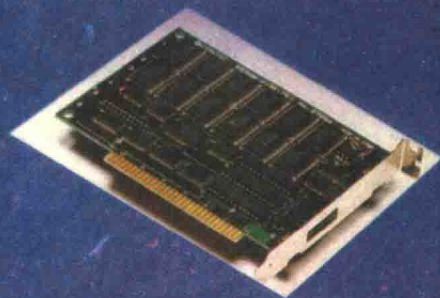


计算机知识普及系列丛书



NDP—C 386 使用手册

谢俊编写
王刚

学苑出版社

计算机知识普及系列丛书

NDP C—386 使用手册

谢 俊 王 刚 编写

吴 欢 审校

学苑出版社

1993.

(京)新登字 151 号

内 容 提 要

本书是 NDP C-8 的用户手册。它讲述了在 386 或 486 机器(带协处理器)上,运行编译器和创建程序所必需的信息,包括编译选项和开关、实用程序和硬件支持,如协处理器支持和保护模式编程等。此外,它还讲述了 NDP C-8 的优化处理、NDP 语言之间的接口、向 NDP C 移植程序的方法和跟踪与调试程序的方法,详细描述了 NDP C-8 的全部内部函数与例程,以及 GREX 扩展图形库中的所有函数。

欲购本书的用户,请直接与北京 8721 信箱联系,电话 2562329,邮码 100080。

计算机知识普及系列丛书

NDP C-386 使用手册

编 写:谢 俊 王 刚

审 校:吴 欢

责任编辑:甄国宪

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:施园印刷厂

开 本:787×1092 1/16

印 张:28.5 字 数:675 千字

印 数:1~3000 册

版 次:1993 年 11 月北京第 1 版第 1 次

ISBN7-5077-0821-7/TP·19

本册定价:19.00 元

学苑版图书印、装错误可随时退换

目 录

第0章 引 言	(1)
第一章 NDP C 编译器的运行	(2)
1.1 手册的设计与更新原则	(2)
1.2 系统的硬件需求	(2)
1.3 系统的软件需求	(3)
1.4 软件及文档组成	(3)
1.5 安装过程	(4)
1.6 测试已安装的编译器	(5)
1.7 编译驱动程序,cc	(7)
1.8 环境变量和编译驱动程序	(7)
1.9 编译程序使用概述	(10)
1.10 使用 Phar Lap 汇编程序	(15)
1.11 使用 Phar Lap 连接程序	(16)
1.12 Phar Lap 的 DOS 扩展程序 RUN386	(17)
1.13 不带协处理器时的编译	(19)
第二章 使用编译驱动程序——cc	(20)
2.1 cc 的描述	(20)
2.2 cc 的语法	(21)
2.3 编译驱动程序的选项和开关	(21)
2.4 通用选项和开关	(22)
2.5 编译控制开关	(23)
2.6 数值协处理器控制开关	(23)
2.7 代码生成器的控制	(24)
2.8 优化开关	(25)
2.9 关闭缺省的优化开关	(25)
第三章 使用 Mini—Make	(26)
3.1 启动 Mini—Make	(26)
第四章 NDP C—386 的扩展	(31)
4.1 NDP C—386 规范	(31)
4.2 基本 C 语言的扩展	(31)
第五章 优 化	(35)
5.1 引 言	(35)
5.2 优 化	(35)

5.3	基准测试.....	(36)
5.4	NDP C—386 实现的存储类型	(46)
5.5	通用优化技术.....	(52)
第六章	与 NDP 编译程序的接口	(54)
6.1	引言.....	(54)
6.2	NDP 语言之间的接口	(55)
6.3	NDP 语言与汇编语言之间的接口	(75)
6.4	与实模式例程的接口	(125)
第七章	向 NDP C—386 移植程序	(159)
7.1	与其他 MicroWay NDP 编译程序的兼容性	(159)
7.2	有关字长的问题	(159)
7.3	有关字节顺序的问题	(159)
7.4	对齐的要求	(160)
7.5	浮点数的范围和精度	(160)
7.6	汇编语言接口	(160)
7.7	表达式的求值顺序	(160)
7.8	编译优化的非法假定	(160)
7.9	隐含的寄存器用法	(161)
7.10	内存分配假定.....	(161)
7.11	—MO 和 —OLM	(161)
7.12	源程序级调试器的有关问题.....	(161)
7.13	编译程序的内存大问题.....	(161)
第八章	错误信息和恢复.....	(163)
8.1	第一遍扫描错	(163)
8.2	第二遍扫描错	(163)
8.3	运行错误	(164)
第九章	NDP C—386 库函数	(165)
9.1	概 述	(165)
9.2	库函数详解	(165)
abort		(165)
abs		(165)
access		(166)
acos		(168)
asctime		(169)
asin		(169)
assert		(170)
atan		(170)
atan2		(170)
atexit		(172)
atof		(173)
atoi		(173)

