链滴

汇编语言 学习笔记 (四)

作者：houjie

原文链接：https://ld246.com/article/1510213556401

来源网站：链滴

汇编语言学习笔记

五、汇编语言程序设计

顺序

分支

标号

循环

- 初始化：设置循环的初始状态
- 循环体：循环的工作部分及修改部分
- 控制条件：
  - 已知次数(loop)
  - 次数及特征值(loopz,loopnz)
  - 不知道次数，其它条件

JS　opr　；　结果为负转移,SF=1
JNS　opr　；　结果为正转移,SF=0
JE/JZ　opr　；　相等或为零转移,ZF=1
JNE/JNZ　opr　；　不相等或不为零转移,ZF=0

子程序

子程序的设计方法

过程定义伪操作

```
procedure name    PROC  Attribute
...
procedure name    ENDP
```

其中**Attribute**是可以是**NEAR**或**FAR**。
如调用程序和过程在同一个代码段中，则使用NEAR属性；反之则要使用FAR属性。
如果定义的过程是NEAR属性的，那么对它的调用和返回也一定是NEAR属性的；反之则都是FAR属性。

子程序的调用和返回

由CALL和 RET指令完成子程序的调用和返回。特别要注意的是在子程序中使用堆栈时，应保证进栈

出栈的数据个数要相等，否则会造成程序无法正确返回。

```
call  [near ptr]  subp        ;   段内调用
call  far  ptr  subp       ;    段间调用
```

保存与恢复寄存器

主程序和子程序都使用 CPU中的寄存器。需要在子程序的开始将子程序要使用的寄存器的内容压入堆，在退出子程序前把这些寄存器的内容恢复原状。

(有些运算有隐含的寄存器，指令涉及的寄存器都要注意保存)**用来传递参数的寄存器的内容则不需要护。**

子程序的参数传递

通过寄存器传送参数

主程序与子程序间传递的参数都在约定的寄存器中—传递单元在CPU内部。

在调用子程序前主程序将入口参数送约定寄存器中,子程序直接从这些寄存器中取得参数进行计算处理经加工处理后得到的结果(出口参数)也放在约定寄存器中 ,返回主程序后,主程序直接到该寄存器中取结果。该法简单、直观,信息传递快,但寄存器个数有限,所以适用于参数较少的情况

直接访问模块中的变量

如过程和调用程序在同一源文件(同一程序模块)中，则过程可直接访问模块中的变量。

通过堆栈传送参数或参数地址

入口和出口参数都放在堆栈中—传递单元在SS段。

调用前,入口参数由主程序送入堆栈,子程序从堆栈中取得这些参数进行处理,处理后的结果又送到堆栈中返回主程序后，主程序从堆栈取得结果。此法不占用公共寄存器,也无需另外开辟单元，但因为子程序返回地址也在堆栈中,所以一定要小心计算参数与地址,否则出错。

通过地址表传送参数地址

子程序的嵌套和举例

一个子程序作为调用程序去调用另外一个子程序，这种情况称为子程序的嵌套。

在子程序嵌套的情况下，如果一个子程序调用的子程序是其自身，称为递归调用。这样的子程序称为归子程序。

高级汇编语言技术

宏汇编

定义后的宏指令就可以在源程序中调用，称为宏调用。宏调用的格式为

macro name [actual parameter list]

其中actual parameter list称为实元表。

当源程序被汇编时，汇编程序将对每个宏调用做宏展开。宏展开就是用宏定义体取代源程序中的宏指名，而且用实元取代宏定义中的哑元