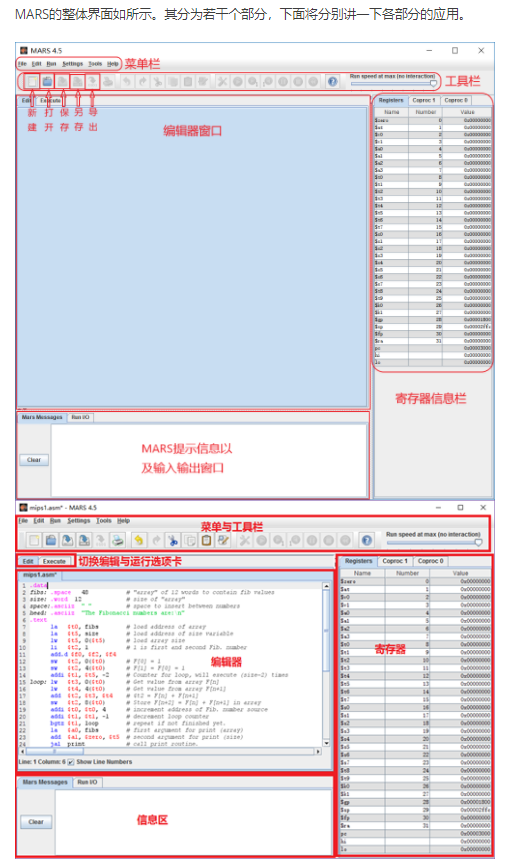




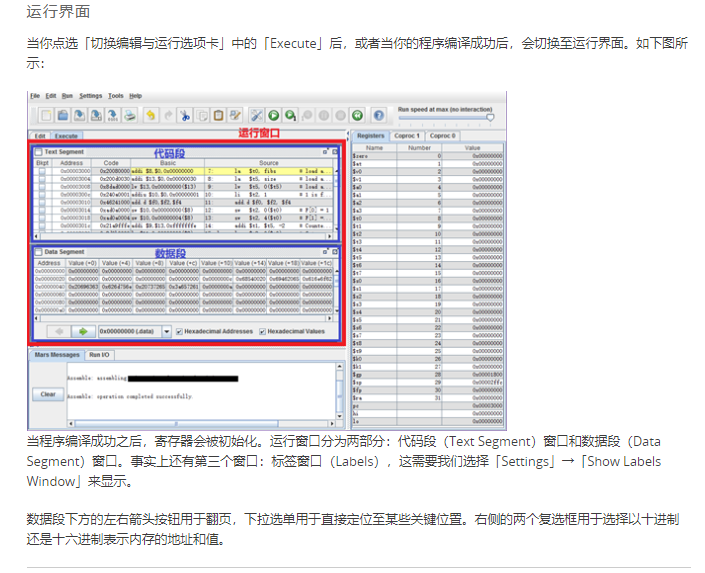


MARS的使用：

MARS的界面：







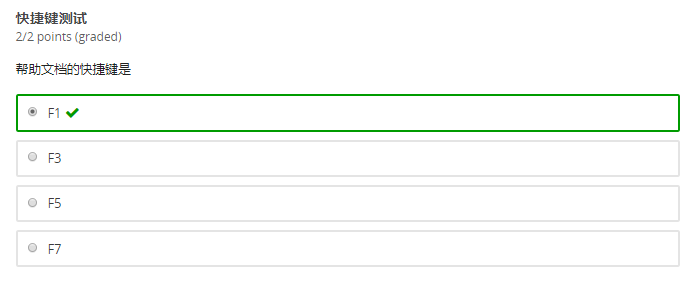




* **延迟槽: 菜单栏 Settings->Delayed branching P5后会使用，在学习延迟槽的具体原理前请勿勾选**
* **MARS->Command: 命令行操作规范(会在P7中使用)**

