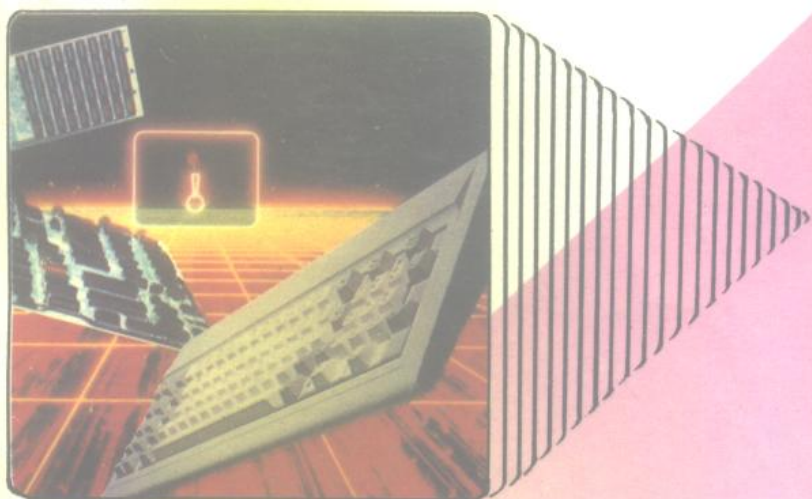




单片机 实用开发 指南

甘登岱 李广东 徐建平 编著

科学出版社

单片机实用开发指南

甘登岱 李广东 徐建平 编著

科学出版社

1994

(京)新登字 092 号

内 容 简 介

本书介绍了目前微型计算机上最流行的单片机开发工具 C-51,并对单片机的基础知识、典型接口及有关问题作了详尽描述。全书共分两部分:第一部分从 C-51 的安装开始,系统地介绍了 C-51 的编译连接方法、扩展语言关键字的使用、如何配置 C-51 系统,以及编译程序命令行选项、库函数等;第二部分介绍了单片机典型接口及相应程序,这些接口包括键盘、显示器、打印机及 AD/DA 转换等,本部分还简单介绍了单片机开发中应注意的若干问题。

本书适合于从事单片机开发工作的科技人员和有关人员参考。

单片机实用开发指南

甘登岱 李广东 徐建平 编著

责任编辑 那莉莉

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

北京外国语大学印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1994 年 5 月第 一 版 开本:787×1092 1/16

1994 年 5 月第一次印刷 印张:19

印数:1—4 100 字数:439 千字

ISBN 7-03-004040-6/TP·338

定价:19.00 元

前 言

众所周知,随着单片机集成度的提高,其应用范围越来越广。但是,同使用汇编语言开发软件一样,对单片机软件的开发始终存在着一个开发复杂、移植和维护困难的问题。今天,这个问题终于可以用 C-51 解决了。这就是说,广大单片机开发人员终于可以使用 C 语言进行单片机开发了。

本书共分两大部分。第一部分主要讲述 C-51 的安装及使用方法,共包括四章及若干附录。

在第一部分,我们将目前最流行的单片机开发工具 C-51 介绍给读者。但是作为一本有关单片机开发的实用性指导手册,仅有这些内容显然是不够的,为此我们还将单片机的基本原理、各种通用接口及开发过程中应注意的一些问题介绍给读者。这就是本书第二部分要讲述的内容。

本书由北京航空航天大学 304 教研室甘登岱主编,参加本书编写工作的还有李广东、徐建平、苏为民、章强、宋岩葵、林海、高木林、刘连生、苏伟。本书的录入排版工作由章群山、苏海东和刘威负责,他们为本书的出版付出了辛勤的劳动,在此对他们表示由衷的感谢。

