

洪岷生 叶仰明 蔡经球 等

C语言及其开发工具

— C语言及编译器

(上)

73.874
29-087

814323-2

58

7887

88 87

C语言及其开发工具

——C语言及编译器



(上册)

洪岷生 叶仰明 蔡经球等 编译

厦门大学出版社

1988. 3

C 语言及其开发工具

——C 语言及编译器

(上)

洪岷生 叶仰明 蔡经球等 编译

厦门大学出版社出版

福建省新华书店发行

三明市印刷厂印刷

开本 787×1092 1/32 19.875印张 483千字

1988年8月第1版 1988年8月第1次印刷

印数：1—1,000册

ISBN 7-5615-0067-9

TP·1 定价：3.95元

序 言

C语言目前在软件工程领域里引起人们极大的关注，它不仅成为各种系统软件和大量应用软件优先考虑使用的程序设计语言，而且成为从微机到大型机都已广泛实现的语言。因为C语言将高级语言和汇编语言两者的特点有机地结合在一起，具有丰富的运算符和灵活实用的表达式，并采用现代的数据结构和先进的控制流，所以它具有语言简洁、表达力强、代码质量高和移植性好等优点。

Microsoft公司1986年推出的Microsoft C4.0编译器适用于运行MS-DOS（即PC-DOS）操作系统的各种PC微机（IBM PC、PC/XT、PC AT、长城0520、长城286以及各种IBM PC兼容机和非兼容机），它不仅实现了B.W. Kernighan和D.M. Ritchie在《C程序设计语言》一书中所描述的C语言的全部功能，而且具有下列特点：

1. 与286 XENIX操作系统和正在发展的ANSI（美国国家标准局）C标准兼容。
2. 具有先进的优化能力，可自动生成效率高和质量好的代码。
3. 提供五种标准内存模式（小型、紧凑型、中型、大型和巨型），以便设计各种规模的程序。此外，允许用户灵活设计其他各种专用内存模式的程序。
4. 提供200多个库例程序，其中多数与XEMX库例程序兼容，而且完全实现了UNIX SYSTEM V的库函数。
5. 对浮点运算提供全面的支持。
6. 支持多种语言的混合程序设计，可跟Microsoft FORTRAN、Pascal和宏汇编生成的目标模块互相连接。
7. 支持文件共享和文件锁存，从而支持MS-DOS网络及IBM局部网络。
8. 有增强的语法检查机制，并提供充分的错误提示和警告信息。

此外，Microsoft C编译软件包还提供多种程序开发和维护工具，主要有以下几种：

1. LINK（复盖连接程序）可以把多个目标文件连接成单一的可执行文件，并可生成复盖程序。
2. LIB（库管理程序）可以生成、组织和维护各种运行库。
3. MAKE（程序维护实用程序）可以自动维护各种高级语言程序。
4. Code View（窗口符号调试器）可以用鼠标器或键盘命令来调试C语言或汇编语言程序。

鉴于Microsoft C4.0编译器及其配套工具的先进性和实用性，它们在软件开发中可以发挥很大作用。因此我们特编译此书，供软件专业师生和软件工作者做为教学参考书和设计工具书使用。

本书主要根据下列资料编译而成：

- Microsoft C Language Reference
- Microsoft C Compiler User's Guide
- Microsoft C Run-Time Library Reference
- Microsoft Code View

除了对原资料中发现的错误加以更正外，我们还根据最新版本的Microsoft C编译软件包提供的README.DOC文件对全书内容进行修订。

全书内容共有四篇，分上、下两册出版。这四篇的内容简述如下：

- 第一篇“Microsoft C语言”，叙述Microsoft公司实现的C语言的功能与定义。
- 第二篇“Microsoft C编译器”，介绍Microsoft C编译器的功能，叙述如何编译、连接和运行各种C程序。此外，本篇还提供多种维护和开发工具的使用说明。
- 第三篇“窗口符号调试器Code View”，介绍新型的多窗口多功能的符号调试器Code View的使用方法。
- 第四篇“Microsoft C运行库”，第一部分概述C运行库的使用与分类，第二部分按字母顺序详细叙述运行库的200多个例行程序的功能。

