

应用型
特色教材

Win32

汇编语言程序设计

田民格 秦彩杰 吕良智◎主编



清华大学出版社

Win32

汇编语言程序设计



田民格 秦彩杰 吕良智◎主编

清华大学出版社
北京

内 容 简 介

本书共分为9章,具体介绍了汇编语言基础知识、Win32 汇编语言基本组成、Win32 汇编语言的编译运行、CPU 指令系统、FPU 指令系统、选择结构程序设计、循环结构程序设计、模块化程序设计和吾爱破解软件使用简介等内容。先详细介绍了汇编语言程序设计中要用到的相关理论基础和计算机的组成与工作原理,如数值与字符的表示方法、存储体系结构与工作原理等;再全面介绍了32位CPU指令与浮点指令;最后详尽介绍了汇编语言程序设计的方法,使读者可以把更多的精力用于汇编的程序设计上。本书力争做到,初学者用C语言能实现的功能,用本书的32位汇编语言也能实现。

本书所有例子的源程序都是完整的,并有详细的注释,且都能在Windows XP和Windows 7环境上编译运行通过,特别适合初学者。本书有配套的考试系统和相应的题库,既可作为标准化试题(单项选择题、多项选择题、判断题和填空题)的测评,又可作为程序设计题(程序改错题、程序填空题和程序设计题)的测评。

本书可作为应用型本科院校计算机类专业《Win32 汇编语言程序设计》的教材,也可作为《计算机组成原理与汇编语言(Win32)》的教材,适合既要求有一定理论知识,又要求突出实践的院校,尤其适合强调提高编程能力的院校和读者。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Win32 汇编语言程序设计/田民格,秦彩杰,吕良智主编. —北京:清华大学出版社,2017
ISBN 978-7-302-47694-8

I. ①W… II. ①田… ②秦… ③吕… III. ①汇编语言-程序设计 IV. ①TP313

中国版本图书馆CIP数据核字(2017)第155838号

责任编辑:苏明芳
封面设计:刘超
版式设计:魏远
责任校对:马子杰
责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62788903

印 装 者:北京泽宇印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:16.75 字 数:413千字

版 次:2017年8月第1版 印 次:2017年8月第1次印刷

印 数:1~2500

定 价:45.00元

产品编号:073830-01

前 言

Foreword

随着计算机技术的发展, 学生要学习的课程越来越多, 每门课的学时越来越少, 而汇编语言从 16 位过渡到 32 位, 指令越来越多, 如何在有限的时间内学到尽可能多的知识, 特别是如何提高汇编语言的编程能力, 压力越来越大。于是, 我们结合多年的教学与软件开发的经验, 编写了本书。

本书虽详尽介绍了 CPU 指令和 FPU 指令等, 但并不是每条指令都需要掌握, 因为实际程序设计过程中, 往往可能只需要掌握其中 20% 的指令, 就能解决现实工作中 80% 的问题, 而剩余 20% 的问题, 可以根据具体的需要, 查阅具体的指令。因此, 在教学过程中, 可根据具体的情况, 选修相关内容。

作者建议, 本书按 12 个主题进行教与学, 每个主题讲授与训练约 3 个学时, 课堂测试与讲评约 1 个学时, 合计 48 学时。

序 号	主 题	主 要 内 容
1	数据类型	Win32 汇编语言程序结构基本组成, 各种数据类型数据的定义与输入输出
2	MASM 整数+-*/%	简单整数数据传送指令, 整数的加减乘除余, 实现整数表达式的计算
3	MASM 实数+*/	浮点数数据传送指令, 浮点数的加减乘除, 实现浮点数表达式的计算
4	MASM 函数	算术指令, 如 FSQRT、FSIN、FCOS、FABS 等函数的使用
5	选择结构	实现整数(含字符)大小和浮点数大小的比较
6	循环结构	.while 和.repeat 循环及 LOOP 循环指令, 实现各种循环重复操作
7	C 嵌入汇编	串操作指令结合 C 嵌入汇编, 在 C 中实现各种字符串功能
8	子程序(函数)	子程序(函数)的定义与调用, 实现模块化程序设计
9	递归程序设计	汇编实现递归程序设计及 C 与汇编相互调用、函数重载
10	软件破解	OD 实现逆向工程, 破解其他开发工具生成的.exe 文件的登录密码
11	浮点数表示	计算机浮点数的表示等, 实现浮点数与机器码的相互转换, 其他编码转换
12	CPU 与存储器连接	32 位 CPU 体系结构、存储访问工作原理等