代码导出：

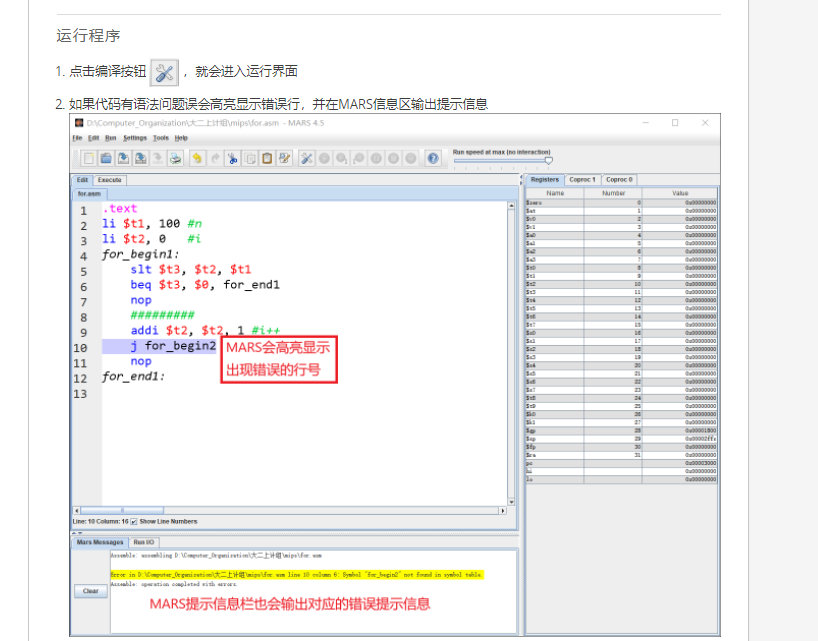


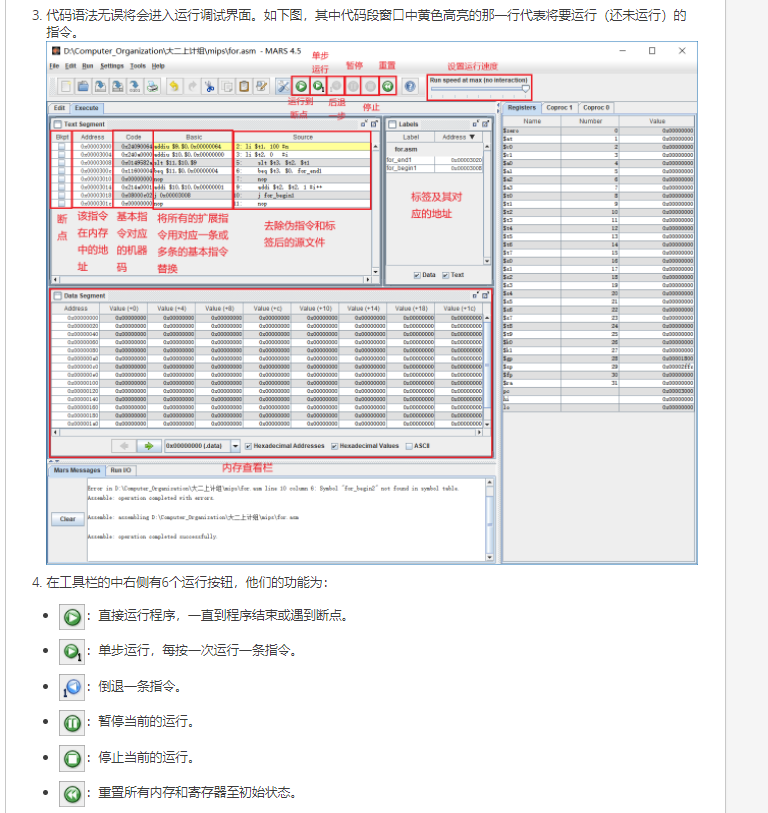


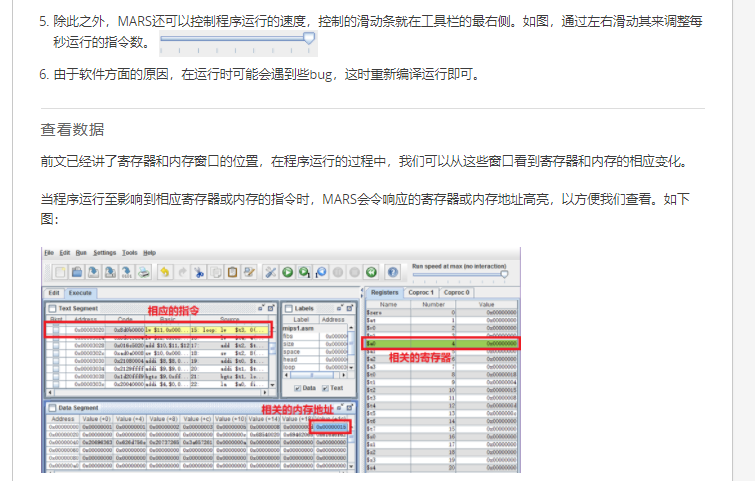