全书由任天电子信息技术研究所审校,在此对任天电子信息技术研究所的全体同仁表示谢意。

尽管作者在编撰本书时已竭尽全力,但由于时间仓促,加之水平有限,不当之处在所难免,敬请读者批评指正。

目 录

第一部分 C-51 使用指南	1
第一章 C-51 安装指导	2
1.1 安装前	2
1.2 磁盘中的文件	3
1.3 使用 INSTALL.EXE 文件	4
1.4 手工安装 C-51 开发工具集	4
1.5 环境变量	6
1.6 安装高容量 DXT 程序文件	7
1.7 IBM PC DOS 和 OS/2 错误返回值	9
第二章 C-51 概述	10
2.1 编译一个简单的 C 程序	11
2.2 连接一个简单的 C 程序	17
2.3 修改 CSTARTUP.S03 和 PUTCHAR.C	27
2.4 扩充语言介绍	33
2.5 范例 5:多模块示例	39
第三章 用户指南	46
3.1 8051 和 C-51 4.0 版	46
3.2 段和 C-51 编译器	53
3.3 存储模式与硬件	55
3.4 ANSI C 和 C-51 4.0 版	60
3.5 数据类型与存储类	62
3.6 扩展 C 语言关键字	65
3.7 配置	71
3.8 连接	80
3.9 可编程 PROM 代码、PROM 编程器和开发工具	86
3.10 写高效率的微处理器 C 代码	89
3.11 局部变量定位、重入及递归	93
3.12 堆栈	97
3.13 中断	100
3.14 汇编语言接口	106
3.15 组存储器模式	115
第四章 C 参考	123
4.1 命令行选择项参考	123
4.2 8051 内存数据表示	140
4.3 扩展 C 语言关键字参考	145
4.4 段参考	157

4.5 C库函数	159
4.6 包含文件	171
4.7 两个要求特殊处理的函数类型	173
4.8 编译器诊断	174
附录 A 附加 Archimedes 软件开发工具	189
A.1 SimCASE	189
A.2 SimI/O	189
A.3 附加工具和服务	190
附录 B 升级到 C-51 V4.0	191
B.1 C-51 2.x 版和 3.x 版之间的不同	191
B.2 C-51 3.x 版和 4.x 版之间的差别	192
B.3 Archimedes C-51 4.0 版扩展语言	193
附录 C ANSI 标准 C 与 K&R C	195
附录 D 附加 XLINK 错误信息	198
附录 E 术语汇编	199
第二部分 单片机开发基础	205
第一章 单片机基本结构及通用接口	206
1.1 概述	206
1.2 MCS-51 单片机硬件结构及指令系统	208
1.3 单片机片外存储器	223
1.4 应用中规模集成电路作通用接口	226
1.5 8031 片内通用接口	228
第二章 单片机键盘、显示器接口	232
2.1 硬件设计要求	232
2.2 BCD 拨盘与 8031 接口	232
2.3 8031 与动态扫描显示器接口	233
2.4 8279 键盘显示器接口	235
2.5 8031 与 CRT 接口	239
第三章 打印机接口应用举例	245
3.1 8031 和 TPμP-40A 打印机接口应用举例	245
3.2 8031 与 PP40 绘图仪接口应用举例	253
第四章 8031 与模数、数模转换应用实例	269
4.1 硬件设计首先考虑的问题	269
4.2 8031 与 ADC0809 接口	269
4.3 8031 与 G14433 接口	272
4.4 A/D, D/A 应用软件举例	273
第五章 单片机应用系统开发	275
5.1 单片机应用系统的软、硬件开发方法	275
5.2 应用系统结构与应用系统设计内容	280
5.3 微机系统中的主要干扰渠道及抗干扰措施	283
5.4 印刷电路板及电路的抗干扰设计	289

5.5 微机系统的布线设计	292
5.6 软件的抗干扰设计.....	293

第一部分 C-51 使用指南

单片机的软件开发同使用汇编语言开发软件一样,始终存在着开发复杂,移植和维护困难的问题。今天,这个问题终于解决了,这就是 C-51。这就是说,广大单片机开发人员终于可以使用 C 语言进行单片机开发了。

本部分共包括四章及五个附录:

第一章主要讲述如何在 PC 个人微机上安装 C-51 编译器。

第二章主要提供一组编译和连接程序的练习、不同类型硬件的系统配置,以及 C-51 4.0 版扩展语言关键字的使用。

第三章包括六节,每节讲述一个专题,其复杂度逐步增加。当然,如果需要的话,也可以任意阅读各节,而不一定依次进行。第一节讲述诸如 C-51 和硬件基本关系的一些基础问题,中间各节描述了如何配置 C-51 系统,及如何根据需要选择合适的编译和连接选项。最后讲述了有关汇编语言接口和组存储模式等问题。

