

少即是多，慢即是快
不够认真就肤浅了，过于认真就较真了

首页 日志 LOFTER 相册 音乐 收藏 博友 关于我

GNU风格 ARM汇编语法指南（非常详细）4

什么是UML

关于我

网易 博客 LOFTER—似桃花微醺了芬芳 LOFTER-夏日里的清新女神

马来西亚仙本娜的天空之境 加关注

GNU风格 ARM汇编语法指南（非常详细）5

2010-12-14 20:24:17 | 分类： ARM学习 | 标签： gnu风格 arm汇编语法指南（非常详细）5 | 字号 | 举报 订阅

我的照片书 | 下载LOFTER

6、 GNU汇编程序中的常数

- <1> 十进制数以非0数字开头,如:123和9876;
- <2> 二进制数以0b开头,其中字母也可以为大写;
- <3> 八进制数以0开始,如:0456,0123;
- <4> 十六进制数以0x开头,如:0xabcd,0X123f;
- <5> 字符串常量需要用引号括起来,中间也可以使用转义字符,如: "You are welcome!\n";
- <6> 当前地址以"."表示,在GNU汇编程序中可以使用这个符号代表当前指令的地址;
- <7> 表达式: 在汇编程序中的表达式可以使用常数或者数值,"-"表示取负数,"~"表示取补,"<>"表示不相等,其他的符号如:+、-、*、/、%、<、<<、>、>>、|、&、^、!、==、>=、<=、&&、|| 跟C语言中的用法相似。

7、 GNU ARM汇编的常用伪操作

在前面已经提到过了一些为操作，还有下面一些为操作：

- 数据定义伪操作：.byte、.short、.long、.quad、.float、.string/.asciz/.ascii，重复定义伪操作.rept，赋值语句.equ/.set；
- 函数的定义；
- 对齐方式伪操作 .align；
- 源文件结束伪操作.end；
- .include伪操作；
- if伪操作；
- .global/.globl 伪操作；
- .type伪操作；
- 列表控制语句；

别于GNU AS汇编的通用伪操作,下面是ARM特有的伪操作：

.reg，.unreq，.code，.thumb，.thumb_func，.thumb_set，.ltorg，.pool

<1> 数据定义伪操作

- .byte:单字节定义，如：.byte 1,2,0b01,0x34,072,'s'；
- .short:定义双字节数据，如:.short 0x1234,60000；
- .long:定义4字节数据，如:.long 0x12345678,23876565
- .quad:定义8字节，如:.quad 0x1234567890abcd
- .float: 定义浮点数，如：.float 0f-314159265358979323846264338327195028841971.693993751E-40 @ - pi
- .string/.asciz/.ascii: 定义多个字符串，如：
.string "abcd", "efgh", "hello!"
.asciz "qwer", "sun", "world!"
.ascii "welcome\0"



reboot

* 每天都修炼一下英语流利说 * 不要让注意力局限在技术细节上 * 耐心地接受外界建设性的信息 * 注意休息,保重身体

加博友 关注他



文章分类

- 移动设备安全（1）
- 内核+驱动（17）
- 英语学习（89）
- IT专业术语（53）
- ARM学习（29）
- C/C++\数据结构（87）
- 宏观角度（19）
- 引导加载程序--boot（43）
- 更多 >

LOFTER精选

- .rept:重复定义伪操作, 格式如下:

 .rept 重复次数

 数据定义

 .endr @结束重复定义

例:

 .rept 3

 .byte 0x23

 .endr

- .equ/.set: 赋值语句, 格式如下:

 .equ(.set) 变量名, 表达式

例:

 .equ abc, 3 @让abc=3

<2> 函数的定义伪操作

- 函数的定义,格式如下:

 函数名:

 函数体

 返回语句

一般的,函数如果需要在其他文件中调用,需要用到.global伪操作将函数声明为全局函数。为了不至于在其他程序在调用某个C函数时发生混乱,对寄存器的使用我们需要遵循APCS准则。函数编译器将处理函数代码为一段.global的汇编码。

- 函数的编写应当遵循如下规则:

a. a1-a4寄存器（参数、结果或暂存寄存器, r0到r3 的同义字）以及浮点寄存器f0-f3(如果存在浮点协处理器)在函数中是不必保存的;

b. 如果函数返回一个不大于一个字大小的值,则在函数结束时应该把这个值送到 r0 中;

c. 如果函数返回一个浮点数,则在函数结束时把它放入浮点寄存器f0中;

d. 如果函数的过程改动了sp（堆栈指针, r13）、fp（框架指针, r11）、sl（堆栈限制, r10）、lr（连接寄存器, r14）、v1-v8（变量寄存器, r4 到 r11）和 f4-f7,那么函数结束时这些寄存器应当被恢复为包含在进入函数时它所持有的值。

<3> .align .end .include .incbin伪操作

- .align:用来指定数据的对齐方式,格式如下:

 .align [absexpr1, absexpr2]

以某种对齐方式,在未使用的存储区域填充值. 第一个值表示对齐方式,4, 8,16或 32. 第二个表达式值表示填充的值。

- .end:表明源文件的结束。
- .include:可以将指定的文件在使用.include 的地方展开,一般是头文件,例如:

 .include "myarmasm.h"
- .incbin伪操作可以将原封不动的一个二进制文件编译到当前文件中,使用方法如下:

 .incbin "file"[,skip[,count]]

skip表明是从文件开始跳过skip个字节开始读取文件,count是读取的字数.