本书由洪岷生、叶仰明、吴克西、蔡经球、廖泉文、李茂青同志共同编译，由洪岷生同志最后定稿。

由于水平有限，错误和缺点在所难免，请广大读者批评指正。

编者

一九八七年八月

内 容 简 介

本书由Microsoft 公司出版的Microsoft C语言及其开发工具的使用说明等资料编译而成。较全面地叙述了Microsoft C语言的功能及有关软件开发工具，语言功能较齐全，所提供的工具先进且实用。全书由四个部分组成，上册主要内容为：Microsoft 公司实现的C语言的功能与定义；C编译器的功能以及各种维护和开发工具的使用说明等。不仅可作为高等院校计算机软件专业的教材，也适于从事软件工作的各层次软件人员参考。

目 录

第一篇 Microsoft C 语言

第一章 引言	(I - 1)
第二章 C 的基本成分	(I - 3)
2.1 概述	(I - 3)
2.2 字符集	(I - 3)
2.2.1 字母和数字	(I - 3)
2.2.2 空白符	(I - 4)
2.2.3 标点符号和特殊符号	(I - 4)
2.2.4 换码串	(I - 4)
2.2.5 运算符	(I - 5)
2.3 常量	(I - 6)
2.3.1 整数常量	(I - 6)
2.3.2 浮点常量	(I - 7)
2.3.3 字符常量	(I - 8)
2.3.4 字符串常量	(I - 8)
2.4 标识符	(I - 9)
2.5 关键字	(I - 10)
2.6 注释	(I - 10)
2.7 标识单元	(I - 11)
第三章 程序结构	(I - 12)
3.1 概述	(I - 12)
3.2 源程序	(I - 12)
3.3 源文件	(I - 13)
3.4 程序执行	(I - 14)
3.5 生存期与可见性	(I - 14)
3.6 命名类	(I - 16)
第四章 说明	(I - 18)
4.1 概述	(I - 18)
4.2 类型设定符	(I - 18)
4.3 说明符	(I - 21)
4.3.1 指针、数组和函数的说明符	(I - 21)
4.3.2 复杂说明符	(I - 21)
4.3.3 带特殊关键字的说明符	(I - 23)
4.4 变量说明	(I - 24)
4.4.1 简单变量说明	(I - 25)
4.4.2 枚举说明	(I - 25)
4.4.3 结构说明	(I - 27)
4.4.4 联合说明	(I - 29)
4.4.5 数组说明	(I - 30)
4.4.6 指针说明	(I - 31)
4.5 函数说明	(I - 32)
4.6 存储类	(I - 33)
4.6.1 外部变量说明	(I - 34)
4.6.2 内部变量说明	(I - 36)
4.6.3 函数的外部说明和内部说明	(I - 37)
4.7 初始化	(I - 37)
4.7.1 基本型和指针型	(I - 38)
4.7.2 构造型	(I - 38)
4.7.3 字符串初值符	(I - 40)
4.8 类型说明	(I - 41)
4.8.1 结构、联合和枚举的类型说明	(I - 42)
4.8.2 typedef 说明	(I - 44)
4.9 类型名	(I - 46)
第五章 表达式与赋值	(I - 44)
5.1 概述	(I - 44)
5.2 操作数	(I - 44)
5.2.1 常量	(I - 44)
5.2.2 标识符	(I - 44)
5.2.3 字符串	(I - 45)
5.2.4 函数调用	(I - 45)
5.2.5 下标表达式	(I - 45)
5.2.6 成员选择表达式	(I - 47)
5.2.7 带运算符的表达式	(I - 48)

