

lxgeek

有的事情现在不做，以后一辈子都不会做了

博客园 首页 新随笔 联系 订阅 管理

随笔 - 111 文章 - 0 评论 - 22

汇编指令

GAS中每个操作都是有一个字符的后缀，表明操作数的大小。

C声明	GAS后缀	大小(字节)
char	b	1
short	w	2
(unsigned) int / long / char*	l	4
float	s	4
double	l	8
long double	t	10/12

注意：GAL使用后缀“l”同时表示4字节整数和8字节双精度浮点数，这不会产生歧义因为浮点数使用的是完全不同的指令和寄存器。

操作数格式：

格式	操作数值	名称	样例（GAS = C语言）
\$Imm	Imm	立即数寻址	\$1 = 1
Ea	R[Ea]	寄存器寻址	%eax = eax
Imm	M[Imm]	绝对寻址	0x104 = *0x104
(Ea)	M[R[Ea]]	间接寻址	(%eax) = *eax
Imm(Ea)	M[Imm+R[Ea]]	(基址+偏移量)寻址	4(%eax) = *(4+eax)

公告

昵称：lxgeek
园龄：10年
粉丝：33
关注：2
[+加关注](#)

2011年1月						
<	日	一	二	三	四	五
	26	27	28	29	30	31
	2	3	4	5	6	7
	9	10	11	12	13	14
	16	17	18	19	20	21
	23	24	25	26	27	28
	30	31	1	2	3	4
						5

搜索

找找看

谷歌搜索

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

随笔分类

[c\(22\)](#)
[kernel\(27\)](#)
[linux\(15\)](#)
[MPTCP\(8\)](#)
[PF_RING\(2\)](#)
[python\(14\)](#)
[每日一技\(67\)](#)
[内核源码分析\(3\)](#)
[算法\(8\)](#)
[网络编程\(10\)](#)
[我的实验\(13\)](#)
[字节缓存\(3\)](#)

随笔档案

[2015年10月\(1\)](#)
[2015年8月\(2\)](#)
[2015年6月\(1\)](#)
[2015年3月\(7\)](#)
[2014年12月\(3\)](#)
[2014年11月\(4\)](#)
[2014年10月\(2\)](#)
[2014年8月\(2\)](#)
[2014年6月\(1\)](#)
[2014年4月\(1\)](#)
[2014年1月\(1\)](#)
[2013年11月\(3\)](#)
[2013年6月\(1\)](#)

(Ea,Eb)	M[R[Ea]+R[Eb]]	变址	(%eax,%ebx) = *(eax+ebx)
Imm (Ea,Eb)	M[Imm+R[Ea]+R[Eb]]	寻址	9(%eax,%ebx)=*(9+eax+ebx)
(,Ea,s)	M[R[Ea]*s]	伸缩化变址寻址	(,%eax,4)= *(eax*4)
Imm(,Ea,s)	M[Imm+R[Ea]*s]	伸缩化变址寻址	0xfc(,%eax,4)= *(0xfc+eax*4)
(Ea,Eb,s)	M(R[Ea]+R[Eb]*s)	伸缩化变址寻址	(%eax,%ebx,4)= *(eax+ebx*4)
Imm(Ea,Eb,s)	M(Imm+R[Ea]+R[Eb]*s)	伸缩化变址寻址	8(%eax,%ebx,4)= *(8+eax+ebx*4)

注：M[xx]表示在存储器中xx地址的值，R[xx]表示寄存器xx的值，这种表示方法将寄存器、内存都看出一个大数组的形式。

数据传送指令：

指令	效果	描述
movl S,D	D <-- S	传双字
movw S,D	D <-- S	传字
movb S,D	D <-- S	传字节
movsbl S,D	D <-- 符号扩展S	符号位填充(字节->双字)
movzbl S,D	D <-- 零扩展S	零填充(字节->双字)
pushl S	R[%esp] <-- R[%esp] - 4; M[R[%esp]] <-- S	压栈
popl D	D <-- M[R[%esp]]; R[%esp] <-- R[%esp] + 4;	出栈

