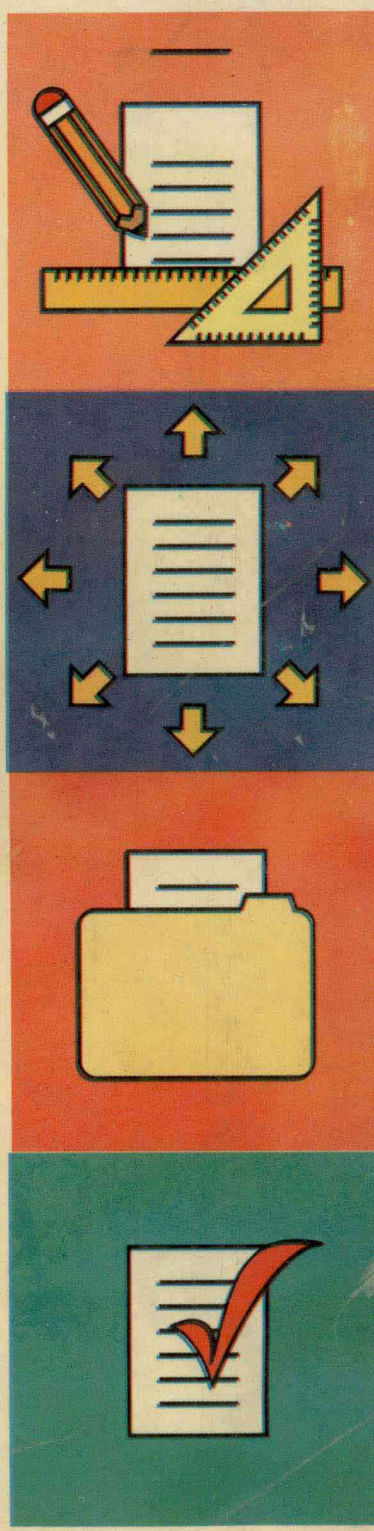
学苑出版社

计算机实用软件技术系列丛书



宏汇编编程技术与调试工具

袁　力
李　争　编写
柳　萍

沙　枫　审校

计算机实用软件技术系列丛书

宏汇编编程技术与调试工具

袁　力
李　争　编写
柳　萍
沙　枫　审校

学苑出版社

1993·北京

(京) 新登字 151 号

内 容 摘 要

本书介绍了宏汇编语言编程的基本概念和基本方法，以及完整的指令和伪指令系统；论述了汇编语言程序员常遇的问题；给出了大量的编程实例．本书取材先进，内容丰富，实用性强；叙述层次分明，详细透彻，易于理解．本书是汇编语言程序员很好的参考书．

计算机实用软件技术系列丛书

宏汇编编程技术与调试工具

编　　写：袁 力　李 争　柳 萍
责任编辑：徐建军
出版发行：学苑出版社　　邮政编码：100032
社　　址：北京市西城区成方街 33 号
印　　刷：列电印刷厂
开　　本：787×1092
印　　张：26.125　　字数：604 千字
印　　数：1—5000
版　　次：1993 年 11 月第 1 版第一次
ISBN 7-5077-0876-41/TP.25
本册定价：29.00 元

学苑版图书印、装错误可随时退换

目　录

上 篇　CodeView 使用指导

此为试读,需要完整PDF请访问: www.ertongbook.com

下 篇 MASM 编程技术

上 篇　使用 CodeView 指导

第一章 简介

1.1 简介

CodeView 调试程序是一个可执行程序，能帮助你调试用 C 语言和宏汇编语言编写的软件。

CodeView 调试程序是一个面向对象的工具，使用它可以跟踪程序中的逻辑错误；允许在程序实时运行时分析程序。CodeView 调试程序显示源代码或汇编代码，指明将要执行的程序行，动态观察变量值（局部变量或全局变量），切换屏幕来显示程序输出，完成其它功能。对汇编语言及高级语言的程序员来说，掌握该调试程序是很容易的。

使用 CodeView 时，首先要从所编译的目标文件中产生一个可执行文件。制作一个程序的可执行文件时，它必须是本系统可以装入并运行的格式。必须使用正确的选项编译和连接可执行文件，生成 CodeView 所需要的行号和符号表。可以使用能调用链接程序 ld 的编译程序或 CC。使用 CodeView 时的编译和连接的正确选项在第二章"启动"中进行介绍。

1.2 关于本篇

本篇介绍了 CodeView 调试程序的使用，有关的命令、显示及界面在本篇中进行详细介绍。

本部分由下列章节组成:

- 第二章"启动"介绍了如何建立一个能在CodeView调试程序上运行的C或汇编程序；怎样启动调试程序及如何选择各种命令选项。
- 第三章"CodeView的显示"介绍了CodeView的显示屏幕及界面，包括功能键及键盘命令。
- 第四章"对话命令的使用"介绍CodeView命令的基本形式。
- 第五章"CodeView表达式"介绍了如何建立命令中所使用的复杂的表达式。
- 第六章"代码执行"介绍了CodeView执行程序中的代码的命令。
- 第七章"检查数据和表达式"介绍了几种数据求值命令。
- 第八章"断点管理"介绍了如何使用断点中断执行。

此为试读，需要完整PDF请访问：www.ertongbook.com

- 第九章"管理观察语句"介绍了使用观察语句命令设置、删除和列表显示观察语句.
- 第十章"代码检测"介绍了几个能检测程序代码或有关代码的数据的命令.
- 第十一章"修改代码或数据"介绍了如何暂时改变代码以便在CodeView调试程序中进行测试.
- 第十二章"CodeView系统控制命令的使用"介绍了控制CodeView调试程序操作的命令.

第二章 启动

2.1 简介

启动 CodeView 调试程序需要以下几步。首先必须准备好需要调试的程序的特殊格式的可执行文件，然后才调用调试程序，还要指定能影响调试程序操作的选项。

本章介绍了使用 C 语言或汇编语言如何生成 CodeView 格式的可执行文件以及 CodeView 调试程序如何装入程序。本章列出了编译或汇编前所应考虑的有关调试程序的一些限制及编程规则。本章最后介绍了宏汇编中如何使用调试程序。

2.2 限制

不能使用 CodeView 调试程序去调试包含文件中的源代码。这一限制只局限于 CodeView 调试程序，而不考虑所使用的语言。

2.3 为 CodeView 调试程序准备程序

要使用 CodeView 调试程序，必须用正确选项编译并连接程序。这些选项可以指示编译器和连接器产生一个可执行文件，该文件除了可执行代码外还包括行号以及一个符号表。

注意：

为了简化，本节及下面三小节中所使用的术语"编译"是指产生目标模块的过程。除了本章的"汇编程序的准备"这一节外，本章中有关编译的概念都等效于汇编。

并不是所有版本的编译和连接程序都支持 CodeView 选项。请参考有关编译程序版本的文字资料。如果想调试没有使用 CodeView 选项编译连接过的可执行文件，或者你使用了一个不支持这些选项的编译器，那么只能使用汇编级的调试程序。这就意味着 CodeView 调试程序不能显示源代码或理解源级符号，如函数和变量符号。

CodeView 的两种基本显示模式是源模式，即用源程序行显示程序；汇编模式，即按汇编语言指令方式显示程序。这两种模式还可以组合成一种混合模式，即同时用源程序行方式和汇编语言指令方式显示程序。

2.3.1 编程规则

可以对任何合理的 C 或宏汇编源代码进行编译或汇编以产生可执行文件，该文件可用 CodeView 调试程序调试，但无论怎样必须承认一些技巧性编程给调试带来了困难。

C 语言和宏汇编语言允许将代码放在分离的包含文件中，并使用包含伪指令将文件读入源文件中。但不能在包含文件中使用 CodeView 调试源代码。开发程序的最好方法是建立不同的目标模块，将目标模块连入程序。CodeView 调试程序支持相同段中的分离目标模块的调试。如果在每一行中只写一条源语句会使 CodeView 调试程序执行起来更有效更方便，许多语言允许源文件的一行上写多条语句，这种技巧虽不阻止 CodeView 调试程序运行，但调试程序必须将每一行作为一个单元，因为它不会将这一行拆成分离语句。因此如果在同一行上有三个语句，就不能在语句之间设断点或冻结执行。最多只能在这三条语句的开始或下一行的开始冻结执行。

C 语言和宏汇编支持一种宏扩展类型，CodeView 调试程序不能调试源模式中的宏，必须调试前先扩展宏，否则调试程序将它们作为简单的语句或指令来处理。

2.3.2 CodeView 编译选项

当编译所要调试的程序的源文件时，必须指定命令行中的-Zi 选项。-Zi 选项指示编译程序在目标文件中包含行号及符号信息。还可以使用-g，它和-Zi 同义。

如果一些模块中不需要符号信息，那么编译这些模块时可使用-Zd 选项而不是-Zi。-Zd 选项不在目标文件中写入符号信息，因此使用此选项可节省磁盘空间和内存。例如，如果程序由五个模块组成，而只需要调试其中的一个模块，可以使用-Zi 选项编译此模块而使用-Zd 选项编译其它模块。还可以检查全局变量，并可查看用-Zd 选项编译的模块的源程序行，但不能检查局部变量。