atol	(173)	fgetpos	(199)
bcmp	(174)	fgets	(199)
bcopy	(174)	fileno	(201)
bsearch	(175)	filln	(202)
bufcpy	(175)	floor	(203)
bzero	(175)	fls	(203)
cabs	(176)	fmod	(204)
calloc	(177)	fopen	(204)
ceil	(178)	fprintf	(205)
cfree	(178)	fputc	(206)
__cleanup	(180)	fputs	(206)
clearerr	(181)	fread	(207)
clearn	(181)	free	(209)
clock	(182)	freopen	(209)
close	(182)	frexp	(210)
cos	(183)	fscanf	(211)
cosh	(183)	fseek	(212)
creat	(184)	fsetpos	(213)
ctime	(185)	ftell	(213)
difftime	(186)	ftruncate	(215)
div	(186)	fwrite	(215)
__doprnt	(186)	gamma	(217)
__doscan	(186)	gcvt	(218)
ecvt	(187)	getc	(219)
erf	(188)	getchar	(220)
erfc	(190)	getenv	(220)
exit	(191)	getl	(220)
__exit	(191)	getpid	(221)
exp	(192)	gets	(221)
fabs	(193)	getw	(223)
fclose	(193)	gmtime	(224)
fcvt	(193)	hypot	(224)
fdopen	(195)	index	(225)
feof	(196)	isalnum	(225)
ferror	(196)	isalpha	(225)
fflush	(197)	iscentrl	(225)
ffs	(198)	isdigit	(226)
fgetc	(198)	isgraph	(226)
		islower	(226)

isprint	(226)	putl	(247)
ispunct	(226)	puts	(248)
isspace	(227)	putw	(248)
isupper	(227)	pxerror	(250)
isxdigit	(227)	qsort	(250)
isatty	(227)	raise	(251)
j0	(227)	racos	(251)
j1	(228)	rand	(252)
jn	(229)	rasin	(252)
labs	(230)	ratan	(253)
ldexp	(230)	ratan2	(254)
ldiv	(231)	rcos	(256)
localeconv	(231)	rcosh	(257)
localtime	(232)	read	(258)
log	(232)	realloc	(259)
log10	(233)	rename	(261)
longjmp	(233)	rewind	(261)
lseek	(234)	rexp	(262)
malloc	(234)	rindex	(263)
mblen	(235)	rsin	(265)
mbstowcs	(235)	rsinh	(266)
mbtowc	(235)	rsqrt	(267)
memcpy	(235)	rtan	(268)
memchr	(236)	rtanh	(269)
memcmp	(237)	sbrk	(270)
memcpy	(237)	scanf	(271)
memicmp	(238)	set_buf_size	(272)
memmove	(239)	setmode	(274)
memset	(240)	set_num_buffers	(275)
mktime	(240)	setbuf	(277)
modf	(241)	setjmp	(278)
ndptype_	(241)	setlinebuf	(279)
open	(242)	setlocale	(280)
perror	(243)	setvbuf	(280)
pow	(244)	signal	(281)
printf	(244)	sin	(281)
putc	(246)	sinh	(282)
putchar	(246)	spawn	(282)