通过以上 12 讲的教学, 学生在 48 学时下就能具有比较强的汇编编程能力, 又能初步了解计算机的组成与工作原理, 因此, 本教材也可作为《计算机组成原理与汇编语言(Win32)》的教材。

本书由田民格、秦彩杰、吕良智担任主编, 其中秦彩杰编写第 1、2、3 章, 田民格编写第 4、5、6 章, 吕良智编写第 7、8、9 章和附录。

本书配套的电子课件和源程序等资料可登录作者的 FTP 下载，网址为 FTP://218.5.241.13，用户名为 tmg，密码为 123456。本书资料在 masm 文件夹中，其中“组成与汇编单机版.rar”是本书单机版考试系统，根目录下的 ksxt.exe 是本书网络版考试系统。

为使本书篇幅不致过长，其中的案例是根据作者多年教学经验归纳、总结、设计的，文字表述也是经过再三斟酌的，插图也是经过专门加工的。在使用本书的过程中，若发现有任何问题，可与作者进行沟通联系，以使本书臻于完善。作者 E-mail 为 TmgDelphi@163.com。

编 者
2017.7

目 录

Contents

第 1 章 汇编语言基础知识	1
1.1 计数制	1
1.1.1 十进制 (Decimal)	1
1.1.2 二进制 (Binary)	1
1.1.3 八进制 (Octal)	2
1.1.4 十六进制 (HexaDecimal)	2
1.2 进制数间的转换	3
1.2.1 十进制转二进制	3
1.2.2 十进制转八进制和十六进制	4
1.2.3 十进制转二进制加法口算方法	4
1.2.4 十进制转二进制减法口算方法	5
1.2.5 十进制转二进制其他口算方法	6
1.3 计算机计量单位	6
1.3.1 计算机存储容量计量单位	6
1.3.2 计算机时钟周期计量单位	6
1.4 数值数据的表示	7
1.4.1 无符号整数的表示	7
1.4.2 有符号整数的表示	7
1.4.3 移码	8
1.4.4 BCD 码	9
1.4.5 浮点数	9
1.5 字符数据的表示	12
1.5.1 ASCII 码	12
1.5.2 机内码	12
1.5.3 Unicode	13
1.5.4 UTF-8	14
习题 1	14
第 2 章 Win32 汇编语言基本组成	16
2.1 程序结构	16
2.1.1 处理器选择伪指令	17

2.1.2	.model 伪指令	18
2.1.3	指明是否区分大小写	18
2.1.4	要引用的头文件和库文件	18
2.1.5	函数原型 PROTO 声明	19
2.1.6	函数原型 EXTRN 声明	19
2.1.7	变量的定义及使用	20
2.1.8	数据段和代码段的定义	21
2.1.9	注释	22
2.1.10	指令、标号和分行	22
2.1.11	invoke 伪指令调用函数	23
2.1.12	Call 指令调用函数	23
2.1.13	函数调用返回值	24
2.1.14	函数的定义	24
2.1.15	局部变量的定义	26
2.1.16	程序结束	27
2.1.17	汇编结束	27
2.2	数据类型	27
2.2.1	整数	27
2.2.2	整数常量表达式	28
2.2.3	浮点数	30
2.2.4	字符和字符串	33
2.2.5	结构体	34
	习题 2	36
第 3 章	Win32 汇编语言的编译运行	40
3.1	配置编译链接环境	40
3.1.1	配置 VC6.0 环境	41
3.1.2	MASM32 的安装	41
3.1.3	配置 MASM32 环境	44
3.1.4	通过注册表配置 VC 和 MASM32 环境	44
3.2	命令提示符下编译链接和运行	44
3.3	VC 环境下编译链接和运行	46
3.4	C/C++ 嵌入汇编指令	48
3.4.1	汇编指令访问 C 整型变量	48
3.4.2	汇编指令读取 C 整型数组元素	49
3.4.3	汇编指令写入 C 字符数组	50
3.5	C 程序反汇编生成汇编源程序	51
3.5.1	C 程序编译时生成汇编语言源程序	51
3.5.2	修改 C 程序反汇编生成的汇编源程序	52

