

# 13015 计算机系统原理

## 第三章 课后习题 计算题讲解(部分)

讲师/公众号：**小飞学长Pro**

# 13015 计算机系统原理【第三章课后题】

3. 对于以下 AT&T 格式汇编指令，根据操作数的长度确定对应指令助记符中的长度后缀，并说明每个操作数的寻址方式。

- (1) `mov 8(%ebp, %ebx, 4), %ax`
- (2) `mov %al, 12(%ebp)`
- (3) `add ( , %ebx, 4), %ebx`
- (4) `or (%ebx), %dh`
- (5) `push $0xF8`
- (6) `mov $0xFFF0, %eax`
- (7) `test %cx, %cx`
- (8) `lea 8(%ebx, %esi), %eax`

**答：**根据【寄存器】来判断，无关是源还是目的

- 1) 后缀: w, 源: 基址+比例变址+偏移, 目: 寄存器
- 2) 后缀: b, 源: 寄存器, 目: 基址+偏移
- 3) 后缀: l, 源: 比例变址, 目: 寄存器
- 4) 后缀: b, 源: 基址, 目: 寄存器
- 5) 后缀: l, 源: 立即数, 目: 栈
- 6) 后缀: l, 源: 立即数, 目: 寄存器
- 7) 后缀: w, 源: 寄存器, 目: 寄存器
- 8) 后缀: l, 源: 基址+变址+偏移, 目: 寄存器

**举例：**8 (%ebp, %ebx, 4)

**偏移量** (基址寄存器, 变址寄存器, 比例因子)

# 13015 计算机系统原理【第三章课后题】

4. 使用汇编器处理以下各行 AT&T 格式代码时都会产生错误，请说明每一行存在什么错误。

- (1) `movl 0xFF, (%eax)`
- (2) `movb %ax, 12(%ebp)`
- (3) `addl %ecx, $0xF0`
- (4) `orw $0xFFFF0, (%ebx)`
- (5) `addb $0xF8, (%dl)`
- (6) `movl %bx, %eax`
- (7) `andl %esi, %esx`
- (8) `movw 8(%ebp, , 4), %ax`

答：

- 1) 源操作数是立即数 0xFF，需在前面加 '\$'
- 2) 源操作数为16位，而长度后缀是字节 'b'，不一致
- 3) 目的操作数不能是立即数寻址
- 4) 操作数位数超过16位，而长度后缀为16位的 'w'
- 5) 不能用8位寄存器作为目的操作数地址所在寄存器
- 6) 源操作数寄存器与目操作数寄存器长度不一致
- 7) 不存在ESX寄存器
- 8) 源操作数地址中缺少变址寄存器

**举例：** `8 (%ebp, %ebx, 4)`

**偏移量** (**基址寄存器**, **变址寄存器**, **比例因子**)

# 13015 计算机系统原理【第三章课后题】

5. 假设变量 `x` 和 `ptr` 的类型声明如下：

```
src_type x;  
dst_type * ptr;
```

这里，`src_type` 和 `dst_type` 是用 `typedef` 声明的数据类型。有以下几个 C 语言赋值语句：

```
* ptr = (dst_type) x;
```

若 `x` 存储在寄存器 `EAX` 或 `AX` 或 `AL` 中，`ptr` 存储在寄存器 `EDX` 中，则对于表 3.12 中给出的 `src_type` 和 `dst_type` 的类型组合，写出实现上述赋值语句的机器级代码。要求用 AT&T 格式表示机器级代码。

表 3.12 题 5 用表

src_type	dst_type	机器级表示
char	int	
int	char	
int	unsigned	
short	int	
unsigned char	unsigned	
char	unsigned	
int	int	

# 13015 计算机系统原理 【第三章课后题】

**x** 存储在寄存器 EAX 或 AX 或 AL 中

**ptr** 存储在寄存器 EDX 中

\* ptr = (dst\_type) x

序号	src_type	dst_type	机器级表示
1	char	int	<code>movsbl %al, %eax</code> <code>movl %eax, (%edx)</code>
2	int	char	<code>movb %al, (%edx)</code>
3	int	unsigned	<code>movl %eax, (%edx)</code>
4	short	int	<code>movswl %ax, %eax</code> <code>movl %eax, (%edx)</code>
5	unsigned char	unsigned	<code>movzbl %al, %eax</code> <code>movl %eax, (%edx)</code>
6	char	unsigned	<code>movsbl %al, %eax</code> <code>movl %eax, (%edx)</code>
7	int	int	<code>movl %eax, (%edx)</code>

扩展操作时，若被转换数据是无符号整型，则采用零扩展；若被转换数据是带符号整型，则采用符号扩展。例如，将一个 unsigned short 型或 unsigned char 型数据转换为 int 型时，采用的是零扩展。

**总结一句话**：源 → 目的

若小 → 大 (扩展)

源是**无符号** → **零扩展**

源是**有符号** → **符号扩展**

若大 → 小 (截断)

# 13015 计算机系统原理【第三章课后题】

7. 假设变量  $x$  和  $y$  分别存放在寄存器 EAX 和 ECX 中，给出以下每条指令执行后寄存器 EDX 中的结果。

(1) `leal (%eax), %edx`

(2) `leal 4(%eax, %ecx), %edx`

(3) `leal (%eax, %ecx, 8), %edx`

(4) `leal 0xc(%ecx, %eax, 2), %edx`

(5) `leal (, %eax, 4), %edx`

(6) `leal (%eax, %ecx), %edx`

答:

1)  $R[edx]=x$

2)  $R[edx]=x+y+4$

3)  $R[edx]=x+8*y$

4)  $R[edx]=y+2*x+12$       0xc是十六进制的12

5)  $R[edx]=4*x$

6)  $R[edx]=x+y$

举例:

`8 (%ebp, %ebx, 4) //  $R[ebp]+R[ebx]*4+8$`

偏移量 (基址寄存器, 变址寄存器, 比例因子)

谢谢大家