<4> .if伪操作

根据一个表达式的值来决定是否要编译下面的代码,用.endif伪操作来表示条件判断的结束,中间可以使用.else来决定.if的条件不满足的情况下应该编译哪一部分代码。

.if有多个变种:

- .ifdef symbol @判断symbol是否定义
- .ifc string1,string2 @字符串string1和string2是否相等,字符串可以用单引号括起来
- .ifeq expression @判断expression的值是否为0
- .ifeqs string1,string2 @判断string1和string2是否相等,字符串必须用双引号括起来
- .ifge expression @判断expression的值是否大于等于0



王凯 TFBBoys 深夜食堂

王者荣耀 森系 鹿晗 女神

萌宠 樱花 美妆

白丝

八招诀窍，教你实力撩妹 >

网易考拉推荐



GACHA精选

.ifle expression @判断expression的值是否小于等于0

.iflt absolute expression @判断expression的值是否小于0

.ifnc string1,string2 @判断string1和string2是否不相等, 其用法跟.ifc恰好相反。

.ifndef symbol, .ifndef symbol @判断是否没有定义symbol, 跟.ifdef恰好相反

.ifne expression @如果expression的值不是0, 那么编译器将编译下面的代码

.ifnes string1,string2 @如果字符串string1和string2不相 等, 那么编译器将编译下面的代码。

<5> .global .type .title .list

- .global/ .globl : 用来定义一个全局的符号, 格式如下:
.global symbol 或者 .globl symbol
- .type: 用来指定一个符号的类型是函数类型或者是对象类型, 对象类型一般是数据, 格式如下:
.type 符号, 类型描述

例:

```
.globl a
.data
.align 4
.type a, @object
.size a, 4
a:
.long 10
例:
.section .text
.type asmfunc, @function
.globl asmfunc
asmfunc:
mov pc, lr
```

<6> 列表控制语句:

.title: 用来指定汇编列表的标题,例如:

.title "my program"

.list: 用来输出列表文件.

<7> ARM特有的伪操作

- .reg: 用来给寄存器赋予别名,格式如下:
别名 .req 寄存器名
- .unreq: 用来取消一个寄存器的别名,格式如下:
.unreq 寄存器别名

注意被取消的别名必须事先定义过,否则编译器就会报错,这个伪操作也可以用来取消系统预制的别名, 例如r0, 但如果没有必要的话不推荐那样做。

- .code伪操作用来选择ARM或者Thumb指令集,格式如下:
.code 表达式

如果表达式的值为16则表明下面的指令为Thumb指令,如果表达式的值为32则表明下面的指令为ARM指令。

- .thumb伪操作等同于.code 16, 表明使用Thumb指令, 类似的.arm等同于.code 32
- .force_thumb伪操作用来强制目标处理器选择thumb的指令集而不管处理器是否支持
- .thumb_func伪操作用来指明一个函数是thumb指令集的函数
- .thumb_set伪操作的作用类似于.set, 可以用来给一个标志起一个别名, 比.set功能增加的一点是可以把一个标志标记为thumb函数的入口, 这点功能等同于.thumb_func
- .ltorg用于声明一个数据缓冲池(literal pool)的开始,它可以分配很大的空间。
- .pool的作用等同.ltorg
- .space <number_of_bytes> {,<fill_byte>}



热门动漫一网打尽



armasm中的SPACE功能相同)

- .word <word1> {,<word2>} ...

插入一个**32-bit**的数据队列。(与armasm中的DCD功能相同)。可以使用**.word**把标识符作为常量使用。

例:

Start:

valueOfStart:

.word Start

这样程序的开头**Start**便被存入了内存变量**valueOfStart**中。

- .hword <short1> {,<short2>} ...

插入一个**16-bit**的数据队列。(与armasm中的DCW相同)

8、 GNU ARM汇编特殊字符和语法

- <1> 代码行中的注释符号: '@'
- <2> 整行注释符号: '#'
- <3> 语句分离符号: ';'
- <4> 立即数前缀: '#' 或 '\$'

阅读(3321) | 评论(0)

[转载](#) [推荐](#)

[GNU风格 ARM汇编语法指南（非常详细）4](#) [什么是UML](#)

历史上的今天

类型安全 2009-12-14 20:24:54

LOFTER七夕点歌台
送给爱的人一首歌

马上参与



参与送耳机

评论

登录后你可以发表评论，请先登录。[登录>>](#)