习题 3	55
第 4 章 CPU 指令系统	57
4.1 系统结构	58
4.2 80386 微处理器结构	59
4.3 CPU 寄存器	61
4.3.1 16 位寄存器组	61
4.3.2 32 位寄存器组	61
4.3.3 标志寄存器 EFlags	62
4.4 80X86 处理器工作模式	64
4.5 存储器	65
4.6 操作数寻址方式	67
4.7 数据传送类指令	71
4.7.1 通用数据传送 MOV/MOV[SI]X	71
4.7.2 数据交换 XCHG	73
4.7.3 字节查表转换 XLAT[B]	75
4.7.4 字节反向存储 BSWAP	76
4.7.5 入栈 PUSH/PUSHA[D]	76
4.7.6 出栈 POP/POPA[D]	77
4.7.7 取地址 LEA/L[DEFGS]S	77
4.7.8 EFlags 低 8 位与 AH 传送 LAHF/SAHF	78
4.7.9 EFlags 出入栈 PUSHF[D]/POPF[D]	79
4.7.10 进位位 CF 操作 CLC/STC/CMC	79
4.7.11 方向位 DF 操作 CLD/STD	79
4.7.12 中断允许位 IF 操作 CLI/STI	79
4.8 整数算术运算指令	79
4.8.1 加法 ADD/ADC/INC/XADD	80
4.8.2 减法 SUB/SBB/DEC/NEG	82
4.8.3 乘法 MUL/IMUL	82
4.8.4 除法 DIV/IDIV	83
4.8.5 符号扩展 CBW/CWD/CDQ	85
4.8.6 整数比较 CMP/CMPXCHG[8B]	86
4.9 调整指令 (实现大数运算)	87
4.9.1 数字字符加法调整 AAA	87
4.9.2 数字字符减法调整 AAS	89
4.9.3 二进制编码调整为 BCD 码 AAM	90
4.9.4 BCD 码调整为二进制编码 AAD	90
4.9.5 BCD 码加法调整 DAA	91
4.9.6 BCD 码减法调整 DAS	92

4.10	逻辑运算指令	93
4.10.1	逻辑与操作 AND	93
4.10.2	逻辑或操作 OR	94
4.10.3	逻辑非操作 NOT	94
4.10.4	逻辑异或操作 XOR	94
4.10.5	逻辑比较测试 TEST	95
4.11	位操作指令	96
4.11.1	算术移位 SAL/SAR	96
4.11.2	逻辑移位 SHL/SHR	97
4.11.3	双精度移位 SHLD/SHRD	98
4.11.4	不带进位循环移位 ROL/ROR	100
4.11.5	带进位循环移位 RCL/RCR	100
4.11.6	位扫描 BSF/BSR	101
4.11.7	第 i 位操作 BT[CRS]	103
4.12	串操作指令	104
4.12.1	重复串操作 REP[E Z NE NZ]	105
4.12.2	移串操作 MOVS[B W D]	106
4.12.3	取串操作 LODS[B W D]	108
4.12.4	存串操作 STOS[B W D]	109
4.12.5	输入串操作 INS[B W D]	110
4.12.6	输出串操作 OUTS[B W D]	111
4.12.7	串扫描操作 SCAS[B W D]	111
4.12.8	串比较操作 CMPS[B W D]	112
4.13	CPU 控制指令	114
4.13.1	空操作指令 NOP	114
4.13.2	等待指令 WAIT	115
4.13.3	暂停指令 HLT	115
4.13.4	封锁数据指令 LOCK	115
4.13.5	获得 CPU 信息 CPUID	115
4.13.6	读时间戳计数器 RDTSC	116
	习题 4	117
第 5 章	FPU 指令系统	127
5.1	FPU 寄存器	127
5.1.1	浮点数据寄存器	127
5.1.2	浮点标记寄存器	128
5.1.3	浮点状态寄存器	128
5.1.4	浮点控制寄存器	130
5.2	FPU 指令系统的约定	131