5.2.8	括号中的表达式	(I-48)
5.2.9	类型强制转换表达式	(I-48)
5.2.10	常量表达式	(I-49)
5.3	运算符	(I-49)
5.3.1	一般算术转换	(I-49)
5.3.2	取反运算符	(I-50)
5.3.3	间址和取址运算符	(I-51)
5.3.4	sizeof 运算符	(I-51)
5.3.5	乘除运算符	(I-52)
5.3.6	加减运算符	(I-52)
5.3.7	移位运算符	(I-53)
5.3.8	关系运算符	(I-54)
5.3.9	按位运算符	(I-55)
5.3.10	逻辑运算符	(I-55)
5.3.11	顺序求值运算符	(I-56)
5.3.12	条件运算符	(I-57)
5.4	赋值运算符	(I-57)
5.4.1	左值表达式	(I-58)
5.4.2	单目增量与单目减量	(I-58)
5.4.3	简单赋值	(I-59)
5.4.4	复合赋值	(I-59)
5.5	优先级与求值顺序	(I-59)
5.6	副作用	(I-61)
5.7	类型转换	(I-62)
5.7.1	赋值转换	(I-62)
5.7.2	类型强制转换	(I-65)
5.7.3	运算符的转换	(I-65)
5.7.4	函数调用的转换	(I-65)
第六章	语句	(I-66)
6.1	概述	(I-66)
6.2	break 语句	(I-66)
6.3	复合语句	(I-67)
6.4	continue 语句	(I-68)
6.5	do 语句	(I-68)
6.6	表达式语句	(I-69)
6.7	for 语句	(I-69)
6.8	goto 语句和标号语句	(I-70)
6.9	if 语句	(I-71)
6.10	空语句	(I-72)
6.11	return 语句	(I-72)
6.12	switch 语句	(I-73)
6.13	while 语句	(I-76)

第七章 函数 (I-77)

7.1	概述	(I-77)
7.2	函数定义	(I-77)
7.2.1	存储类	(I-77)
7.2.2	返回类型	(I-78)
7.2.3	形式参数	(I-79)
7.2.4	函数体	(I-81)
7.3	函数说明	(I-82)
7.4	函数调用	(I-83)
7.4.1	实际参数	(I-85)
7.4.2	参数数目可变的函数调用	(I-86)
7.4.3	递归调用	(I-87)

第八章 预处理程序伪指令与编

译指令	(I-88)	
8.1	概述	(I-88)
8.2	明意常量与宏	(I-88)
8.2.1	#define 伪指令	(I-89)
8.2.2	#undef 伪指令	(I-90)
8.3	包含文件	(I-91)
8.4	条件编译	(I-92)
8.4.1	#if、#elif、#else和 #endif 伪指令	(I-92)
8.4.2	#ifdef 和 #ifndef 伪指令	(I-95)
8.5	行控制	(I-95)
8.6	编译指令	(I-96)
附录A	语言差别	(I-97)
附录B	语法小结	(I-99)
B.1	标识单元	(I-99)
B.1.1	关键字	(I-99)
B.1.2	标识符	(I-99)
B.1.3	常量	(I-100)
B.1.4	字符串	(I-102)
B.1.5	运算符	(I-102)
B.1.6	分隔符	(I-102)
B.2	表达式	(I-102)
B.3	说明	(I-104)
B.4	语句	(I-107)
B.5	定义	(I-108)
B.6	预处理程序伪指令	(I-108)
B.7	编译指令	(I-109)