sprintf	(283)	swab	(299)
sqrt	(284)	system	(300)
srand	(285)	tan	(300)
sscanf	(286)	tanh	(301)
strcat	(287)	time	(302)
strchr	(287)	tmpfile	(302)
strcmp	(288)	tmpnam	(302)
strcoll	(289)	toascii	(303)
strcpy	(289)	tolower	(303)
strcspn	(290)	toupper	(304)
strerror	(290)	truncate	(305)
strindex	(290)	ungetc	(306)
strlen	(291)	unlink	(306)
strncat	(292)	va__start	(307)
strncmp	(293)	va__arg	(307)
strncpy	(294)	va__end	(308)
strpbrk	(294)	ufprintf	(308)
strrchr	(294)	vprintf	(308)
strrindex	(295)	vsprintf	(308)
strsave	(296)	wctowbs	(308)
strspn	(297)	wctowb	(309)
strstr	(297)	write	(309)
strtok	(298)	y0	(309)
strtol	(298)	y1	(310)
strtoul	(299)	yn	(311)
strxfrm	(299)		

第十章 数值协处理器支持..... (313)

10.1 编译器的浮点选项——开关 n	(314)
10.2 数值异常——主动与被动处理.....	(315)
10.3 协处理器结构.....	(317)
10.4 IEEE 数值系统介绍	(320)
10.5 内部 NDP 异常处理函数	(329)
10.6 NDP 异常处理程序的定制和编写	(333)
10.7 舍入、精度、及无穷大控制.....	(344)

第十一章 NDP C—386 扩展图形库 GREX

11.1 引言	(354)
11.2 设计的考虑.....	(354)

11.3	库的组成.....	(355)
11.4	库中的函数.....	(357)
11.4.1	图形例程.....	(357)
11.4.2	文本方式例程.....	(358)
11.4.3	端口、块与中断例程	(359)
11.4.4	DOS 时间与日期函数	(359)
11.5	图形例程详解.....	(360)
11.6	文本例详解.....	(379)
11.7	端口、块中断例程详解	(398)
11.8	DOS 时间与日期函数	(401)
11.9	与 Microsoft C 图形库兼容的函数	(402)
11.10	例子程序 cdemo. c 和 textwin. c	(404)
第十二章	跟踪与调试.....	(405)
12.1	引言	(406)
12.2	调试过程.....	(406)
12.3	内存保护错误.....	(406)
12.4	编制一个快速的 _user _debug _ 例程	(407)
第十三章	寄存器别名变量——C 嵌入汇编.....	(410)
13.1	寄存器别名变量.....	(411)
第十四章	NDP C—386 语言概述	(422)
14.1	引言	(422)
14.2	语法	(422)
14.3	基本数据类型.....	(422)
14.4	与硬件有关的信息.....	(424)
14.5	关键字.....	(424)
14.6	操作码.....	(426)
14.7	表达式.....	(428)
14.8	语句	(428)
14.8	预处理伪指令.....	(429)
附录 A	编译错误信息.....	(432)
附录 B	运行错误信息	(435)
附录 C	可从 Fortran 和 C 中调用的时间与日期函数	(438)

第〇章 引 言

0.1 软件及文档组成:

- ☐ 两张 $5\frac{1}{2}$ 英寸 1.2M 软盘;本软件也可以 $3\frac{1}{2}$ 英寸软盘提供。
- ☐ 《NDP C-386 用户手册》(本手册)。

0.2 软件的安装

安装本软件的简单方法是将 1 号盘插入驱动器 A,键入 a:install,并按回车键。然后,按照屏幕提示进行安装。

本软件也可以手工安装,请参阅本手册第一章。

0.3 编译器的测试

在安装 NDP C-386 编译器之后,可用程序 HI.C 对它进行测试。该程序在 EXAMPLES 目录中,参见本手册第一章。

如果您的计算机上安装了 Intel80287 或 80387 协处理器,则应键入命令:

```
cc hi. c
```

当编译过程完成后,请用下述命令运行刚生成的可执行程序:

```
run386 hi
```

如果您的计算机上安装了 mW1167 或 W3167 协处理器,则应键入命令:

```
cc hi. c -n4
```

当编译过程完成后,请用下述命令运行刚生成的可执行程序:

```
run386 hi -1167 on
```

该程序的输出应是:

```
hi!
```

第一章 NDP C 编译器的运行

1.0 概述

本章论述了在硬盘系统上安装 NDP C-386 编译器的方法,并演示了编译一个示范程序的过程。本章还讨论了文件的组织和系统的硬件需求,以及 MicroWay 的编译驱动程序 cc。cc 控制了整个编译过程,它按照开关的设置方式激活编译器、汇编器和连接器。