第四章按字母顺序分别讲述了编译程序命令行选项、C 库例程、C-51 语言关键字等。此外,本章还描述了编译器错误信息及编译器支持的 `#include` 文件。

附录包括了 C-51 相关产品的介绍,C-51 2.x,3.x,4.x 等各个版本的差别,ANSI C 和旧的 K&R C 之间的差别。

本部分的内容主要是为那些用 C 语言对 8051 微控制器系列硬件进行编程的读者编写的。因此,如果读者对 8051 结构不熟悉的话,请先阅读本书第二部分的内容。

第一章 C-51 安装指导

本章讲述如何安装和运行 C-51 开发工具集。首先介绍了所提供的安装工具,然后就环境变量、MS-DOS 扩充存储器等有关专题进行了讨论。SUN,DEC 或者 HP 的用户也可以参考本部分内容。

本章阐述怎样在 IBM PC 系统上安装 C-51 开发工具集。如果使用不同的系统,如 VAX 或 SUN 工作站等,可参考相应的资料。

本章分以下七个部分来详细地进行阐述:

- (1) 系统要求及开始安装前所需要做的几个步骤。
- (2) 所提供文件的概述。
- (3) 利用 INSTALL.EXE 程序进行自动安装。
- (4) 手工安装 C-51 开发工具集。
- (5) 使用 Archimedes 定义的环境变量。
- (6) 安装 DOS 扩展存储器以支持编译和连接。
- (7) DOS 和 OS/2 编辑出错的返回值摘要。

1.1 安 装 前

在安装 C-51 前请阅读 README.DOC 文件。

利用 MS-DOS 命令行,有几种方法可以打印 README.DOC 文件:

```
C:\ARCHIMED>print README.DOC
```

```
C:\ARCHIMED>copy readme.doc prn
```

```
C:\ARCHIMED>type readme.doc > prn
```

安装 C-51 前做一份 C-51 磁盘备份。

本章的其他部分适用于 IBM PC 的安装。如果用户使用的是非 PC 宿主机系统(VAX, HP, 或 SUN 工作站等),请阅读单独的说明,它们包含了适当的安装信息。

注意,在 OS/2 下,Archimedes 可执行程序可在“DOS 兼容机”上运行而没有“DOS 640K 内存容量”的限制。

IBM PC 版本的 C-51 开发工具集包括两张 5 1/4 英寸 1.2 兆 MS-DOS 格式软盘。

IBM PC 系统的要求是:

- (1) IBM PC 机或兼容机,运行 MS-DOS(或 PC DOS)系统 2.11-4.01 版本,或者 OS/2 系统,1.0 以上版本。
- (2) 硬盘最小自由空间为 1.5 兆字节,以编译及存放有关程序。
- (3) 至少 512K 的随机存储器 RAM 空间(建议用 640K 空间),其中至少要有 450K 可用空间用来编译和连接。用户可以用 DOS CHKDSK 工具来检查 RAM 可用自由空间,如:

```
C:\>chkdsk
```

如果没有足够的可用 RAM 存储空间,用户将会得到其中一条 DOS 错误信息:

- program too lange to fit in memory
- Exception failure

或编译错误信息:

- Compiler out of memory

如果系统有至少 450K 的可用空间,但仍然出现“out of memory”或者“decution failure”错误信息,请删除已安装的 Terminate Stay Resident(TSR)程序(例如 Sidekick 或网络软件)。

另外,由于和 TSR 不兼容会引起 XLIB 管理程序在开始时是正确的,但在 * 提示符下不接受任何命令的情况。

80286,80386,80486 PC 机的扩充内存可用来进行 C 编译和连接,这要求使用 C-51 工具集所提供的用来编译和连接的可执行版本 .DXT 文件,而不使用缺省的 .EXE 文件。

要使用带有 .DXT 版本的扩充内存,可利用安装工具手工拷贝文件的方法来安装下面几部分所显示的工具集,然后按照本章 1.6 节中的提示去做。

注意“扩充”内存和“扩展”内存是不同的,后者不能用来进行 C-51 编译和连接。

1.2 磁盘中的文件

工具集上的实际文件随版本号而变。当前版本的文件列表可在 C-51 工具集中的 README.DOC 文件中得到。README.DOC 文件还包括本产品的最新信息,而这些信息由于太新因而没有包含在本书中。