2013年5月(4)
2012年10月(1)
更多

最新评论

1. Re:MPTCP 源码分析(一) MPTCP的三次握手
@ allen-zhao123@allen-zhao123引用博主，求源码啊，实在是搞不懂...
--Chirfen
2. Re:MPTCP 源码分析(一) MPTCP的三次握手
博主，求源码啊，实在是搞不懂
--allen-zhao123
3. Re:装了 nProbe 之后
xuexile
--规格严格-功夫到家
4. Re:汇编指令
终于有一篇总算看懂了..大爱你了~~~
--infiniteable
5. Re:vlan 介绍
你真是好厉害，我最近也想学习无线网络的问题，下了一本书就看了一个开头。
--liulili

阅读排行榜

1. 汇编指令(55428)
2. MPTCP 理解(15972)
3. PF_RING 总结(15943)
4. 排查 “Detected Tx Unit Hang”问题(5167)
5. opendpi 源码分析 （一）(4833)

评论排行榜

1. 今天才知道 printf 有返回值，而且是什么意思(3)
2. 装了 nProbe 之后(3)
3. MPTCP 源码分析(三) 子路径选择(2)
4. MPTCP 源码分析(一) MPTCP的三次握手(2)
5. 在结构体里放 string 是不行的(2)

推荐排行榜

1. 汇编指令(9)
2. 输出C语言中 变量的类型(2)
3. opendpi 源码分析 （一）(2)
4. MPTCP 理解(1)
5. 轮询算法 这是一个印度人写的，学习下。来自 co deproject(1)

注：均假设栈往低地址扩展。

算数和逻辑操作地址：

指令	效果	描述
leal S,D	D = &S	movl地版，S地址入D，D仅能是寄存器
incl D	D++	加1
decl D	D--	减1
negl D	D = -D	取负
notl D	D = ~D	取反
addl S,D	D = D + S	加
subl S,D	D = D – S	减
imull S,D	D = D*S	乘
xorl S,D	D = D ^ S	异或
orl S,D	D = D S	或
andl S,D	D = D & S	与
sall k,D	D = D << k	左移
shll k,D	D = D << k	左移(同sall)
sarl k,D	D = D >> k	算数右移
shrl k,D	D = D >> k	逻辑右移

特殊算术操作：

指令	效果	描述
imull S	$R[\%edx]:R[\%eax] = S * R[\%eax]$	无符号64位乘
mull S	$R[\%edx]:R[\%eax] = S * R[\%eax]$	有符号64位乘
cld S	$R[\%edx]:R[\%eax] = \text{符号位扩展} R[\%eax]$	转换为4字节
idivl S	$R[\%edx] = R[\%edx]:R[\%eax] \% S;$ $R[\%eax] = R[\%edx]:R[\%eax] / S;$	有符号除法，保存余数和商
divl S	$R[\%edx] = R[\%edx]:R[\%eax] \% S;$ $R[\%eax] = R[\%edx]:R[\%eax] / S;$	无符号除法，保存余数和商

注：64位数通常存储为，高32位放在edx，低32位放在eax。

条件码：

条件码寄存器描述了最近的算数或逻辑操作的属性。

CF：进位标志，最高位产生了进位，可用于检查无符号数溢出。

OF：溢出标志，二进制补码溢出——正溢出或负溢出。

ZF：零标志，结果为0。

SF：符号标志，操作结果为负。

比较指令：

指令	基于	描述
cmpb S2,S1	$S1 - S2$	比较字节，差关系
testb S2,S1	$S1 \& S2$	测试字节，与关系
cmpw S2,S1	$S1 - S2$	比较字，差关系
testw S2,S1	$S1 \& S2$	测试字，与关系

cmpl S2,S1	S1 – S2	比较双字，差关系
testl S2,S1	S1 & S2	测试双字，与关系

访问条件码指令：

指令	同义名	效果	设置条件
sete D	setz	D = ZF	相等/零
setne D	setnz	D = ~ZF	不等/非零
sets D		D = SF	负数
setns D		D = ~SF	非负数
setg D	setnle	D = ~(SF ^OF) & ZF	大于（有符号 >）
setge D	setnl	D = ~(SF ^OF)	小于等于(有符号 >=)
setl D	setnge	D = SF ^ OF	小于(有符号<)
setle D	setng	D = (SF ^ OF) ZF	小于等于(有符号 <=)
seta D	setnbe	D = ~CF & ~ZF	超过(无符号>)
setae D	setnb	D = ~CF	超过或等于(无符号 >=)
setb D	setnae	D = CF	低于(无符号<)
setbe D	setna	D = CF ZF	低于或等于(无符号 <=)