第二篇 Microsoft C 编译器

第一章 引言	(I -110)	3.8 浮点选项的选择	(I -144)
第二章 初始步骤	(I -111)	3.8.1 有协处理器的情形	(I -145)
2.1 概述	(I -111)	3.8.2 无协处理器的情形	(I -146)
2.2 磁盘备份	(I -111)	3.8.3 非IBM兼容机的情形	(I -146)
2.3 磁盘目录	(I -111)	3.8.4 浮点选项的兼容性	(I -147)
2.4 硬盘设置步骤	(I -114)	3.9 80186、80188或80286处理器的 使用	(I -147)
2.5 软盘设置步骤	(I -116)	3.10 错误信息概述	(I -148)
2.6 环境设置	(I -120)	3.10.1 C编译器的信息	(I -149)
2.7 CONSYS文件的设置	(I -121)	3.10.2 警告级别的设置	(I -150)
2.8 文件的组织	(I -122)	3.11 调试准备	(I -151)
2.9 实例操作	(I -122)	3.12 优化	(I -151)
2.10 批文件的使用	(I -125)	3.13 大型程序的编译	(I -152)
第三章 编译	(I -127)	第四章 连接	(I -154)
3.1 概述	(I -127)	4.1 概述	(I -154)
3.2 编译器的运行	(I -127)	4.2 连接程序的运行	(I -154)
3.2.1 文件名约定	(I -128)	4.2.1 文件名约定	(I -154)
3.2.2 专用文件名	(I -128)	4.2.2 目标模块提示	(I -155)
3.2.3 源文件名提示	(I -129)	4.2.3 运行文件提示	(I -155)
3.2.4 目标文件名提示	(I -129)	4.2.4 列表文件提示	(I -155)
3.2.5 源列表提示	(I -129)	4.2.5 库文件提示	(I -156)
3.2.6 目标列表提示	(I -129)	4.2.6 分隔输入项	(I -156)
3.2.7 选择缺省回答	(I -130)	4.2.7 选择缺省回答	(I -157)
3.2.8 更换磁盘	(I -130)	4.2.8 终止连接过程	(I -157)
3.2.9 命令行方式	(I -130)	4.2.9 命令行方式	(I -157)
3.2.10 选项	(I -132)	4.2.10 响应文件方式	(I -158)
3.3 编译器选项列表	(I -133)	4.2.11 临时文件	(I -159)
3.4 目标文件的命名	(I -133)	4.3 C程序文件的连接	(I -159)
3.5 列表文件的生成	(I -134)	4.3.1 main函数	(I -159)
3.6 预处理程序的控制	(I -139)	4.3.2 缺省库与库的搜索路径	(I -159)
3.6.1 常量和宏的定义	(I -140)	4.3.3 改变缺省库	(I -160)
3.6.2 预定义的标识符	(I -141)	4.3.4 避免使用的选项	(I -160)
3.6.3 预定义标识符的定义撤消	(I -141)	4.4 列表文件的格式	(I -160)
3.6.4 预处理列表的生成	(I -141)	4.5 复盖	(I -161)
3.6.5 保留注释	(I -142)	4.5.1 复盖的限制	(I -162)
3.6.6 包含文件的搜索	(I -142)	4.5.2 复盖管理程序的提示	(I -162)
3.7 语法检查	(I -143)	4.6 连接程序的选项	(I -163)
3.7.1 标识语法错误	(I -143)	4.6.1 显示选项清单	(I -163)
3.7.2 产生函数说明	(I -143)		

4.6.2	暂停连接过程	(I -163)	6.4	LIB的功能	(I -183)
4.6.3	压缩可执行文件	(I -164)	6.4.1	建立库文件	(I -183)
4.6.4	公用符号列表	(I -164)	6.4.2	修改库文件	(I -183)
4.6.5	使映像文件包含行号	(I -165)	6.4.3	增加库模块	(I -184)
4.6.6	清除库	(I -165)	6.4.4	删除库模块	(I -184)
4.6.7	字母大小写的敏感性	(I -165)	6.4.5	替换库模块	(I -184)
4.6.8	忽略缺省库	(I -166)	6.4.6	复制库模块	(I -184)
4.6.9	控制堆栈大小	(I -166)	6.4.7	移出库模块	(I -184)
4.6.10	设置最大分配空间	(I -167)	6.4.8	合并库	(I -184)
4.6.11	控制段数	(I -167)	6.4.9	生成相互对照表	(I -185)
4.6.12	设置复盖中断号	(I -167)	6.4.10	一致性检查	(I -185)
4.6.13	段序排列	(I -168)	6.4.11	设置页面尺寸	(I -185)
4.6.14	数据的装入控制	(I -168)	第七章	程序维护	(I -186)
4.6.15	运行文件的装入控制	(I -168)	7.1	概述	(I -186)
4.6.16	保持兼容性	(I -169)	7.2	MAKE的使用	(I -186)
4.7	连接程序的执行步骤	(I -169)	7.2.1	MAKE说明文件	(I -186)
4.7.1	段的定位	(I -169)	7.2.2	MAKE的调用	(I -187)
4.7.2	帧地址	(I -170)	7.2.3	MAKE的选项	(I -188)
4.7.3	段的顺序	(I -170)	7.2.4	宏定义	(I -188)
4.7.4	段的组合	(I -170)	7.2.5	宏定义的嵌套	(I -189)
4.7.5	组	(I -171)	7.2.6	特殊的宏定义	(I -189)
4.7.6	装配	(I -171)	7.2.7	推导规则	(I -190)
第五章	运行	(I -173)	7.3	程序维护的例子	(I -191)
5.1	概述	(I -173)	第八章	内存模式	(I -193)
5.2	命令行参数传送	(I -173)	8.1	概述	(I -193)
5.2.1	通配参数的扩展	(I -174)	8.2	标准内存模式	(I -194)
5.2.2	取消命令行处理	(I -175)	8.2.1	小型模式程序	(I -194)
5.3	出口码的回送	(I -176)	8.2.2	中型模式程序	(I -195)
5.4	取消空指针检查	(I -176)	8.2.3	紧凑模式程序	(I -195)
第六章	库管理	(I -178)	8.2.4	大型模式程序	(I -196)
6.1	概述	(I -178)	8.2.5	巨型模式程序	(I -196)
6.2	LIB操作概述	(I -178)	8.3	关键字 near、far 和 huge	(I -196)
6.3	LIB的运行	(I -179)	8.3.1	near、far 和 huge 的库支持	(I -197)
6.3.1	库文件名提示	(I -179)	8.3.2	用 near、far 和 huge 说明数据	(I -198)
6.3.2	库操作提示	(I -180)	8.3.3	用 near 和 far 说明函数	(I -199)
6.3.3	列表文件提示	(I -180)	8.3.4	指针转换	(I -200)
6.3.4	输出库文件提示	(I -181)	8.4	专用内存模式	(I -202)
6.3.5	命令行方式	(I -181)	8.4.1	代码指针	(I -202)
6.3.6	响应文件方式	(I -182)	8.4.2	数据指针	(I -203)
6.3.7	续行	(I -182)	8.4.3	段的设置	(I -203)
6.3.8	终止库处理过程	(I -183)			
6.3.9	选择缺省回答	(I -183)			