表 1.1 列出了 C-51 创建和使用的扩充文件类型。这些文件类型不是每一种都包含在磁盘上(例如,不包含 .LST 或 .MAP 文件)。

表 1.1 文件类型及其扩展类型

*.EXE	MS-DOS 可执行文件
.DXT	被重新命名的“.exe”可执行文件,在配备扩充存储器的 80286,80386 或 80486 机型中运行
*.C	C 语言源文件
*.H	C 语言包含(“#include”)文件
*.SO3	8051 汇编语言源文件
*.INC	8051 汇编语言“include”文件
*.RO3	编译和汇编输出的 C 运行时程序库文件和目标文件
*.A03	连接输出的 8051 可执行文件
*.LST	编译和连接输出的缺省表文件
*.MAP	为连接表(“图”)文件而增设的推荐文件
*.XCL	连接命令行选择项的命令文件
*.DOC	补充文档的正文文件
*.BAT	控制编译和连接的批处理文件
*.00T	为连接可执行输出文件而增设的缺省文件

1.3 使用 INSTALL.EXE 文件

INSTALL.EXE 为程序安装文件,用户可使用它自动执行安装。INSTALL.EXE 利用在第一张盘上的一个临时文件 INSTALL.DAT 来工作。INSTALL.DAT 文件是一个 ASCII 码文件,如需要可以被检测和修改,但在 Archimedes's Software Technical Support 组中是不支持这种修改的。

要运行安装工具,请把 C-51 第一张盘插入 A 驱动器,转到 A 驱动器并输入

```
A:\>INSTALL
```

INSTALL.EXE 可以把 C-51 文件安装在单目录下或多目录下。缺省目录名如表 1.2 所示。

表 1.2 目录和文件

目录	内容
单目录:	
\ARCH	所有文件
多目录:	
\ARCH\BIN	可执行文件
\ARCH\LIB	库
\ARCH\SOURCE	C 和汇编源文件
\ARCH\INCLUDE	"#include"文件

从安装工具中可以把缺省目录名取代。如果确认为多目录配置,则 INSTALL.EXE 工具对 AUTOEXEC.BAT 文件做如下修改:

```
set c_INCLUDE=cpatch to # include file directory>
```

```
set XLINK_DFLIDIR=cpath to library file d. rectory>
```

接着重新启动或手工执行 AUTOEXEC.BAT 文件,这些命令将设置搜索路径的编译和连接。

1.4 手工安装 C-51 开发工具集

本节讲述在没有选用安装工具 INSTALL.EXE(或者不用 PC 版本,因为只有 PC 版本才有安装工具)的情况下,怎样手工安装 C-51 开发工具集。第一节讲述怎样安装到单目录中,第二节讲述怎样安装到多目录中。

单目录安装比较简单,不需要为操作系统设置“路径”。可是如果项目比较复杂,则在此目录下就会有許多文件。多目录安装需要多个步骤,但对文件的组织有较好的控制。记住:C-51 开发工具集的文件需要有将近 1.5 兆字节的自由硬盘空间。命令行列出的例子是基于 MS-DOS 系统的安装。

1.4.1 在单目录上安装 C-51

为把所有文件安装到单个目录上,请创建一个新目录(如 ARCHIMED),并把在盘上的 C-

51 开发工具集的所有文件都拷贝到这个目录上,例如可用下列所示的 DOS 命令序列:

```
C:\> md archimed
C:\> cd archimed
C:\ARCHIMED > copy a: *.* *
```

上面序列的第一行显示了使用 MD 命令来创建一个子目录 ARCHIMED。第二行用 CD 命令改变当前目录,使 ARCHIMED 成为当前目录。第三行用 COPY 命令把第一张软盘的所有文件拷贝到\ARCHIMED 目录下。当第一张盘中的内容拷贝完后,插入第二张盘再拷贝其中的文件。

1.4.2 在多目录中安装 C-51

本节描述怎样创建一个多目录结构,描述哪些文件应在哪个目录中以及怎样建立 C-51 可执行文件的 DOS 路径。

创建目录结构

为创建一个多目录结构,请创建如下目录:

\ARCHIMED\BIN	可执行文件
\ARCHIMED\LIB	C 函数库
\ARCHIMED\SOURCE	C 和汇编源文件
\ARCHIMED\INCLUDE	C“# include”文件

可以用下列 DOS 命令序列来实现:

```
C:\> md archimed
C:\> cd archimed
C:\ARCHIMED> md bin
C:\ARCHIMED> md include
C:\ARCHIMED> md source
C:\ARCHIMED> md lib
C:\ARCHIMED> copy a: *.exe bin
C:\ARCHIMED> copy a: *.h include
C:\ARCHIMED> copy a: *.r03 lib
C:\ARCHIMED> copy a: *.c source
C:\ARCHIMED> copy a: *.s03 source
C:\ARCHIMED> copy a: *.inc source
C:\ARCHIMED> copy a: *.xcl source
C:\ARCHIMED> copy a: *.doc source
```