跳转指令：

--	--	--	--

指令	同义名	跳转条件	描述
jmp Label		1	直接跳转
jmp *Operand		1	间接跳转
je Label	jz	ZF	等于/零
jne Label	jnz	~ZF	不等/非零
js Label		SF	负数
jnz Label		~SF	非负数
jg Label	jnle	~(SF^OF) & ~ZF	大于(有符号>)
jge Label	jnl	~(SF ^ OF)	大于等于(有符号 >=)
jl Label	jnge	SF ^ OF	小于 (有符号<)
jle Label	jng	(SF ^ OF) ZF	小于等于(有符号 <=)
ja Label	jnbe	~CF & ~ZF	超过(无符号>)
jae Label	jnb	~CF	超过或等于(无符号 >=)
jb Label	jnae	CF	低于(无符号<)
jbe Label	jna	CF ZF	低于或等于(无符号 <=)

转移控制指令：（函数调用）：

指令	描述
call Label	过程调用，返回地址入栈，跳转到调用过程起始处，返回地址是call后面那条指令的地址

call *Operand	址
leave	为返回准备好栈，为ret准备好栈，主要是弹出函数内的栈使用及%ebp

好文要顶

关注我

收藏该文

[lxgeek](#)
[关注 - 2](#)
[粉丝 - 33](#)
[+加关注](#)

9

推荐

0

反对

» 下一篇: [2011.1.18 运算符优先级](#)
posted @ 2011-01-01 14:11 lxgeek 阅读(55428) 评论(1) 编辑 收藏

评论列表

#1楼 2016-03-12 11:18 infinitable
终于有一篇总算看懂了..大爱你了~~~

支持(0)

反对(0)

刷新评论

刷新页面

返回顶部

登录后才能发表评论，立即 [登录](#) 或 [注册](#)， [访问](#) [网站首页](#)

- 【推荐】News: 大型组态、工控、仿真、CADGIS 50万行VC++源码免费下载
- 【推荐】有你助力，更好为你——博客园用户消费观调查，附带小惊喜！
- 【推荐】AWS携手博客园为开发者送福利，注册立享12个月免费套餐
- 【推荐】博客园x丝芙兰-圣诞特别活动：圣诞选礼，美力送递
- 【推荐】了不起的开发者，挡不住的华为，园子里的品牌专区
- 【推荐】未知数的距离，毫秒间的传递，声网与你实时互动
- 【推荐】新一代 NoSQL 数据库，Aerospike专区新鲜入驻

广告 X

香港服务器**免备案**

10大游戏公司都在用

· 免备案 · 10ms延时 · CN2高速带宽

29元起

- 相关博文:
- AT&T汇编指令
 - 汇编指令和标志寄存器
 - 汇编指令和标志寄存器
 - (深入理解计算机系统)AT&T汇编指令
 - NASM汇编指令复习
- » 更多推荐...

企业 0元上云

注册即享[万元]新人礼包

立即注册 ▶

- 最新 IT 新闻:
- Google Chrome浏览器地址栏即将变得更安全、更快速

- 百度地图推出“疫情小区”搜索功能
 - 我国量子存储器取得重大进展！世界首次实现“按需读取”
 - 2021「蛋壳」要彻底碎了？CFO 等高管相继离职，上市不满一年深陷暴雷风波
 - 依图CTO颜水成被曝离职，已加入东南亚电商独角兽Shopee
- » 更多新闻...

Copyright © 2021 lxgeek
Powered by .NET 5.0 on Kubernetes