

微机原理与接口技术

MIPS 汇编程序设计

华中科技大学 左冬红



术语

与指令集相关

汇编程序设计通常除了使用符号指令之外，还会使用到两类指令：伪指令和宏指令

与汇编程序相关

伪指令

为汇编程序将符号指令翻译成机器指令提供信息，没有对应的机器指令。

宏指令

将一段汇编符号指令序列定义为一条宏指令，简化编程人员撰写程序的工作。
对应机器指令序列

常用伪指令

定义程序结构

伪指令格式

.data[地址]

.text[地址]

伪指令功能

定义数据段的起始地址，
申明之后的内容为数据

定义代码段的起始地址，
申明之后的内容为指令

地址为可选参数，缺省表示由系统自动分配

常用伪指令

存储空间分配及数据类型申明

伪指令格式

[变量名:].byte b0[,b1,b2,...bn]

[变量名:].half h0[,h1,h2,...hn]

[变量名:].word w0[,w1,w2,...wn]

[变量名:].float f0[,f1,f2,...fn]

[变量名:].double d0[,d1,d2,...dn]

[变量名:].ascii "....."

[变量名:].asciiz "....."

边界对齐存储数据

同一定义中各数据连续存储

变量名表示各个存储区首地址

伪指令功能

字节，每个数据占1个存储单元

半字，每个数据占2个存储单元

字，每个数据占4个存储单元

单精度，每个数据占4个存储单元

双精度，每个数据占8个存储单元

ASCII，每个字符占1个存储单元

ASCII，每个字符占1个存储单元，并在下一个单元存储字符串结束符

常用伪指令

伪指令格式

`.align n`

`[变量名:].space n`

`.global 变量名/标号名`

伪指令功能

指示下一数据地址为 2^n 的整数倍

预留n个存储单元，不初始化存储空间

声明全局变量、标号

数据定义伪指令示例

```
.data 0x10014000
.align 2
str: .ascii "abcd"
strn: .ascii "abcdefg"
b0: .byte 1,2,3,4,5
h0: .half 1,2,3,4
w0: .word 1,2,3,4
w1: .word str, strn, b0, h0, w0
```

	0	1	2	3	4	5	6	7
0x10014000	0x61	0x62	0x63	0x64	0x61	0x62	0x63	0x64
0x10014008	0x65	0x66	0x67	0x00	0x01	0x02	0x03	0x04
0x10014010	0x05		0x00	0x01	0x00	0x02	0x00	0x03
0x10014018	0x00	0x04			0x00	0x00	0x00	0x01
0x10014020	0x00	0x00	0x00	0x02	0x00	0x00	0x00	0x03
0x10014028	0x00	0x00	0x00	0x04	0x10	0x01	0x40	0x00
0x10014030	0x10	0x01	0x40	0x04	0x10	0x01	0x40	0x0c
0x10014030	0x10	0x01	0x40	0x12	0x10	0x01	0x40	0x1c

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10014000	0x64636261	0x64636261	0x00676665	0x04030201	0x00010005	0x00030002	0x00000004	0x00000001
0x10014020	0x00000002	0x00000003	0x00000004	0x10014000	0x10014004	0x1001400c	0x10014012	0x1001401c
0x10014040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10014000 是否 2^2 的整数倍?

0x10014011 是否 2 的整数倍?

0x1001401a 是否 4 的整数倍?

常用宏指令

指令格式

li \$Rt, Imm

la \$Rt, Label

指令功能

RF[\$Rt] = Imm (32位)

RF[\$Rt] = Label (32位)

lui \$at, Imm/Label(高16位)
ori \$Rt, \$at, Imm/Label(低16位)

自定义宏指令

```
.macro done  
li $v0,10  
syscall  
.end_macro
```

```
.macro print_int (%x)  
li $v0, 1  
add $a0, $zero, %x  
syscall  
.end_macro
```

```
.macro  
.end_macro为伪指令
```

done为宏指令名称，此宏指令无操作数

```
print_int (%x)
```

%x: 寄存器、立即数

详情见Mars help

系统功能调用

系统功能调用由系统软件提供，作用为屏蔽不同特定硬件的具体操作，作为硬件抽象层。

Mars SPIM提供的系统功能调用包含基本输入、输出调用。

系统功能调用使用步骤为：

设置调用号

设置入口参数

调用系统功能

处理出口参数

常用系统功能调用

输出十进制整数

\$v0=1
\$a0=输出的数据
syscall

输入十进制整数

\$v0=5
syscall
\$v0=输入十进制数的二进制值

输出字符串

\$v0=4
\$a0=字符串首地址
syscall

直到字符串结束符停止

常用系统功能调用

输入字符串

\$v0=4
\$a0=字符串首地址
\$a1=最大输入字符个数
syscall

输入字符个数**包含回车**，若达到最大个数，最后一个字符填充**结束符**；若未达到最大个数，最后一个字符填充**回车符**。

退出程序

\$v0=10
syscall

宏汇编程序示例

.data

.align 2

str: .ascii "abcd"

strn: .asciiz "ABCDEFGH"

b0: .byte 1,2,3,4,5

.text

main: li \$v0,1

li \$a0,0x200

syscall

li \$v0,4

la \$a0,str

syscall

la \$a0,strn

syscall

li \$v0,5

syscall

li \$v0,8

la \$a0,b0

li \$a1,5

syscall

li \$v0,10

syscall

小结

- 宏汇编程序
 - 符号指令
 - 伪指令
 - 宏指令
- 汇编程序一般结构
 - 数据段
 - 代码段
- 系统功能调用
 - 系统函数，硬件抽象层

下一讲：微处理器基本构成