8.4.4	库的支持	(I -204)	10.3.6	Pascal 到FORTRAN或C	(I -232)
第九章	高级功能	(I -205)		的接口	(I -232)
9.1	概 述	(I -205)	10.3.7	Pascal 调用FORTRAN或	(I -232)
9.2	取消特殊关键字	(I -205)		C的过程	(I -232)
9.3	压缩结构的存储	(I -205)	10.3.8	C到FORTRAN或 Pascal	(I -233)
9.4	限制外部名字的长度	(I -206)		的接口	(I -233)
9.5	标记目标文件	(I -206)	10.3.9	C调用FORTRAN或Pascal	(I -233)
9.6	取消选择缺省库	(I -206)		的过程	(I -233)
9.7	改变 char 类型的缺省设定	(I -207)	10.3.10	数据类型	(I -233)
9.8	控制栈和堆的分配	(I -207)	10.3.11	返回值	(I -245)
9.9	控制浮点运算	(I -208)	10.3.12	共享数据	(I -246)
9.9.1	在连接时改换库	(I -208)	10.3.13	输入和输出	(I -247)
9.9.2	使用NO87环境变量	(I -209)	10.3.14	编译和连接	(I -247)
9.10	高级优化	(I -209)	10.3.15	错误信息	(I -247)
9.10.1	取消堆栈检查	(I -210)	附录A	ASCII字符集	(I -248)
9.10.2	最大优化	(I -211)	附录B	命令小结	(I -250)
9.11	控制函数调用顺序	(I -211)	B.1	概述	(I -250)
9.12	控制二进制和文本方式	(I -212)	B.2	编译器小结	(I -250)
9.13	设置数据阈值	(I -213)	B.3	LINK小结	(I -253)
9.14	模块和段的命名	(I -213)	B.4	LIB小结	(I -255)
9.15	窗口软件的编译	(I -214)	B.5	MAKE小结	(I -255)
第十章	语言接口	(I -215)	B.6	EXEPACK小结	(I -257)
10.1	概述	(I -215)	B.7	EXEMOD小结	(I -257)
10.2	汇编语言接口	(I -215)	B.8	SETENV小结	(I -257)
10.2.1	分段模式	(I -215)	附录C	CL命令	(I -259)
10.2.2	C调用顺序	(I -218)	C.1	概述	(I -259)
10.2.3	Pascal/FORTRAN调用顺序	(I -220)	C.2	命令格式和选项	(I -259)
10.2.4	汇编例行程序入口	(I -220)	C.3	CL命令的连接	(I -261)
10.2.5	返回值	(I -221)	C.4	附加的选项	(I -262)
10.2.6	汇编例行程序出口	(I -221)	C.5	与XENIX兼容的选项	(I -262)
10.2.7	命名约定	(I -222)	附录D	EXEPACK、EXEMOD和SETENV	(I -264)
10.2.8	寄存器考虑	(I -222)	D.1	引言	(I -264)
10.2.9	程序例子	(I -222)	D.2	EXEPACK实用程序	(I -264)
10.3	混合语言程序设计	(I -225)	D.3	EXEMOD实用程序	(I -265)
10.3.1	内存模式	(I -226)	D.4	SETENV实用程序	(I -266)
10.3.2	选择调用约定	(I -226)	附录E	出口码	(I -268)
10.3.3	命名约定	(I -229)	E.1	概述	(I -268)
10.3.4	FORTRAN到Pascal 或C的接口	(I -230)	E.2	MAKE 与出口码	(I -268)
10.3.5	FORTRAN调用Pascal或C的过程	(I -231)	E.3	MS-DOS批文件与出口码	(I -268)
			E.4	C编译器软件包程序的出口码	(I -269)
			E.4.1	编译器出口码	(I -269)