5.3	实数传送指令	132
5.4	实数常量加载指令	136
5.5	实数比较指令	136
5.6	实数加法指令	143
5.7	实数减法指令	144
5.8	实数乘法指令	146
5.9	实数除法指令	147
5.10	算术指令	148
5.11	FPU 控制指令	161
5.11.1	初始化 FPU 操作 F[N]INIT	161
5.11.2	保存状态字 F[N]STSW	161
5.11.3	保存控制字 F[N]STCW	161
5.11.4	加载控制字 FLDCW	161
5.11.5	清除异常 F[N]CLEX	161
5.11.6	保存环境 F[N]STENV	161
5.11.7	加载环境 FLDENV	163
5.11.8	存环境与数据 F[N]SAVE	163
5.11.9	读环境与数据 FRSTOR	165
5.11.10	增加 FPU 栈指针指令 FINCSTP	166
5.11.11	减少 FPU 栈指针 FDECSTP	166
5.11.12	st(i)清空 FFREE st(i)	167
5.11.13	FPU 空操作 FNOP	168
5.11.14	同步 FPU 与 CPU 指令[F]WAIT	168
	习题 5	168
第 6 章	选择结构程序设计	171
6.1	.IF 伪指令实现双分支选择	171
6.2	.IF...ELSEIF 伪指令实现多分支选择	177
6.3	JMP 和 Jcc 转移指令	179
6.4	测试条件转存指令 SETcc	185
6.5	浮点数的大小比较	187
6.6	散转程序设计	190
	习题 6	195
第 7 章	循环结构程序设计	200
7.1	当循环伪指令 .while	200
7.2	重复伪指令 .repeat	202
7.3	退出伪指令 .break	205
7.4	短路伪指令 .continue	205
7.5	循环指令 LOOP	206

7.6	ECX 为零转移指令 JECXZ	211
7.7	LOOP/JECXZ 循环指令存在的问题	212
	习题 7	214
第 8 章	模块化程序设计	223
8.1	子程序的定义	223
8.1.1	子程序的属性	224
8.1.2	局部变量的定义	224
8.2	子程序的调用与返回	225
8.2.1	子程序调用 CALL 或 INVOKE	225
8.2.2	返回指令 RET	226
8.3	不同数据类型作形参时的传递方法	226
8.3.1	整数参数的传递	226
8.3.2	字符参数的传递	227
8.3.3	整型数组参数的传递	228
8.3.4	字符串参数的传递	231
8.3.5	双精度浮点数参数的传递	232
8.4	汇编语言递归函数求累加和	233
8.4.1	C 语言实现递归函数求累加和	233
8.4.2	汇编语言实现递归函数求累加和	234
8.4.3	C 程序反汇编得到的源程序实现递归求和	236
8.5	C 程序调用汇编语言的子程序	238
8.5.1	C 程序调用汇编语言子程序的方式	238
8.5.2	C 整型参数不同调用方式传入汇编	239
8.5.3	C 整型数组参数传入汇编	241
8.5.4	C 字符数组参数传入汇编	242
8.5.5	C 调用汇编语言实现函数重载	242
	习题 8	243
第 9 章	吾爱破解软件使用简介	248
9.1	吾爱破解软件主线程模块界面	248
9.2	如何用吾爱破解软件找到登录软件密码	249
9.3	用吾爱破解软件修改可执行文件并保存	250
附录 A	252	
参考文献	255	

第 1 章

汇编语言基础知识

本章主要介绍数值数据和字符数据在计算机内部的表示方法,包括各种进制数及相关计量单位;数值数据详细介绍整数和浮点数(实数)的表示方法及显示方法,字符数据详细介绍 ASCII 码、机内码、Unicode 编码和 UTF-8 编码的表示方法及显示方法。通过本章的学习,读者应该完成以下学习目标:

- (1) 掌握十进制、二进制、十六进制的表示方法及其相互转换方法。
- (2) 了解八进制和二进制口算方法。
- (3) 掌握无符号整数和有符号整数的表示方法。
- (4) 掌握移码的表示方法以及了解 BCD 码的表示方法。
- (5) 掌握浮点数(实数)的表示方法。
- (6) 了解 ASCII 码常用字符的编码规律。
- (7) 了解机内码和区位码的编码规律及其相互转换方法。
- (8) 了解 Unicode 和 UTF-8 的编码规律及其相互转换方法。

1.1 计 数 制

1.1.1 十进制 (Decimal)

十进制用 0~9 共 10 个数码来表示,其基数为 10,运算规则为逢十进一,各位权是 10^i (整数部分最低位位权为 10^0 ,其余各位位权从右至左指数依次增大;小数部分最高位位权为 10^{-1} ,其余各位位权从左至右指数依次减小),十进制数后缀字母为 D,一般省略, n 位整数 m 位小数十进制数 N 表示如下:

$$N = a_{n-1}a_{n-2}\cdots a_0.a_{-1}\cdots a_{-m}D$$
$$= a_{n-1} \times 10^{n-1} + a_{n-2} \times 10^{n-2} + \cdots + a_0 \times 10^0 + a_{-1} \times 10^{-1} + \cdots + a_{-m} \times 10^{-m}$$

例如:

$$N = 325.46D$$
$$= 3 \times 10^2 + 2 \times 10^1 + 5 \times 10^0 + 4 \times 10^{-1} + 6 \times 10^{-2}$$
$$= 325.46$$

1.1.2 二进制 (Binary)

二进制用 0 和 1 两个数码来表示,其基数为 2,运算规则为逢二进一,各位权是 2^i ,

二进制数后缀字母为 B，n 位整数 m 位小数的二进制数 N 表示如下：

$$N = a_{n-1}a_{n-2}\cdots a_0.a_1\cdots a_mB$$

$$= a_{n-1}\times 2^{n-1} + a_{n-2}\times 2^{n-2} + \cdots + a_0\times 2^0 + a_1\times 2^{-1} + \cdots + a_m\times 2^{-m}$$

例如：

$$N = 1011\ 1101.11B$$

$$= 1\times 2^7 + 0\times 2^6 + 1\times 2^5 + 1\times 2^4 + 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 1\times 2^{-2}$$

$$= 1\times 128 + 0\times 64 + 1\times 32 + 1\times 16 + 1\times 8 + 1\times 4 + 0\times 2 + 1\times 1 + 1\times 0.5 + 1\times 0.25$$

$$= 189.75$$



注意

(1) 二进制整数部分各位的位权，特别是后 4 位的位权分别是 8、4、2 和 1，这是常用的二进制数口算数值，如 $1111B = 8 + 4 + 2 + 1 = 15$ 。

(2) 非十进制数不能读成几百几十，例如， $111B$ 不能读成“一百一十一”，只能读成“一一一 B”，其他非十进制数读法类似。

(3) 无论是数值还是字符，甚至是音频视频等信息，在计算机底层都是用二进制表示和存储的，只是为了阅读或使用方便，应用软件都把它们转换为十进制或十六进制表示和显示。因此，我们使用计算机时一般看不到二进制数。

1.1.3 八进制 (Octal)

八进制用 0~7 共 8 个数码来表示，其基数为 8，运算规则为逢八进一，各位权是 8^i ，八进制数后缀字母为 O 或 Q，n 位整数 m 位小数的八进制数 N 表示如下：

$$N = a_{n-1}a_{n-2}\cdots a_0.a_1\cdots a_mQ$$

$$= a_{n-1}\times 8^{n-1} + a_{n-2}\times 8^{n-2} + \cdots + a_0\times 8^0 + a_1\times 8^{-1} + \cdots + a_m\times 8^{-m}$$

例如：

$$N = 377.34Q$$

$$= 3\times 8^2 + 7\times 8^1 + 7\times 8^0 + 3\times 8^{-1} + 4\times 8^{-2}$$

$$= 255.4375$$



注意

(1) 在 C 语言（本书所提到的 C 语言多指 VC 下的 C++）中八进制数用前缀 0（数字零）表示，如 0175。

(2) 八进制相对比较少用，了解即可。

1.1.4 十六进制 (HexaDecimal)