运行：

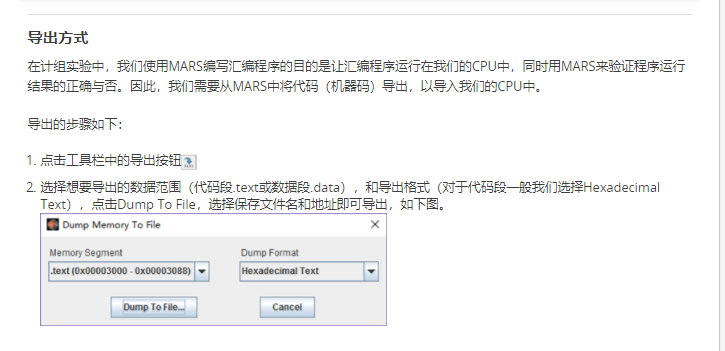


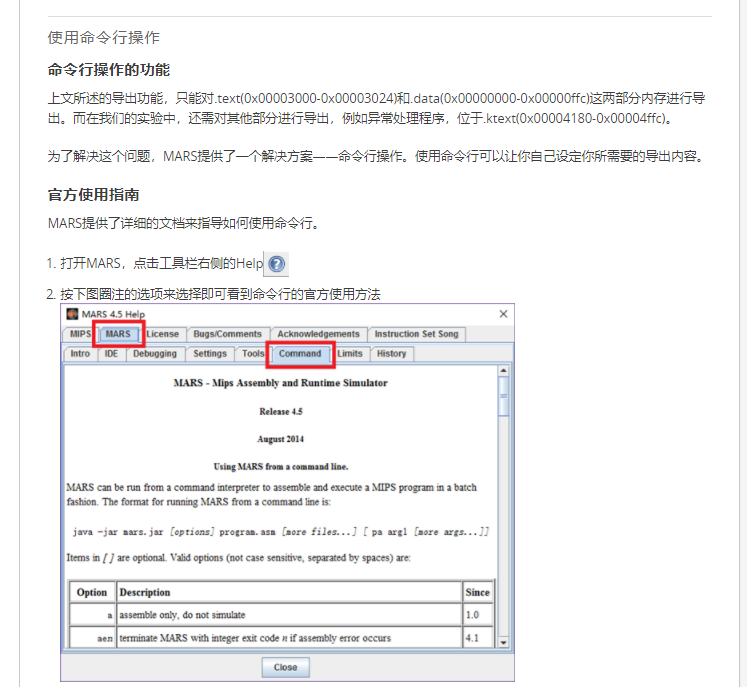


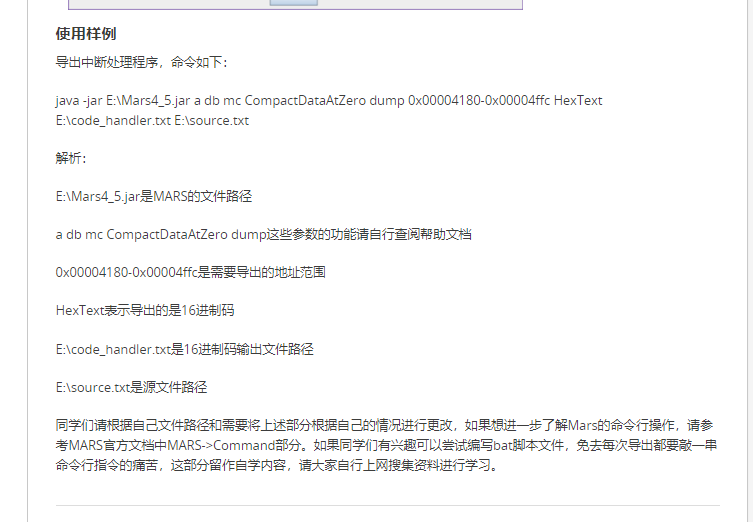