E.4.2	LINK出口码	(I-269)	G.2	运行时错误信息	(I-279)
E.4.3	Code View 出口码	(I-269)	G.2.1	运行库错误信息	(I-279)
E.4.4	LIB出口码	(I-269)	G.2.2	浮点异常	(I-280)
E.4.5	MAKE出口码	(I-270)	G.2.3	运行限制	(I-281)
E.4.6	EXEPACK出口码	(I-270)	G.3	编译器错误信息	(I-282)
E.4.7	EXEMOD出口码	(I-270)	G.3.1	警告错误信息	(I-283)
E.4.8	SETENV出口码	(I-270)	G.3.2	致命错误信息	(I-286)
附录F	可移植程序	(I-271)	G.3.3	编译错误信息	(I-289)
F.1	概述	(I-271)	G.3.4	命令行错误信息	(I-297)
F.2	程序的可移植性	(I-271)	G.3.5	编译器限制	(I-299)
F.3	机器硬件的差别	(I-272)	G.4	LINK错误信息	(I-299)
F.4	编译器的差别	(I-275)	G.5	LIB错误信息	(I-303)
F.5	环境的差别	(I-278)	G.6	MAKE错误信息	(I-305)
F.6	数据的可移植性	(I-278)	G.7	EXEPACK错误信息	(I-306)
F.7	字节顺序一览	(I-278)	G.8	EXEMOD错误信息	(I-306)
附录G	错误信息	(I-279)	G.9	SETENV错误信息	(I-307)
G.1	概述	(I-279)			

第一篇 Microsoft C语言

第一章 引言

C语言是一种通用的程序设计语言，其程序效率高、语言精练和可移植性均闻名于世。这些优点使它几乎被各种程序设计优先选用，但是事实证明，C语言在系统程序设计中尤其有用。因为它允许程序员编写速度快和紧凑的程序，并可将这些程序移植到其他系统上去。有许多情形，编写良好的C程序在运行速度上可以同汇编语言程序相比，但前者具有更好的可读性以及更容易维护等优点。

C将效率和功能结合在一个相对小的语言中。C语言并不包括完成诸如输入和输出、存储分配、屏幕操作和过程控制等任务的嵌入函数。因此C程序员必须依赖运行库例行程序来完成这些任务。

这种设计有利于C语言的适应性和紧凑性。因为该语言相当简练，它不假设或者强加具体的程序设计模型，而由运行库例行程序根据需要提供支持。程序员根据要求可以最小限度地使用库例行程序，或者为了特殊用途而裁剪库例行程序。

这种设计也有助于将C语言的特征同C的某个具体实现中使用的处理器所持有的特征区分开来，从而有助于程序员编写可移植的代码。该语言的严格定义使它独立于任何具体的操作系统或机器；与此同时，程序员可以很容易地增加系统专用的例行程序，以充分利用机器的效能。