除此而外，如果使用的是高级语言，使用-Od 选项可以取消优化，因为优化代码虽然在安排上更有效，但却使程序指令不严格地对应源程序行。在调试结束后，你可以使用所喜欢的优化去编译程序的最终版本。

注意：

使用宏汇编时，-Od选项不起作用。

在编译没有完成之前，不能使用调试程序，CodeView 调试程序不会更正你的程序中的语法错误或编译错误。一旦成功编译完程序，就可以使用调试程序来定位程序中的逻辑错误。

在有关用具体语言进行编译及连接的章节后将给出编译示例。

2.3.3 CodeView 连接选项

如果分别使用 ld 连接目标文件或需要调试的文件，那么应指定-g 选项。该选项指示连接程序将符号的地址及源程序行编入执行文件中。

注意如果使用某个驱动程序自动启动连接器（例如，C 中的 cc），那么无论什么时候在命令行中描述了-Zi，连接程序便自动选择-g 选项启动。

虽然使用-g 选项连接的可执行文件也可执行，但它们要比其它文件大，因为文件中有多余的符号信息。为了缩减程序，应该使用 strip 命令，或者完成程序调试后，不使用-Zi 选项编译连接最终版本。参考“程序员参考手册”中的 strip 命令解释。

“C 和汇编语言的编译和连接”节将给出连接的示例。

2.3.4 C 程序的准备

若想使用 CodeView 来调试 C 程序，需要用 C 编译器来编译．该编译器的低版本不支持 CodeView 编译选项，请详细参考《开发系统版本注释》.

C 源代码的编写

C 语言支持包含文件的使用，虽然也支持#include 伪指令．但不能调试放入包含文件中的源代码．因此在#define 宏和结构定义中保留包含文件的使用.

C 语言允许一行上有几条语句，这一技巧却使调试这些行的代码变得很困难．例如，下列代码在 C 语言中是合法的.

```
code = buffer[conut];  if  (code = = '\n')  ++lines;
```

这个代码由三条独立的源语句组成，但放在一行上，在调试时不能独立选取，如，不可能在++lines 处停止程序的执行．而下列这种形式的代码则在调试时很容易.

```
code = buffer[count];
if  (code = = '\n')
++lines;
```

因为这样使代码易读，并且也符合好的编程技巧.

使用 CodeView 调试程序不容易调试宏，因为调试程序不能中止宏．因此，当调试具有潜在副作用的复杂宏时，首先要象正规源代码那样编写宏.

C 程序的编译和连接

C 编译支持-Zi、-Zd 和-Od 选项（若想了解这些选项的含义，参看"CodeView 编译选项"节）．cc 驱动程序接受这些选项.

CodeView 调试程序支持几种语言的混合编程．例如：如何连接 C 模块和其它语言模块，可参看本章中的"汇编程序的准备"节.

举例

```
cc -Zi -Od -o example example.c
cc -c -Zi -Od example.c
cc -g -o example example.o
cc -Zi -Od -c mod1.c
cc -Zd -Od -c mod2.c
cc -Zi mod1.o mod2.o
```

在第一个例子中，用 CC 编译连接源文件 example.c，同时 cc 命令还可以建立 CodeView 格式的目标文件 example.o，并且自动启动选择-g 选项的连接程序．第二个例子示范了如何使用 cc 中的-c 选项编译连接源文件 example.c．因为 cc-c 不能启动连接程序，还必须第二次键入 cc 以连接目标文件．这些例子产生一个具有 CodeView 调试程序所需要的行号信息、符号表、未优化代码的可执行文件 example.