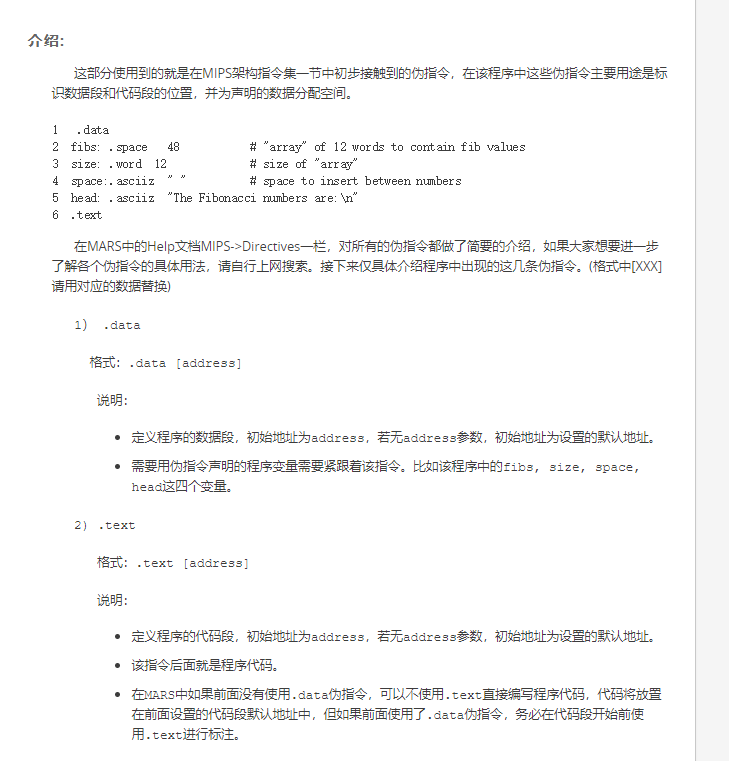
导出数据:



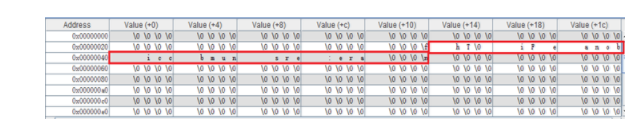
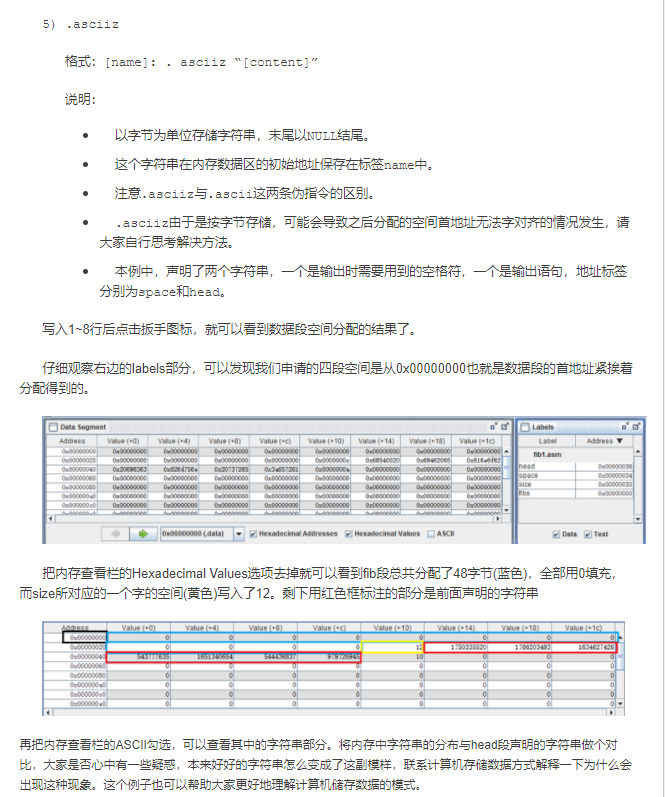