C语言的一些显著特点如下：

- C提供一套完整的循环、条件和传送语句，逻辑地和有效地控制程序流，并支持结构程序设计。
- C提供一个相当大的运算符集。C的许多运算符对应着通常的机器指令，从而允许直接翻译成机器代码。各式各样的运算符让程序员可以用最少量的代码清楚地规定各种不同的运算。
- C的数据类型包括若干种整数类型以及单精度和双精度浮点类型。程序员可以设计更为复杂的数据类型，例如数组和数据结构，以满足特殊程序的需要。
- C程序员可以说明变量和函数的指针。某一项的指针对应着该项的机器地址。明智地使

用指针可以大大提高程序效率，因为指针让程序员在引用该项的方式上和机器一样。C还支持指针算术运算，允许程序员直接存取和处理内存地址。

- C 预处理程序（一个文本处理器）对编译前的文本文件进行处理。对于C程序最有用的是程序常量的定义、宏的使用以及条件编译。预处理程序并不限于处理C文件，它可以处理任何文本文件。
- C是一种灵活的语言，它将大部分的决策留给程序员去做。为了保持这个属性，C对诸如类型转换这类事件施加很少的限制。虽然这样做往往是个优点，但是C程序员必须很好地了解语言，才能理解程序的行为。

本篇定义Microsoft公司实现的C语言，目的是给对C或者其他程序设计语言已有经验的程序员提供参考。

至于Microsoft C与由B·W·Kernighan和D·M·Ritchie所著《C程序设计语言·》一书附录A有关C的定义之间的差异，请查阅本篇的附录A。本篇附录B总结Microsoft C语言的语法。

第二章 C 的基本成分

2.1 概 述

本章叙述C程序设计语言的基本成分，这些基本成分用来构造C程序的名字、数字和字符。本章实际上叙述下列内容。

- 字符集
- 常量
- 标识符
- 关键字
- 注释
- 标识单元

2.2 字符集

C程序使用两个字符集：C字符集和可表示字符集。

C字符集包括字母、数字以及对C编译器有特殊意义的标点符号。C程序就是通过将C字符集中的字符组合成为有意义的语句加以构造的。

C字符集是可表示字符集的子集。可表示字符集包括所有的字母、数字以及可以用单个字符表示的图形符号。可表示字符集的范围取决于所用的终端、控制台或字符装置的种类。

除了字符串常量、字符常量和注释可以使用任何可表示的字符外，C程序只能包含C字符集中的字符。对C编译器来说，C字符集中的各个字符都有一个明确的意义。当C编译器遇到误用的字符或者不属于C字符集的字符时，它就产生错误信息。

下面几节介绍C字符集中的字符和符号，并解释如何以及何时使用它们。

2.2.1 字母和数字

C字符集包括英文字母表的大写、小写字母以及阿拉伯数字系统的10个十进制数字。

- 大写英文字母

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

- 小写英文字母

a b c d e f g h i j k l m n o p q r s t u v w x y z

- 十进制数字

0 1 2 3 4 5 6 7 8 9

这些字母和数字可以用来形成后面将要介绍的常量、标识符和关键字。

C编译器把大写字母和小写字母当作不同的字符。如果在一个给定的项中规定一个小写

字母 a, 就必须使用该小写字母, 而不能在该位置上用一个大写字母 A 代替。

2.2.2 空白符

空格符、水平制表符、换行符、回车符、换页符、垂直制表符以及新行符都称为空白符。因为它们跟打印页上字间和行间的空白起同样的作用。在程序中这些字符将用户定义的项, 例如常量和标识符, 同其他的项分隔开来。

CONTROL-Z 被视为文件结束的标志。编译器对任何跟在 CONTROL-Z 标志后的文本不予理采。

C 编译器忽略各空白字符, 除非它们用作分隔符或者作为字符常量或文字常量的成分。这意味着可以使用额外的空白符来增加程序的可读性。注释也被当作空白符 (见 2.6 节)。