1.1 手册的设计与更新原则

NDP C 完全实现了 Brian W. Kernighan 和 Dennis Ritchie 在《C 程序设计语言》一书中定义的 C 语言。《C 程序设计语言》保留了标准 C 的定义。NDP C-386 提供的扩展功能便于书写新的软件,或移植原有的应用程序。这些功能包括 Western Electric 的扩展功能以及 Berkeley 编辑器(用于实现 UNIX)的所有功能。

NDP C-386 的文档由本手册组成。它描述了 C 语言的 MicroWay 实现,包括完整的标准 C 语言和所有的 MicroWay 扩展。本手册还描述了在 Intel 80386(带协处理器)上运行编译器和建立程序所需的信息,编辑选项和开关,实用程序以及硬件支持信息,如算术协处理器支持和保护模式编程等。

NDP C-386 软盘 1 上的 README.DOC 文件包含了当前编译器的有关信息。该文件提供了本手册尚未提到的变动情况。较小的变化情况列在 README.DOC 文件中,而较大的修改则包含在新版本的 C 文档中。

1.2 系统的硬件需求

☐ 基于 386 的任何个人计算机,包括 IBM PS/2(70 型和 80 型)、Compaq DeskPro 386、Everex STEP 386、Dell310、AT&T6386、Tandy4000、HP Vectra RS/20、Acer1100,或任何带有 Intel Inboard/386 或 386/387Turbo-AT,或兼容的 386add-on board IBM PC-AT 及其兼容机。

☐ 编译器并非必需一个算术协处理器,但是,如果要求浮点运算,则编译器产生的代码将需要由协处理器来执行。可用的协处理器包括 Intel 80287、80387、MicroWay 公司的 mW1167 和 Weitek3167。

☐ 双面高密 1.2MB,5 $\frac{1}{4}$ 英寸软盘驱动(如有必要,也可用 3 $\frac{1}{2}$ 英寸磁盘)。

☐ 可用空间至少 2M 字节的硬盘驱动器

☐ 母板上存贮容量 640K。

☐ 至少 1MB 的扩展内存。推荐为 4MB。

在编译一个程序时,MS-DOS、Phar Lap DOS 扩展程序、编译程序以及编译驱动程序(任选)必须同时驻留在内存中。在操作过程中,编译器使用扩展内存来存贮符号表信息和建立全局分析树。因此,应至少准备 1MB 的扩展内存。如果程序中具有 1000 行以上的单个过程,则应配备至少 2MB 的扩展内存。如果使用 NDP C 的虚拟存贮功能和 Phar Lap 虚拟存贮管理程序,则可在仅有一兆字节的内存空间和足够的硬盘空间的情况下,编译任意大小的程

序。

1.3 系统的软件需求

- ☐ MS-DOS 3.2 及以上版本。
- ☐ Phar Lap 386ASM 汇编程序, 2.2 及以上版本。
- ☐ Phar Lap 386LINK 连接程序, 2.2 及以上版本。
- ☐ Phar Lap RUN386 DOS 扩展程序, 2.2 及以上版本。

1.4 2.0 版软件及文档组成

- ☐ 两张高密软盘。
- ☐ 《NDP C-386 用户手册》。

1.4.1 磁盘内容

主要文件如下:

README. DOC	说明文件的内容及当前改动(请阅读)
NDPC386. EXP	386 驻留 C 编译程序
INSTALL. EXE	安装程序
_CO0. OBJ	无协处理器的 NDP 模块。
_CO1167. OBJ	Weitek 协处理器的 NDP 模块。
_CO287. OBJ	80287 协处理器的 NDP 模块。
_CO387. OBJ	80387 协处理器的 NDP 模块。
DOS386. OBJ	运行时启动代码和 DOS 接口。
CC. EXE	NDP C 编译驱动程序。
LIBC. LIB	80287 和 80387 的 C 支持库。
LIBC1167. LIB	Weitek 协处理器的 C 支持库。
LIBC	LIB. 80287 和 80387 的 C 支持库。
LIBC1167. LIB	Weitek 协处理器的 C 支持库。
LIBGREX. LIB	MicroWay 图形支持库。
LIBM. LIB	数学库(用 C 语言编写)。
LIBM1167. LIB	Weitek 协处理器的数学库。
LIBM287. LIB	80287 优化数学库。
LIBM387. LIB	80387 优化数学库。
HL. C	C 的测试程序。
\EXAMPLES	包含例子源文件的子目录。

实用程序:

MMC	EXE	Mini-Make 程序。
MMC	DBF	Mini-Make 的数据库文件。

87R	COM	TSR 驻留程序(按下 control R 后转储寄存器的内容)。
LISTNDP	EXP	列表显示源代码与错误信息。
CHECKNDP	EXP	报告机器上已安装的协处理器。
PMEM	EXP	按块(大小为 4KB)分配内存并报告可分配内存的总量。
UTILS	DOC	上述实用程序的文档说明。

1.5 安装过程

NDP C-386 的安装过程如下:

- ☐ 对原盘进行备份。
- ☐ 将磁盘拷贝到硬盘的一个子目录下。
- ☐ 如果硬盘上没有 phar Lap 工具程序,则应将它们拷贝到硬盘的一个子目录下。
- ☐ 提供定义 DOS 环境变量,以便命名上述目录。
- ☐ 改变 DOS 的环境变量 Path,使其包含编译驱动程序和 Phar Lap 工具程序所在的目录。

这些步骤详细描述如下:

步骤 1:

使用适当的 DOS 命令备份原装软盘,例如:

```
C>diskcopy a: b:
```

步骤 2:

将磁盘上的文件拷贝到硬盘上的一个子目录下,即选择合适的驱动器(C:、D:等)并创建编译程序所在的目录。下列 DOS 命令将编译程序拷贝到驱动器 C 的\ndp20 目录下(copy 命令的/v 选项将验证文件拷贝是否正确):

```
C:\> md \ndp20
```

```
C:\> cd \ndp20
```

```
C:\ndp20> copy a: *.* /v
```

只需要在编译器目录中保留与机器上的协处理器有关的文件。使用下表来确定可以删除的文件:

<u>协处理器</u>	<u>所需文件</u>
Intel 80287	_CO287.OBJ LIBM287.LIB,LIBC.LIB,LIBF.LIB
Intel 80387	_CO387.OBJ LIBM387.LIB,LIBC.LIB,LIBF.LIB
Weitek mW1167 或 W3167	_CO1167.OBJ LIBC1167.LIB,LIBM1167.LIB, LIBP1167.LIB

步骤 3:

运行编译程序和编译生成的程序时需要 Phar Lap 软件工具。编译程序的运行并不一定需要子目录\example 和\utils(在原装软盘中)下的文件,但最好还是把这些目录中的文件放到一个合适的地方,如\ndp20 目录及其子目录。下列实用程序应存放在另一个目录下:

386ASM.EXE	Phar Lap 汇编程序
386LINK.EXE	Phar Lap 连接程序
RUN386.EXE	Phar Lap DOS 扩展程序

使用适当的 DOS 命令将这些 Phar Lap 实用程序拷贝到一个目录下。下列 DOS 命令可将 Phar Lap 软盘拷贝到驱动器 C 的\phar 目录:

```
C:\> md \phar
C:\> cd \phar
C:\phar> copy a: *.* /v
```

步骤 4:

编译驱动程序 cc 在建立程序时,通过 DOS 环境变量获得所需文件的位置。这些环境变量可交互式或用批文件来设置。如果需要把上述建立的目录作为编译程序所在的目录,则应敲入下列例子中的命令:

```
set NDP=c:\ndp20
set TOOLS=c:\phar
```