十六进制用 0~9、A、B、C、D、E 和 F 共 16 个数码（字母一般不区分大小写）来表示，A~F 依次相当于十进制数的 10~15，十六进制基数为 16，运算规则为逢十六进一，各位权是 16^i ，十六进制数后缀字母为 H，n 位整数 m 位小数的十六进制数 N 表示如下：

$$N = a_{n-1}a_{n-2}\cdots a_0.a_{-1}\cdots a_{-m}H$$

$$= a_{n-1} \times 16^{n-1} + a_{n-2} \times 16^{n-2} + \cdots + a_0 \times 16^0 + a_{-1} \times 16^{-1} + \cdots + a_{-m} \times 16^{-m}$$

例如:

$$N = FA.8H$$

$$= 15 \times 16^1 + 10 \times 16^0 + 8 \times 16^{-1}$$

$$= 250.5$$



注意

(1) 在汇编指令中,若十六进制数的第一个数字是字母,则必须前缀 0(数字零),以区别于标识符(变量名等),如“MOV EAX,0FAH”,平时书写可不必加前缀 0。

(2) 在 C 语言和吾爱破解等软件中十六进制数用前缀 0x(数字零和字母 x)表示,如 0xFA。

1.2 进制数间的转换

1.2.1 十进制转二进制

将十进制转换为二进制,整数部分和小数部分分别进行转换,然后再合并。

整数和小数转换规则如下。

(1) 整数部分:除 2 取余,先低(位)后高(位)。

(2) 小数部分:乘 2 取整,先高(位)后低(位)。

例 1-1 将十进制数 123.6875 转为二进制数。

<p>整数部分</p> $\begin{array}{r} 2 \overline{) 123} \\ \underline{24} \\ 2 \overline{) 61} \\ \underline{42} \\ 2 \overline{) 19} \\ \underline{18} \\ 2 \overline{) 15} \\ \underline{14} \\ 2 \overline{) 11} \\ \underline{10} \\ 2 \overline{) 1} \\ \underline{0} \end{array}$	<p>1 高位</p> <p style="font-size: 2em;">↑</p> <p>0</p> <p style="font-size: 2em;">↑</p> <p>1 1 1 1</p> <p>1 低位</p>	<p>小数部分</p> $\begin{array}{r} 0.6875 \\ \times 2 \\ \hline 1.3750 \\ \times 2 \\ \hline 0.7500 \\ \times 2 \\ \hline 1.5000 \\ \times 2 \\ \hline 1.0000 \end{array}$	<p>1 高位</p> <p style="font-size: 2em;">↓</p> <p>0</p> <p style="font-size: 2em;">↓</p> <p>1 1 1</p> <p>1 低位</p>
--	---	---	---

$$123 = 1111011B$$

$$0.6875 = 0.1011B$$

$$123.6875 = 1111011.1011B$$

例 1-2 将十进制数 1020.6 转为二进制数。



说明

将十进制数 0.6 转为二进制数,其结果是无限循环小数,默认按就近舍入(Round to Nearest)的方法处理其尾数,具体规定详见浮点控制寄存器 RC(Rounding Control)字段的说明。

整数部分:

2	1	0	2	0					
2	5	1	0	0	高位
2	2	5	5	0	
2	1	2	7	1	
2	6	3	1	
2	3	1	1	
2	1	5	1	
2	7	1	
2	3	1	
2	1	1	
2	0	1	低位

小数部分:

0.	6	
×	2	
1.	2	循环小数
×	2 1 高位
0.	4 0
×	2 0
0.	8 0
×	2 1
1.	6 1
×	2 1 低位
1.	2 1 低位

$$1020=11\ 1111\ 1100B$$

$$0.6=0.\dot{1}00\dot{1}B$$

$$1020.6=11\ 1111\ 1100.\dot{1}00\dot{1}B$$

1.2.2 十进制转八进制和十六进制

十进制转换为八或十六进制规则如下。

(1) 整数部分: 除八或十六取余, 先低后高。

(2) 小数部分: 乘八或十六取整, 先高后低。

由于数字太大, 十进制转八进制或十六进制常先转换为二进制再转换为八进制或十六进制, 规则如下。

整数部分:

(1) 将二进制整数从右往左每 3 位一组 (不够时左边补 0) 分别转换为八进制数。

(2) 将二进制整数从右往左每 4 位一组 (不够时左边补 0) 分别转换为十六进制数。

小数部分:

(1) 将二进制小数从左往右每 3 位一组 (不够时右边补 0) 分别转换为八进制数。

(2) 将二进制小数从左往右每 4 位一组 (不够时右边补 0) 分别转换为十六进制数。

例 1-3 1111111100.1001B 分别转换为八进制和十六进制。

$$1111111100.1001B=001\ 111\ 111\ 100.100\ 100B=1774.44Q$$

$$=0011\ 1111\ 1100.1001B=3FC.9H$$

1.2.3 十进制转二进制加法口算方法

首先, 二进制数各位的位权 2^n 与十进制数的对应关系如下:

$$2^0=1 \quad 2^5=32 \quad 2^{10}=1024 \quad 2^{15}=32768$$

$$2^1=2 \quad 2^6=64 \quad 2^{11}=2048 \quad 2^{16}=65536$$

$$2^2=4 \quad 2^7=128 \quad 2^{12}=4096$$

$$2^3=8 \quad 2^8=256 \quad 2^{13}=8192$$

$$2^4=16 \quad 2^9=512 \quad 2^{14}=16384$$

其次, 观察以下两种进制数的规律:

$$\begin{array}{ll}
 10^1=10 & 2^1=10\text{B} \\
 10^2=100 & 2^2=100\text{B} \\
 10^3=1000 & 2^3=1000\text{B} \\
 \dots & \dots \\
 10^n=10\cdots 0 \text{ (n 个 0)} & 2^n=10\cdots 0\text{B (n 个 0)}
 \end{array}$$

由此可得二进制加法口算方法： 2^n 由 1 个 1 和 n 个 0 构成的二进制数组成，即：

$$2^n = \overbrace{10\cdots\cdots 0}^n \text{B}$$

因此，比该数略大的数只需将若干相应位权的 0 改为 1 即可。如：

$$\begin{aligned}
 \because 128 &= 2^7 = \overbrace{1000\ 0000}^7 \text{B} \\
 \therefore 130 &= 128 + 2 = 2^7 + 2 = 1000\ 0000\text{B} + 10\text{B} = 1000\ 0010\text{B} \\
 \because 256 &= 2^8 = \overbrace{10000\ 0000}^8 \text{B} \\
 \therefore 261 &= 256 + 5 = 2^8 + (4+1) = 1\ 0000\ 0000\text{B} + 101\text{B} = 1\ 0000\ 0101\text{B}
 \end{aligned}$$

同理：

$$\begin{aligned}
 525 &= 512 + 13 = 2^9 + (8+4+1) = 10\ 0000\ 0000\text{B} + 1101\text{B} = 10\ 0000\ 1101\text{B} \\
 1030 &= 1024 + 6 = 2^{10} + (4+2) = 100\ 0000\ 0000\text{B} + 110\text{B} = 100\ 0000\ 0110\text{B}
 \end{aligned}$$

1.2.4 十进制转二进制减法口算方法

首先，观察以下两种进制数的规律：

$$\begin{array}{ll}
 10^1-1=9 & 2^1-1=1\text{B} \\
 10^2-1=99 & 2^2-1=11\text{B} \\
 10^3-1=999 & 2^3-1=111\text{B} \\
 \dots & \dots \\
 10^n-1=9\cdots 9 \text{ (n 个 9)} & 2^n-1=1\cdots 1\text{B (n 个 1)}
 \end{array}$$