2.2.3 标点符号和特殊符号

C 字符集中的标点符号和特殊符号有多种用途, 从组织程序文本到定义编译器或被编译的程序所要执行的任务。表 2.1 列出这些字符。

表 2.1 标点符号和特殊符号

字符	名称	字符	名称	字符	名称	字符	名称
,	逗号)	右括号	!	感叹号	%	百分号
.	句号	[左方括号		垂直线	&	和号
:	分号]	右方括号	/	正斜杠	^	脱字号
:	冒号	{	左花括号	\	反斜杠	*	星号
?	问号	}	右花括号	~	代字号	-	减号
'	单引号	<	左角括号	_	下划线	=	等号
"	双引号	>	右角括号	#	井号	+	加号
(左括号						

这些字符对 C 编译器都有特殊的意义。它们在 C 语言中的用法将在全篇中进行介绍。可表示字符集的标点符号如果没在表 2.1 中出现, 则只能用在字符串常量、字符常量和注释中。

2.2.4 换码串

换码串 (或称转义字符) 是个特殊的字符组合, 它们在字符串和字符常量中表示空白符和非图形字符。它们的典型用法是规定终端和打印机的动作, 如回车和制表动作, 以及提供通常有特殊意义的字符的文字量表示, 如双引号字符。换码串由一个反斜杠及跟着的一个字母或者若干个数字组成。表 2.2 列出 C 语言的换码串。

如果在一个反斜杠后面有一个不在上述表中的字符, 则忽略反斜杠而将该字符表示为文字常量。例如: 在字符串常量或字符常量中, `\c` 表示字符 c。

`\ddd` 和 `\xdd` 序列允许用三位数字的八进制字符代码或二位数字的十六进制字符代码表示 ASCII 字符集里的任何字符。例如, 可以用 `\010` 或者 `\x08` 表示退格符 (backspace);

表 2.2 换 码 串

换码串	含义	换码串	含义
\n	换 行	\a	响铃 (告警)
\t	水平制表	\'	单 引 号
\v	垂直制表	\"	双 引 号
\b	退 格	\\	反 斜 杠
\r	回 车	\ddd	八进制数表示的ASCII字符
\f	换 页	\xdd	十六进制数表示的ASCII字符

用 \0 或者 \x0 表示 ASCII 码空字符 (null)。

在一个八进制的换码串中只能出现数字 0 到 7, 且至少必须出现一个数字, 但可以少于三个数字。例如: 在八进制表示中退格符可以表示为 \10。与此类似, 一个十六进制的换码串必须至少包括一位数字, 第二位数字可以省略。因此, 退格符的十六进制换码串可表示为 \x08 或者 \x8。

注意, 当在字符串中使用八进制或十六进制换码串时, 最安全的做法是给出换码串的所有数字。否则, 如果紧跟在换码串后面的字符恰好是一个八进制数或十六进制数, 则该数将被解释为该换码串的一部分。例如打印字符串 \x7Bell 的结果将是 {ell, 因为 \x7B 被解释为 ASCII 码左花括号字符 ({)。字符串 \x07Bell 才正确地表示响铃字符后紧跟着 Bell。

换码串允许将非图形的控制码送到显示装置上去。例如, 换码串 \033 经常用作终端或打印机的控制命令的第一个字符。

非图形字符必须用换码串加以表示。因为在 C 程序中使用非图形字符将出现不可意料的结果。

用以引入换码串的反斜杠 (\) 在字符串定义和在预处理程序定义中还被当作连续字符使用。当一个换行符跟在反斜杠后面, 则忽略换行符, 而将下一行视为前一行的一部分。

2.2.5 运算符

运算符是一些特殊字符的组合, 它们规定对数值如何进行转换和赋值。编译器将这些字符的每种组合解释为单元, 称为“标识单元”(见 2.7 节)。

表 2.3 列出 C 运算符及其名称。运算符必须严格按照该表的规定, 对于由多字符组成的运算符, 其字符间不得有空白符。sizeof 运算符不包括在本表中, 因为它构成关键字 (sizeof) 而不再是符号了。

表 2.3 运 算 符

运算符	名称	运算符	名称	运算符	名称	运算符	名称
!	逻辑非	-	减, 算术负	&&	逻辑与	?:	条 件
~	按位反	*	乘, 间址		逻辑或	++	增 量
+	加	/	除	,	顺序求值	--	减 量