建议将这些命令放在批文件中,因为交互地设置环境变量很容易出错。

在需要同时定义其他环境变量时,使用批文件显得尤为方便。另外,还可以使用不同版本的批文件以选择不同版本的编译器或 Phar Lap 工具。当然,这些命令可放在 autoexec. bat 文件中。关于批处理文件及其建立过程,请参阅 DOS 手册。

关于编译驱动程序使用的环境变量的完整叙述,请参阅第 1.8 节。

步骤 5:

改变 DOS 的搜索路径,以便在任一目录下启动编译驱动程序。在 autoexec. bat 文件的 Path 命令行中加上 c:\ndp20 就可以了。同时,为了方便地使用 Phar Lap DOS 扩展程序,还应在 DOS 搜索路径中加上\phar 目录。最后的路径命令大致如下所示:

```
PATH=C:\;C:\DOS;C:\PHAR;C:\NDP20
```

1.6 测试已安装的编译器

本节叙述怎样证实编译器已正确安装。方法是编译并运行 hi. c 示范程序(由原盘提供)。该程序由安装程序拷贝到编译器所在的目录。

按照前面所述安装编译器和 Phar Lap 工具程序,并保证 NDP 和 TOOL 环境变量赋值正确。然后,在 DOS 提示符下敲入下列命令之一:

如果安装了 80387 协处理器,则应打入 cc hi. c

如果安装了 80287 协处理器,则应打入 cc hi. c -n0

如果安装了 Weitek 协处理器,则应打入 cc hi. c -n4

(-n0 和 -n4 开关见 2.6 节)。

编译驱动程序依次调用编译程序,汇编程序和连接程序时,在屏幕上显示如下信息:

hi.c;

NDP C-386 Ver 2.0.x Copyright (C) 1987,1990 MicroWay, Inc.

386|ASM; 2.2b Copyright (C) 1986-89 Phar Lap Software, Inc.

386|LINK; 2.2b Copyright (C) 1986-89 Phar Lap Software, Inc.

如果显示下列信息之一:

spawn error: no such file or directory

或

"? cannot open executable file: ndpf386.exp"

则表示环境设置可能不对。

敲入 DOS 命令 SET,检查环境变量的设置是否正确。

该命令显示当前的 MS-DOS 环境变量表。DOS 搜索路径应包括编译驱动程序 cc 所在的目录。NDP 环境变量必须指定包含编译程序和库的目录,TOOLS 环境变量必须包含 Phar Lap 工具程序所在目录。例如:

PATH=C:;C:\ndp20;C:\phar

NDP=C:\ndp20

TOOLS=C:\phar

编译和连接过程的结果是生成一个名为 hi.exp 的二进制可执行文件。使用下列命令之一,通过 Phar Lap DOS 扩展程序运行该文件即可证实这一点:

带有 80287 或 80387 协处理器时应敲入:

RUN386 hi

或

RUN386 hi.exp

应有 Weitek 协处理器时应敲入:

RUN386 -1167 on hi

或

RUN386 -1167 on hi.exp

注:-1167 开关用来选择 386DOS 扩展程序用什么方法去检测 Weitek 协处理器的存在。

-1167 on 告诉 RUN386。假定 1167 或 3167 存在。

示范程序 hi.c 应在屏幕上显示下列信息:

Hi!

如果得到下列信息:

This program requires an NDP type which is not installed on this computer.

则表示由示范程序编译生成的可执行文件与机器上的协处理器不配套。使用正确的协处理器开关重新编译程序,这些开关列在本节的开始部分。checkndp.exp 程序用于检查机器上的所安装的协处理器类型。该程序在 NDP C-386 的第二张软盘上,该程序可以软盘上启动,也可先将其拷贝到一个合适的目录下,再敲入如下命令:

RUN386 checkndp

1.7 编译驱动程序:cc

从源程序到创建一个可执行文件需要经过若干个步骤。首先,NDP C-386 编译程序将源程序翻译成等价的汇编语言形式。然后,再用 Phar Lap 工具程序进行汇编,并将结果与各个目标模块和库连接起来,生成可执行程序。批处理文件可以自动地执行这一过程,但我们为此专门设计了一个程序,称之为编译驱动程序。