上面列出了下述命令的使用:

- DOS MD 命令建立了 ARCHIMED 目录。
- DOS MD 命令使 ARCHIMED 成为当前目录。
- DOS MD 命令在 ARCHIMED 目录中创建了多个子目录。
- DOS MD 命令把文件组放到各自的目录中。

子目录的内容

在上面的多目录安装中,C-51 程序文件在 ARCHIMED\BIN 目录中。其中包括 C 编译程序(C-51.EXE)、连接程序(XLINK.EXE)、汇编程序(A8051.EXE)和库程序(XLIB.EXE)。

“#include”文件带 .H 扩展名,放在 INCLUDE 目录中。这些是包含 ANSI C 和 C-51 特殊定义的包含文件。

LIB 目录包括 6 个不同的 C 函数库文件版本,它们分别适用于六种不同的存储模式。

SOURCE 目录包括所提供的 C 语言源文件(带 .C 扩展名)、汇编语言源文件(带 .SO3 扩展)、汇编语言包含文件(带 .XCL 扩展名)。在这些目录中用户将运行辅助程序并建立第一个 C 程序。

注意,汇编“include”文件必须和汇编源文件放在同一个目录中。这是因为未连接 C-51 C 编译程序,A8031 汇编程序没有搜索路径环境变量使它在单独的目录上找到包含文件。有关搜索路径更详细的内容在本章 1.5 节讲述。

设置路径

建立完子目录后,就可以把 BIN 目录的路径名加到操作系统 PATH 项目中。这使 DOS 能找到编译、连接和库可执行文件,即使当前目录可能不是 SOURCE 目录。

这时,在 DOS 中,AUTOEXEC. BAT 文件的 PATH 语句就包含了 ARCHIMED\BIN,如下所示:

```
C:\ARCHIMED>PATH=C:\;C:\ARCHIMED\BIN;C:\WHATEVER\ELSE
```

另外,必须建立下一节所描述的环境变量。这使 C-51 编译器和 XLINK 连接器能找到各自的包含文件和库文件。

1.5 环境变量

本节解释几个环境变量的使用。这些环境变量通过操作系统可确定编译程序,连接程序和库程序的路径及其他信息,从而使我们不必在每次命令行中打出全部路径和其他信息。

DOS 和其他操作系统提供了宿主内存的“环境”区来确定名称(“环境变量”)及相关正文串。这样,当一个程序(像 C-51 编译程序)在运行时,可检查某个特定变量的“环境”,从中识别和读出一些通过正文串所提供的环境特定信息。

例如,可在 AUTOEXEC. BAT 文件中用下面的 DOS 命令告诉编译程序到哪儿去查找“#include”文件。

```
set c_INCLUDE=c:\arch\include\
```

当编译一个有“#includes”文件的 C 源文件时,编译程序首先在当前目录中查找 #include 文件。如果没有找到文件,编译程序就检查操作系统环境来看看是否设置了“C INCLUDE”变量。如果找到上述变量的话,编译程序就会为 #included 文件在正文串(c:\arch\include)中查找指定的目录。

C_INCLUDE 路径必须在串尾包含一个所示的反斜线符(“\”),因为此路径串被逐字放到包含文件名中。

可利用分号把各个路径分开的方法来确定 C INCLUDE 的多路径。如下所示:

```
c:\ARCHIMED>set C_INCLUDE=drive1:\path1\;driver2:\path2\
```

1.5.1 连接程序的目录模块搜索路径:XLINK_DFLTDIR

XLINK 连接程序为变量 XLINK_DFLTDIR (XLINK, DeFauLT DIR ectory)检查 DOS 环境。

XLINK_DFLTDIR 提供了一个目录,在此目录中可搜索目标文件,如 C 函数库文件 CL8051*.R03,或者编译程序和连接程序创建的其他可重写的目标模块。下面列出了连接程序搜索路径的设置(只有一个路径,没有分号):

```
c:\ARCHIMED>set XLINK_DFLTDIR = drive:\path\
```

1.5.2 通用环境变量:QCC8051 和 XLINK_ENVPAR

另外,C-51 和 XLINK 还提供了“通用”环境变量,可用它来增加命令行。C-51 检查 QCC8051 的环境,而 XLINK 通常检查 XLINK_ENVPAR 的环境。