在第三个例子中，编译源模块 mod1.c 产生一个带所有符号及行信息的目标文件．编译 mod2.c 后产生一个带有限制信息的目标文件．然后再用 cc 连接所得的目标文件．（这

一次，cc 不再编译，因为变量带有.O 扩展名）。键入-Zi 启动带-g 选项的连接器，产生一个可执行文件 a.out，其中的 mod2.c 模块很难调试，因为它们几乎不包含诸如逻辑变量名之类的符号信息。但这样的可执行文件要比其两个模块都用所有符号信息编译的文件所占用的磁盘空间小。

2.3.5 汇编程序的准备

若想用 CodeView 来调试汇编程序，首先要用宏汇编来汇编程序。（本章中"使用低版本的汇编程序工作"节中介绍了如何调试低版本的宏汇编程序。

汇编源代码的编写

对 2.3 版或更新版本的宏汇编来说，可以使用简化段伪指令，这样就可以保证用正确方法中声明段，以供 CodeView 调试程序使用。（这些指令也可辅助混合语言编程）。如果不使用这些伪指令，就需要确认代码段的类名为 CODE。

在源模式下不能跟踪宏，除了在汇编或混合模式外，宏一般作为一条指令，看不见宏的内容或指令。因此只能在把它们作为宏之前调试其代码。

宏汇编也支持包含文件，但不能在包含文件中调试代码。最好保留宏及结构定义的包含文件。

因为汇编语言没有自己的表达式求值，所以要使用 C 的表达式求值。C 语言最接近汇编语言，要保证表达式求值能识别程序中的符号及标号，在编写汇编模块时，应遵循下列条例：

- 汇编语言没有一个明确的方法声明实数。但初始化带小数点的实数和不带小数点的整数时，也可为实数和整数传递正确符号信息。（缺省类型是整数）。例如，下列初始化语句中，REALSUM 是实数，COUNTER 是整数。

```
REALSUM  DD .0.0
COUNTER  DD  0
```

  必须在数据定义时初始化实数，若使用?，那么汇编语言在产生符号信息时将变量认作整数，CodeView 调试程序不能正确求出变量的值。
- 避免在符号名中使用特殊符号。
- 使用混合语言编程时，用-Mx或-MI选项进行汇编可以避免大小写的冲突。缺省情况下，在产生目标代码时，汇编将所有符号转换成大写形式。C 语言则不进行这样的转换。因此除非在使用调试程序时关闭大小写敏感开关，否则，CodeView 调试程序不认为 C 程序中的 var 和汇编程序中的 var 是相同变量。

汇编和连接

汇编程序支持-Zi 和-Zd 汇编时（assemble-time）选项，不支持-Od 选项，所以不能使用-Od 选项。

如果需要连接汇编程序和用 C 写的模块时（有大小写区分），应该选用-Mx 或-MI 选项进行汇编。

汇编后，用-g 选项连接产生 CodeView 格式的可执行文件。

举例

```
masm -Zi  example.asm
cc -g example.c

masm -Zi mod1.asm
masm -Zd mod2.asm
cc -g mod1.o mod2.o
```

第一个例子汇编源文件 example.asm，并产生 CodeView 格式的目标文件 example.o，键入带-g 选项的 cc 启动连接程序，产生包含调试程序所需要的符号表和行号信息的可执行文件 a.out。

第二个例子产生包含符号和行号的目标文件 mod1.o，以及只包含行号不包含符号表的目标文件 mod2.o。接着连接目标文件，结果生成可执行文件 a.out。它的第二个模块较难调试的。第二个模块几乎不包含诸如局部变量名的符号。这样的可执行文件要比两个模块都用-Zi 选项汇编所产生的文件小。

2.4 启动 CodeView 调试程序

启动调试程序前，必须保证所有需要的文件在适当的地方有效存在。下列是源级调试所需要的文件。

文件	位置
/usr/bin/cv	位于/usr/bin目录下的CodeView程序文件。
/usr/lib/cv.hlp	位于/usr/lib目录下的CodeView帮助文件。如果CodeView调试程序找不到帮助文件，继续使用CodeView调试程序是没有问题的，但在使用其中任何一个帮助命令时会报错。
program	所需要调试的程序的可执行文件必须在当前目录下，或者在键入 CodeView 命令行指定的包括路径名的目录下。找不到可执行文件时，CodeView 调试程序便显示错误信息，并且不能启动。
source.ext（扩展名依赖于所用编程语言）	正常情况下，源文件必须在当前目标下。但当在编译时规定了源文件的文件描述，则文件描述变成存贮于可执行文件中的符号信息的一部分。例如，使用命令行参数 demo.ext 编译时，CodeView 调试程序认可源文件在当前目录下。但如果编译时，使用带路径名/surce/demo.ext 的命令行参数，则调试程序认可源文件在目录/source 下。如果调试程序在可执行文件所描述的目录下找不到源文件（通常是当前目录），则程序提示一个新的目录。这时或者键入一个新目录，或者按回车键表示这个模块不需要源文件。但如果没有源文件的话，就必须用汇编模式来调

此为试读，需要完整PDF请访问：www.ertongbook.com