伪指令：

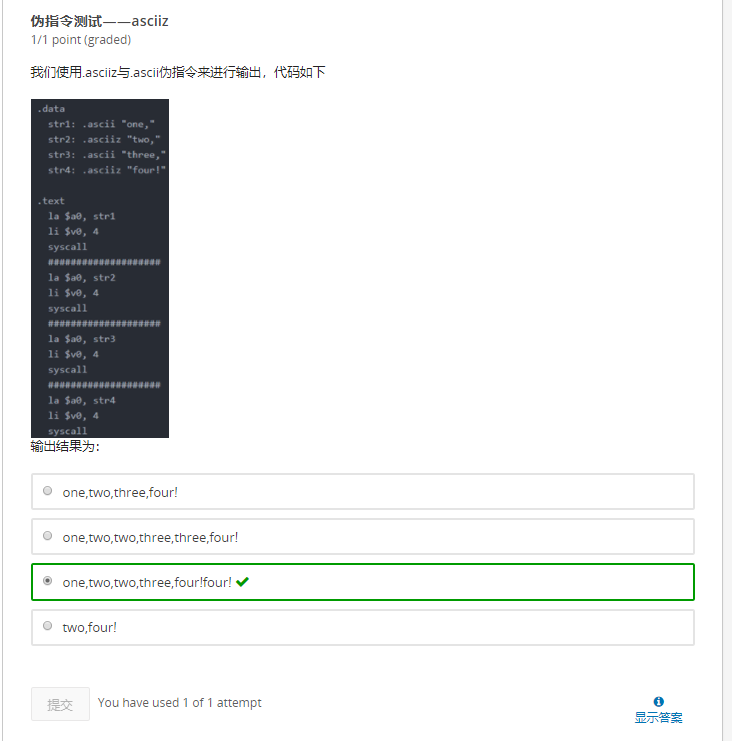






**.space 12 申请12个字节的空间**

**.word 12 申请1个字的空间存12**



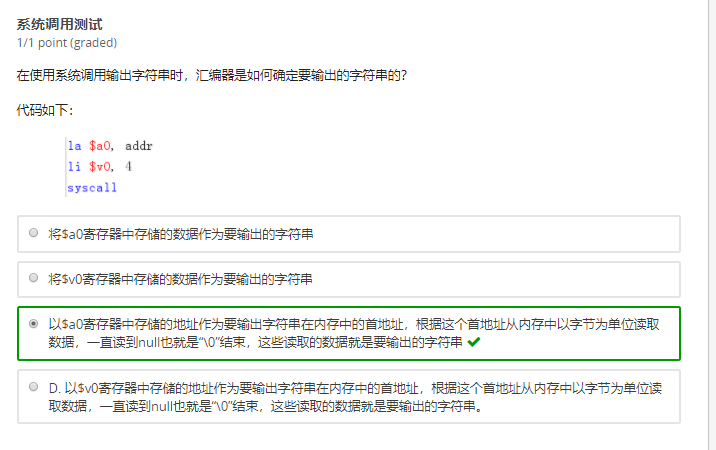
**.asciiz 申请的字符串带\0**

**.ascii 申请的字符串不带\0**

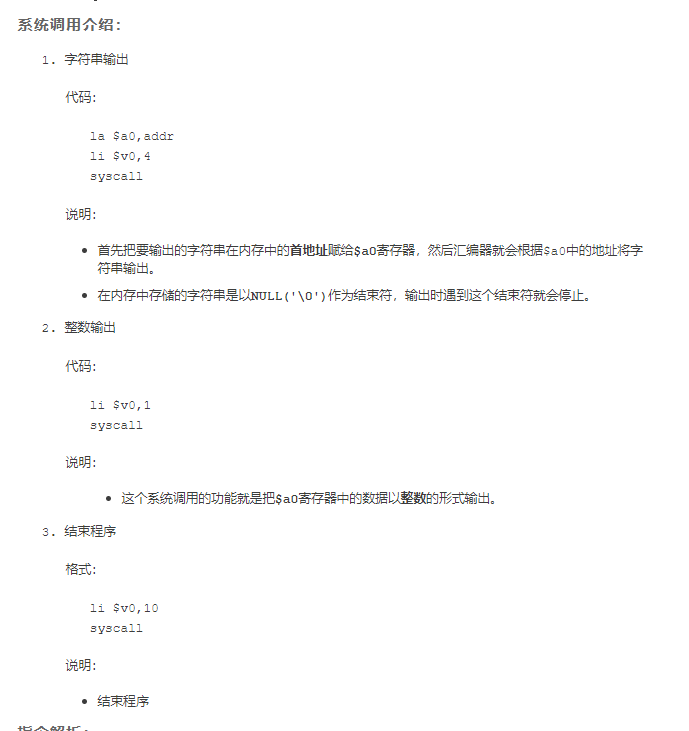
**然后输出时，由于是连续的存储空间，赋给输出指令首地址后，会一直输出到\0为止**



**一个数字一个byte,一个byte两个十六进制**



系统调用：





**$a0存地址**

**$a1存MAXSIZE，但读取的是MAXSIZE-1,留一个给\0**