作为 QCC8051 使用举例,因为编译程序通常产生混和的 C 程序和汇编程序,所以在每次编译时总是需要在编译程序命令行上加上-L-q 选择项:

```
C:\ARCHIMED\SOURCE>C-51 -L -q myfile
```

如果在 AUTOEXEC. BAT 文件中增加如下行:

```
Set QCC8051 = -L -q
```

-L -q 会一直“自动地”被增加到 C-51 命令行的开始部分,用户只要输入

```
C:\ARCHIMED\SOURCE>C-51 myfile
```

而得到的结果命令行为

```
C:\ARCHIMED\SOURCE > C-51 -L -q myfile
```

由于 XLINK 几乎总是被用作“连接程序控制文件”,它包括 XLINK 所要求的众多命令行选项,因此,XLINK_ENVPAR 并不常用。

1.5.3 库环境变量

XLIB 库响应它自己的环境变量设置。

1.5.4 增设的内存环境变量:DOS16M

环境变量“DOS16M”是 DMINF02 工具用来检查 PC 机是否被确认为和它的缺省方式不同的方式。可为多种宿主 PC 机配置 C-51 的 DOS 内存管理系统,请参见本章 1.6 节。

1.6 安装大容量.DXT 程序文件

为在内存范围为 640KB 的 MS-DOS 操作系统下编译和连接大文件,C-51 使用 DOS/16M 系统。DOS/16M 使 C-51 编译程序和 XLINK 连接程序把 PC 内存扩充到 16M 字节。

本节内容包括支持哪种“超过 640K”的 PC 机内存、最小系统要求、如何使用 PMINFO 工具检查兼容性、怎样为硬件配置扩充内存、怎样把编译程序和连接程序的.DXT 可执行版本安装到硬盘上。

C-51 编译程序和 XLINK 连接程序只访问扩充(extended)内存,不能访问扩展(expanded),LIM 或 EMS 内存。

1.6.1 系统要求和兼容性

使用.DXT 程序文件必须要有 AT 型兼容机,并带有至少 256K 字节的扩充内存。

为访问扩充内存,必须使用编译和连接可执行程序的替换版本。这些版本是由分别带有 C-51. DXT 文件和 XLINK. DXT 文件的 C-51 环境工具提供的。

虽然完全建立了 PC 的“扩充内存”硬件,得到了高度的兼容性,但仍然有些 PC 机和 DOS/16M 不兼容。

另外,还可能出现软件和设备驱动程序不能兼容的问题。例如,DOS/16M 和 LDISK 中 RAMDISK DOS 工具兼容,如果系统包含有别的工具建立的 RAM 磁盘,DOS/16M 可能不能检测出此磁盘对扩充内存的使用,并能引起两个程序试图夺取对同一块内存的访问权,结果使系统出现故障。

如果在使用扩充内存时出现问题,可试着重新启动系统而不要用 TSR (Terminate and Stay Resident) 工具(如 Sidekick 或磁盘缓冲程序)。

1. 6. 2 用 PMINFO 工具确定兼容性

为确定 PC 的扩充内存是否与 DOS/16M 兼容,请在 A 驱动器上插入 2# 软盘以运行提供的 PMINFO 工具,并键入

```
A:\>pminfo
```

PMINFO 工具将报告系统可用扩充内存的总量。如果这一步失败了,可使用 . DXT 安装工具继续下去,就像下面一段所描述的那样。

如果运行不成功(如返回不可能的值或死机),则可能有一个“非标准”的扩充内存工具。在这种情况下,用户可以检查表 1. 3 中所示的 DOS 环境适用的 DOS16M 的各种可容许的值,并试着找出最可能的一个,直到 PMINFO 程序完全成功。如果没有适用的 DOS16M 值可工作,则认为 PC 机和扩充内存系统不兼容。

例如,如果 PC 机有 Intel 386 主板(inboard),可试着键入

```
A:\>set DOS16M=INBOARD
```

或者如果是 AT&t 6300+, 试着键入

```
A:\>set DOS16M=6
```

表 1. 3 DOS16M 环境变量值

机器	设置	注释
NEC 98 series	1	*
PS/2	2	自动设置
80386	3	自动设置
80386	INBOARD	* 对于 INTEL INBOARD
Fujitsu FMR-60,-70	5	*
AT&T 6300+	6	*
80286	7	* 以前的 phenenix BIOS
80286	9	自动设置
80286	10	* 快速切换选择
Zenith Z= 24K	13	* 以前的 BIOS

