misterliwei的专栏

莫等闲，白了少年头，空悲切。

个人资料

misterliwei

访问：441767次
积分：5728
等级：BLOG 5
排名：第3377名

原创：88篇
转载：2篇

保护模式、实地址模式及V8086模式下的指令格式（下）

标签：汇编 语言 存储 出版

2010-05-02 18:23 838人阅读 评论(0) 收藏 举报

保护模式、实地址模式及V8086模式下的指令格式（下）

学习汇编语言时，我们都是从寻址方式开始的。所谓寻址方式就是指令中操作数是如何指定的。在16位时，寻址方式有：

1.立即数寻址

操作数直接存放在指令中，紧跟在操作码之后。

2.寄存器寻址

操作数在寄存器中，指令指定寄存器号。

3.存储器寻址，存储器寻址方式是说操作数存放在内存之中，指定这些内存的有效地址（effective address）有下面这几种方法。

3.1 直接寻址

直接寻址时，操作数的有效地址紧跟操作码之后。比如：MOV AX, [0X1000]

3.2 寄存器间接寻址

操作数的有效地址存放在基址寄存器BX和BP、或者变址寄存器SI和DI中。

EA = 16 * DS + SI

或EA = 16 * DS + DI

或EA = 16 * DS + BX

或EA= 16 * SS + BP　(注意：BP的缺省段寄存器为SS，其他都为DS)

3.3 寄存器相对寻址

操作数的有效地址是一个基址或变址寄存器的内容和指令中指定的8位或16为偏移量的和。注意这个偏移量是带符号的，即可正可负。

EA = 16 * DS + SI + DISP8/16

或 EA = 16 * DS + DI + DISP8/16

或 EA = 16 * DS + BX + DISP8/16

或 EA = 16 * SS + BP + DISP8/16

3.4 基址变址寻址

操作数的有效地址是一个基址寄存器和一个变址寄存器的内容之和。

EA = 16 * DS + BX + SI或 DI

或 EA = 16 * SS + BP + SI或 DI

3.5 相对基址变址寻址

操作数的有效地址是一个基址寄存器和一个变址寄存器的内容和8位或16位偏移量之和。同上面一样，这个偏移量是带符号的，即可正可负。

EA = 16 * DS + BX + SI(或 DI) + DISP8/16

或 EA = 16 * SS + BP + SI(或 DI) + DISP8/16

上面的这些寻址方式，我将它在表2-1中表示了出来，见表2-1a。当初学习寻址方式时总是一头雾水，现在将它和I32的指令码格式联系在一起总算明白了。:)

表 2-1a 16 位寻址方式 (ModR/M)

r8(/r) r16(/r) r32(/r) mm(/r) xmm(/r) (In decimal) /digit (Opcode) (In binary) REG =			AL AX EAX MM0 XMM0 0 000	CL CX ECX MM1 XMM1 1 001	DL DX EDX MM2 XMM2 2 010	BL BX EBX MM3 XMM3 3 011	AH SP ESP MM4 XMM4 4 100	CH BP1 EBP MM5 XMM5 5 101	DH SI ESI MM6 XMM6 6 110	BH DI EDI MM7 XMM7 7 111
Effective Address	**Mod**	**R/M**	**Value of ModR/M Byte (in Hexadecimal)**							
[BX+SI]	00	000	00	08	10	18	20	28	30	38
[BX+DI]		001	01	09	11	19	21	29	31	39
[BP+SI]		010	02	0A	12	1A	22	2A	32	3A
[BP+DI]		011	03	0B	13	1B	23	2B	33	3B
[SI]		100	04	0C	14	1C	24	2C	34	3C
[DI]		101	05	0D	15	1D	25	2D	35	3D
disp16[2]		110	06	0E	16	1E	26	2E	36	3E
[BX]		111	07	0F	17	1F	27	2F	37	3F
[BX+SI]+disp8[3]	01	000	40	48	50	58	60	68	70	78
[BX+DI]+disp8		001	41	49	51	59	61	69	71	79
[BP+SI]+disp8		010	42	4A	52	5A	62	6A	72	7A
[BP+DI]+disp8		011	43	4B	53	5B	63	6B	73	7B

