**第十一章 汇编语言**

1.**符号地址**：使用符号表示存储单元的地址。

2.高级语言编程对用户更加友好，但放弃了对指令的精确控制。

3.程序执行前将汇编语言翻译成适合于期望执行计算机的ISA程序。翻译程序是**汇编器**，翻译过程是**汇编**。

4.指令不区分大小写。

5.自由格式，单词间和行间空格不改变程序意义。

6.寄存器共32个，R0 、R1…R31。

7.**寄存器限制**

①**输入输出：R4**

②**局部变量**：**R16、R17、R18、R19、R20、R21、R22、R23**

③**临时值**：**R8、R9、R10、R11、R12、R13、R14、R15、R24、R25**

④**全局指针：R28**

⑤**栈指针：R29**

⑥**帧指针**（**框架指针**）**：R30**

8.**立即数基符号**：**# 十进制**；**x 十六进制**；**b 二进制**

9.**算数/逻辑运算指令**

**I-类型**：OPCODE DR SR1 Imm/LABEL

**R-类型**：OPCODE DR SR1 SR2

**LHI指令**：LHI DR Imm/LABEL

10.**数据传送指令**

**加载指令：**LW/LB DR Imm/LABEL(SR1)

**存储指令：**SW/SB Imm/LABEL(SR1) DR

**LW和SW指令基址寄存器+偏移量得到的地址必须是4的倍数。**

11.**控制指令**

**条件分支指令：**BEQZ/BNEZ SR1 LABEL

**J指令：**J LABEL

**JR指令：**JR SR1

**TRAP指令：**TRAP Imm（**trap向量**）

12.**标记（LABEL）**：用来明确标识存储单元的符号名。

13.标记由**字母**、**数字**、**下划线**组成，**以字母、下划线、$开头**，**以冒号：结尾**。

14.**指令操作码不可以作为标记名**。

15.**注释以分号；开头**，至行末的内容被忽略。

16.使用空格将程序对齐。

17.注释目的：提高可读性。

18.**伪操作**有助于汇编器把输入的字符串作为使用DLX汇编语言写的计算机程序，并将其翻译为DLX的ISA程序。汇编完成即被抛弃，程序执行过程中的伪操作不执行。

19.伪操作**以点.开头。第一行必须为.data address。必须有.global main**20.**汇编过程**至少扫描两趟。

第一趟：构建符号表。

第二趟：把汇编语言指令翻译成机器语言指令。

21.**LC**：**地址计数器**

22.一条汇编语言指令可能翻译成多条机器语言指令。（大立即数由LHI实现）

23.**可执行映像**：当计算机开始一个程序的执行时的被执行实体。

24.链接过程由**链接器**完成。链接器为每个模块重新分配存储空间，因此.text address和.data address中的address是可省略的。

25.**栈**是一种抽象数据类型。

26.**栈协议**：**后进先出**（**LIFO**）

27.**栈顶**（**TOP**）：数据只能经栈顶处访问。

28.**硬件栈**：push时数据向上移动，pop时数据向下移动，栈顶固定在底层。

29.**栈指针**（**R29**）：存储栈顶的单元地址。push时数据向上添加，栈顶上移一格，R2减4；pop时数据不进行物理移动，栈顶下移一格，R29加4，**原数据不删除**。

30.**push**（**压栈**）：把一个元素插入栈。

31.push: subi r29,r29,#4

sw 0(r29),SR

32.**pop**（**出栈**）：从栈中移出一个元素。

33.pop: lw DR,0(r29)

addi r29,r29,#4

34.当变量个数多于寄存器数目，需要使用存储器。**存储时边界对齐**。

35.DLX存储器组织

|  |  |  |
| --- | --- | --- |
| x0000 0000 | 系统空间 |  |
|  | 代码区 | ←PC  ←R28（全局指针，指向全局数据区起始地址） |
|  | 全局数据区  （存储所有静态存储类变量） |
|  | 堆  （动态分配存储单元） |  |
|  | ↓  ↑ | ←R29（栈指针，指向运行时栈栈顶）  ←R30（帧指针，指向运行时栈中） |
|  | 运行时栈  （存储缺省自动存储类变量） |
| xFFFF FFFF | 系统空间 |  |