注: * 表示用户必须为这些机型设置 DOS16M。

1.6.3 把 .DXT 文件拷贝到硬盘驱动器

为使用程序文件的 .DXT 版本,请安装本章所述的全部开发工具,然后拷贝 .DXT 文件覆盖缺省的 .EXE 版本。例如,若是单目录安装,则把此软盘插入 A 驱动器,并键入

```
C:\ARCHIMED>copy a:\C-51.dxt c:\arch\bin\c-51.exe
```

```
C:\ARCHIMED>copy a:\xlink.dxt c:\arch\bin\xlink.exe
```

这将覆盖以前安装在硬盘上的 .EXE 版本,而代之以要求的重命名的 .DXT 版本。

编译程序和连接程序的 .DXT 版本在每种 .EXE 版本中都是相同的。

1.7 IBM PC DOS 和 OS/2 错误返回值

若在 DOS 批文件中使用 C-51 编译程序,则必须检测编译是否成功,以便决定下一步该怎么办。所以 C-51 编译程序返回表 1.4 所示的 DOS(或 OS/2)错误返回值。表中的值可以和 .BAT 或 .CMP 文件中的“IF ERRORLEVEL”一起使用。

表 1.4 操作系统错误返回值

错误层	含义
0	编译成功
1	警告,若使用-W(禁止警告信息)编译选择项则返回 0
2	有错误
3	检测到致命错误(编译中止)

第二章 C-51 概 述

本章将讲解如何编译和连接第一个程序、配置启动例程序和 I/O 目标、使用 C-51 版本 4.0 扩充语言关键词、以及连接 C 和汇编语言多模块。

本章有的内容是 C-51 4.0 开发环境所包含的内容。它提供了取得编译和连接的第一个编码的最快途径。

本章包括如下几方面的内容：

- (1) 编译一个 C 程序。
- (2) 连接一个 C 程序。
- (3) 修改 CSTARTUP 和 putchar()。
- (4) 使用 C-51 扩充语言。
- (5) 一个多模块范例。

范例简介

下面给出几个范例的摘要。与这些范例有关的源文件均在软盘中提供。

第一个范例(2.1 节)将编译一个 C-51 工具提供的简单 C 程序。实例程序访问一个 `#include` 文件且定义了几个数据工具,它可用来检测 C-51 软件是否安装到计算机上,是否创建了一个 C 和汇编的混合语言列表文件。

第二个范例(2.2 节)告诉用户如何连接在第一步中编译好的程序,这可通过使用连接程序控制文件 LNK8051.XCL 加一些命令行选择项进行。第二个连接程序控制文件 TUT1.XCL 是为此范例定制的,它告诉用户怎样为一个特定的应用程序定制一个连接程序控制文件。接着再连接程序,这时使用 TUT1.XCL,并检测产生的连接程序映象文件,它包括各段绝对地址、数据以及目标代码。

第三个范例(2.3 节)告诉用户如何修改缺省 CSTARTUP 运行时的初始代码,这是在复位或加电后马上执行的代码。另外,可修改被 printf()调用的 putchar()单字符 I/O 函数来向 I/O 目标发出每个字符,这种传送一般采用缺省 8051 嵌入式串行 I/O 端口进行。

用户要做的修改将把串行 I/O 端口初始代码加到 CSTARTUP 例程中,并把制表符到空格的转换加到 putch()例程。

然后,用 XLIB 库程序把已修改的版本替换紧凑内存模式的 C 运行库 CL8051C.R03。这将使得用新的版本替换缺省版本来连接。

第四个范例(2.4 节)演习了许多 C-51 版本 4.0 发行的重要扩充关键词,包括 interrupt 关键字、sfr 数据类型、内存区说明符 data 和 xdata。

用户还可学会怎样使用扩充语言的 `#pragma` 等价格式,它允许使用许多扩充特性,但却使程序保持了 ANSI C 的兼容性。

最后一个范例(2.5 节)将编译两个 C 文件并汇编一个汇编语言文件。这些程序描述了 ANSI C 增加的关键词的使用,并告诉用户如何从 C 语言中把参数传入或传出汇编语言例程。

然后用库 XLIB 把三个目标模块放到一个单库文件中。最后,把模块从库文件中连接到一