方案业务连接透明化（中）

最新评论

IoAllocateDriverObjectExtension
misterliwei: @xzyee:是的。然后根据反编译写的伪代码。

IoAllocateDriverObjectExtension
xzyee: 好！这代码是从反编译码得到的？

WINDOWS内核对象
misterliwei: @mr_dong110:可以，保留作者信息就行。

WINDOWS内核对象
mr_dong110: 博主，你写的文章内容 很好 我可以 转载吗？

数据库读写分离和垂直分库、水平
misterliwei: @u014665416:一个从表中读取

有方法读取一个已被其他进程打开
changshenglugu: -_-|||马了再说

创建、分离、重新附加并修复一个
changshenglugu: 看不懂怎么办，先收藏了

有方法读取一个已被其他进程打开
LiAuH: 感谢楼主，顺便mark

MS SQL SERVER 2005 MDF文件
u010548682: @tianyi_iflytek:请问这个后来是怎么解决的，我也是用了这个程序修改之后就提示不是主文件...

GetWindowText函数的一个注意点
D_T: 正在找没取出的原因 好评

[SI]+disp8 [DI]+disp8 [BP]+disp8 [BX]+disp8		100 101 110 111	44 45 46 47	4C 4D 4E 4F	54 55 56 57	5C 5D 5E 5F	64 65 66 67	6C 6D 6E 6F	74 75 76 77	7C 7D 7E 7F
[BX+SI]+disp16 [BX+DI]+disp16 [BP+SI]+disp16 [BP+DI]+disp16 [SI]+disp16 [DI]+disp16 [BP]+disp16 [BX]+disp16	10	000 001 010 011 100 101 110 111	80 81 82 83 84 85 86 87	88 89 8A 8B 8C 8D 8E 8F	90 91 92 93 94 95 96 97	98 99 9A 9B 9C 9D 9E 9F	A0 A1 A2 A3 A4 A5 A6 A7	A8 A9 AA AB AC AD AE AF	B0 B1 B2 B3 B4 B5 B6 B7	B8 B9 BA BB BC BD BE BF
EAX/AX/AL/MM0/XMM0 ECX/CX/CL/MM1/XMM1 EDX/DX/DL/MM2/XMM2 EBX/BX/BL/MM3/XMM3 ESP/SP/AHMM4/XMM4 EBP/BP/CH/MM5/XMM5 ESI/SI/DH/MM6/XMM6 EDI/DI/BH/MM7/XMM7	11	000 001 010 011 100 101 110 111	C0 C1 C2 C3 C4 C5 C6 C7	C8 C9 CA CB CC CD CE CF	D0 D1 D2 D3 D4 D5 D6 D7	D8 D9 DA DB DC DD DE DF	E0 EQ E2 E3 E4 E5 E6 E7	E8 E9 EA EB EC ED EE EF	F0 F1 F2 F3 F4 F5 F6 F7	F8 F9 FA FB FC FD FE FF

寄存器寻址　　寄存器相对寻址

直接寻址　　基址变址寻址

寄存器间接寻址　　相对基址变址寻址

上图中我们还发现一个问题：寄存器间接寻址中可以使用SI、DI、BX、BP这四个寄存器。但是我们看到Mod=00时只能表示SI、DI、BX三个寄存器，那么BP是如何表示的呢？原来它是通过在MOD=01和10时，设置disp8和disp16偏移值来实现的。所以在编译下面两条指令时可能差别很大的。

指令	编码
Mov ax, [bx]	8B 07
Mov ax, [bp]	8B 46 00

32位寻址方式和16位上述的寻址方式基本一致见表2-2a。

表 2-2a 32 位寻址方式 (ModR/M)