由此可得二进制减法口算方法： 2^n-1 由 n 个二进制 1 组成，即：

$$2^n - 1 = \overbrace{1\cdots\cdots 1}^n \text{B}$$

因此，比该数略小的数只需将若干相应位权的 1 改为 0 即可。如：

$$\begin{aligned}
 \because 127 &= 2^7 - 1 = \overbrace{111\ 1111}^7 \text{B} \\
 \therefore 120 &= 127 - 7 = (2^7 - 1) - (4+2+1) = 111\ 1111\text{B} - 111\text{B} = 111\ 1000\text{B} \\
 \because 255 &= 2^8 - 1 = \overbrace{1111\ 1111}^8 \text{B} \\
 \therefore 250 &= 255 - 5 = (2^8 - 1) - (4+1) = 1111\ 1111\text{B} - 101\text{B} = 1111\ 1010\text{B}
 \end{aligned}$$

同理：

$$\begin{aligned}
 505 &= 511 - 6 = (2^9 - 1) - (4+2) = 1\ 1111\ 1111\text{B} - 110\text{B} = 1\ 1111\ 1001\text{B} \\
 1020 &= 1023 - 3 = (2^{10} - 1) - (2+1) = 11\ 1111\ 1111\text{B} - 11\text{B} = 11\ 1111\ 1100\text{B}
 \end{aligned}$$

1.2.5 十进制转二进制其他口算方法

(1) 乘法口算方法：若一个大整数 N 和一个小整数 a 可以表示成：

$$N=a \times 2^n$$

将 N 转换成二进制数，则可先将小整数 a 转换成二进制数，然后右边再补 n 个 0 即可，相当于将其左移 n 位或小数点右移 n 位。如：

$$40=10 \times 2^2=(8+2) \times 2^2=1010\text{B} \times 2^2=1010\ 00\text{B}$$

$$88=11 \times 2^3=(8+2+1) \times 2^3=1011\text{B} \times 2^3=1101\ 000\text{B}$$

(2) 除法口算方法：若一个分数 F 和一个整数 a 可以表示成：

$$F=a/2^n$$

若将 F 转换成二进制数，则可先将整数 a 转换成二进制数，然后将其右移 n 位或小数点左移 n 位。如：

$$11/16=(8+2+1) \times 2^{-4}=1011\text{B} \times 2^{-4}=0.1011\text{B}$$

$$23/64=(16+4+2+1) \times 2^{-6}=10111\text{B} \times 2^{-6}=0.010111\text{B}$$

(3) 其他口算：若整数 $N=a \times 16^n$ ，则可先将其转换成十六进制数再转换成二进制数，如：

$$161=10 \times 16^1+1 \times 16^0=0\text{A}1\text{H}=1010\ 0001\text{B}$$

$$176=11 \times 16^1+0 \times 16^0=0\text{B}0\text{H}=1011\ 0000\text{B}$$

1.3 计算机计量单位

1.3.1 计算机存储容量计量单位

计算机存储容量常用计量单位如下：

$$1\text{Byte}(\text{字节})=8\text{bit}(\text{二进制位, 比特})$$

$$1024\text{B}=2^{10}\text{B}=1\text{KB}$$

$$1024\text{TB}=2^{50}\text{B}=1\text{PB}$$

$$1024\text{KB}=2^{20}\text{B}=1\text{MB}$$

$$1024\text{PB}=2^{60}\text{B}=1\text{EB}$$

$$1024\text{MB}=2^{30}\text{B}=1\text{GB}$$

$$1024\text{EB}=2^{70}\text{B}=1\text{ZB}$$

$$1024\text{GB}=2^{40}\text{B}=1\text{TB}$$

$$1024\text{ZB}=2^{80}\text{B}=1\text{YB}$$

一般个人计算机存储容量达到 TB 就已经很大了，但有些大数据公司，其每天的数据处理量就能达到 100PB。

1.3.2 计算机时钟周期计量单位

计算机时钟周期常用计量单位如下：

$$1\text{ms}=10^{-3}\text{s}$$

$$1\mu\text{s}=10^{-6}\text{s}$$

$$1\text{ns}=10^{-9}\text{s}$$

$$1\text{ps}=10^{-12}\text{s}$$

$$1\text{fs}=10^{-15}\text{s}$$

$$1\text{as}=10^{-18}\text{s}$$