r8(/r) r16(/r) r32(/r) mm(/r) xmm(/r) (In decimal) /digit (Opcode) (In binary) REG =			AL AX EAX MM0 XMM0 0 000	CL CX ECX MM1 XMM1 1 001	DL DX EDX MM2 XMM2 2 010	BL BX EBX MM3 XMM3 3 011	AH SP ESP MM4 XMM4 4 100	CH BP EBP MM5 XMM5 5 101	DH SI ESI MM6 XMM6 6 110	BH DI EDI MM7 XMM7 7 111
Effective Address	**Mod**	**R/M**	**Value of ModR/M Byte (in Hexadecimal)**							
[EAX]	00	000	00	08	10	18	20	28	30	38
[ECX]		001	01	09	11	19	21	29	31	39
[EDX]		010	02	0A	12	1A	22	2A	32	3A
[EBX]		011	03	0B	13	1B	23	2B	33	3B
[--][--][1]		100	04	0C	14	1C	24	2C	34	3C
disp32[2]		101	05	0D	15	1D	25	2D	35	3D
[ESI]		110	06	0E	16	1E	26	2E	36	3E
[EDI]		111	07	0F	17	1F	27	2F	37	3F
[EAX]+disp8[3]	01	000	40	48	50	58	60	68	70	78
[ECX]+disp8		001	41	49	51	59	61	69	71	79
[EDX]+disp8		010	42	4A	52	5A	62	6A	72	7A
[EBX]+disp8		011	43	4B	53	5B	63	6B	73	7B
[--][--]+disp8		100	44	4C	54	5C	64	6C	74	7C
[EBP]+disp8		101	45	4D	55	5D	65	6D	75	7D
[ESI]+disp8		110	46	4E	56	5E	66	6E	76	7E
[EDI]+disp8		111	47	4F	57	5F	67	6F	77	7F
[EAX]+disp32	10	000	80	88	90	98	A0	A8	B0	B8
[ECX]+disp32		001	81	89	91	99	A1	A9	B1	B9
[EDX]+disp32		010	82	8A	92	9A	A2	AA	B2	BA
[EBX]+disp32		011	83	8B	93	9B	A3	AB	B3	BB
[--][--]+disp32		100	84	8C	94	9C	A4	AC	B4	BC
[EBP]+disp32		101	85	8D	95	9D	A5	AD	B5	BD
[ESI]+disp32		110	86	8E	96	9E	A6	AE	B6	BE
[EDI]+disp32		111	87	8F	97	9F	A7	AF	B7	BF
EAX/AX/AL/MM0/XMM0	11	000	C0	C8	D0	D8	E0	E8	F0	F8
ECX/CX/CL/MM/XMM1		001	C1	C9	D1	D9	E1	E9	F1	F9
EDX/DX/DL/MM2/XMM2		010	C2	CA	D2	DA	E2	EA	F2	FA
EBX/BX/BL/MM3/XMM3		011	C3	CB	D3	DB	E3	EB	F3	FB
ESP/SP/AH/MM4/XMM4		100	C4	CC	D4	DC	E4	EC	F4	FC
EBP/BP/CH/MM5/XMM5		101	C5	CD	D5	DD	E5	ED	F5	FD
ESI/SI/DH/MM6/XMM6		110	C6	CE	D6	DE	E6	EE	F6	FE
EDI/DI/BH/MM7/XMM7		111	C7	CF	D7	DF	E7	EF	F7	FF

和16位寻址方式的几个区别：

a.32位的寄存器间接寻址可以使用除ESP外的所有通用寄存器，而不是16位时的四个寄存器（SI、DI、BX、BP）。

b.32位的寄存器相对寻址、基址变址寻址、相对基址变址寻址都可以使用除ESP外的所有的通用寄存器。

c.由于32位寻址方式添加了一位SIB字节（见表2-3），所以在基址变址寻址和相对基址变址方式时，可以使用一个新的特性：比例。一共有1、2、4、8四种比例方式。

参考文献：

1.《IBM-PC汇编语言程序设计》，沈美明，温冬蝉编著，清华大学出版社

上一篇 保护模式、实地址模式及V8086模式下的指令格式（上）
下一篇 控制台程序的标准句柄的重定向

猜你在找

C语言指针与汇编内存地址
C语言指针与汇编内存地址（二）
汇编语言视频课程
C语言指针与汇编内存地址－第3节课
C语言指针与汇编内存地址－第4节
实模式保护模式和虚拟8086方式
什么是实模式保护模式和虚拟8086方式
什么是实模式保护模式和虚拟8086方式
8086键盘输入实验《x86汇编语言从实模式到保护模
80x86保护模式之DOS与虚拟8086模式

查看评论

暂无评论

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide
Maemo Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase
Pure Solr Angular Cloud Foundry Redis Scala Django Bootstrap