NDP C 的编译驱动程序 cc 允许用户控制程序的整个建立过程。用户敲入“cc”,以及需要编译的文件名(一个或多个)和各种执行开关,即可激活编译驱动程序。编译驱动程序调用 NDP C 编译程序、Phar Lap 汇编程序和 Phar Lap 连接程序,以建立所需的可执行程序。

编译驱动程序可放在任一目录下。为了方便起见,该目录通常放到 PATH 指定的系统搜索路径中。一个好的策略是把编译程序和相关文件(驱动程序,目标模块和库)放入同一个目录,然后把该目录包含在搜索路径中。在使用新的编译程序版本时,这样将便于查找和更新的正确文件。

缺省时,编译驱动程序产生的二进制可执行文件带有扩展名 exp。其文件名与第一个源文件名相同。输出文件放入当前目录中。-o 开关允许用户改变文件名和输出文件所在的目录。

1.8 环境变量和编译驱动程序

编译驱动程序工作时需要知道 NDP C 编译程序、Phar Lap 工具程序、目标模块和库的位置。由于这些信息并不经常变动,因此这些文件的目录名可保存到 DOS 环境变量中,从而供编译驱动程序使用。为此,我们设计了五个任选环境变量。下面给出了这些环境变量的名字及其配套的目录:

<u>环境变量</u>	<u>目录内容</u>
NDP	NDP C 编译程序和目标模块
TOOLS	Phar Lap 汇编程序、连接程序和 DOS 扩展程序
LIBP	运行库模块
SRC	源程序
INCLUDE	包含文件

如果没有定义上述环境变量,那么编译驱动程序使用当前驱动器的\ndp20 目录作为缺省目录。另外,当忽略了其他环境变量(除 SRC)时,NDP 环境变量指定的目录作为缺省目录,这样便于从单个目录启动编译程序和 Phar Lap 工具程序。在实际应用中,最好建立三个或三个以上的目录,分别包含:编译程序、库和 Phar Lap 工具程序,并为各源程序组建立一个目录。

下面各小节将详细叙述了每个环境变量。在编译驱动程序中使用 -v 选项可打印出运行各子过程所需的命令,这样,与编译程序和驱动程序使用的搜索路径有关的问题就可以发现了。另外,使用 -rt2 开关可打印出编译程序打开的文件名(详见第 2 章的选项和开关一节)。

1.8.1 编译程序目录和环境变量 NDP

编译程序目录主要用于编译程序的操作。它表示从该处可以找到运行编译程序所需的各种文件。所有相关文件均可放入该目录中,也可分别放入不同的目录中(由环境变量指定)。这在前面部分已讨论过了,下面还将详细讨论。环境变量 NDP 为编译驱动程序指出了下列文件的位置:

ndpc386.exp	dos386.obj
_co0.obj	_co1167.obj
_co287.obj	_co387.obj

在启动时,编译驱动程序确定系统环境中是否包含了 NDP 环境变量。如果有,则假定该变量指定的目录包含了上述文件;否则,编译驱动程序假定上述文件均包含在当前驱动器的 \ndp20 目录中。

1.8.2 Phar Lap 目录和环境变量 TOOLS

TOOLS 环境变量为编译驱动程序提供了下列 Phar Lap 工具程序的位置:

RUN386.EXE
386ASM.EXE
386LINK.EXE

编译驱动程序使用下列目录之一作为 Phar Lap 工具程序的位置:TOOLS 环境变量指定的目录,NDP 环境变量指定的目录或 \ndp20 目录。

首先检查 TOOLS 环境变量,若存在则使用它;其次是 NDP 环境变量,若存在则使用它;最后,若 TOOLS 和 NDP 均未定义,则使用缺省目录——当前驱动器的 \ndp20 目录。

1.8.3 库文件和环境变量 LIBP

若 LIBP 环境变量存在,则向编译驱动程序指定了下列文件的位置:

libc.lib
libf.lib
libm.lib
libc1167.lib
libf1167.lib
libm1167.lib
libm287.lib
libm387.lib

编译驱动程序根据协处理器开关自动地选择正确的库文件。/由库开关 -lname 激活的编译驱动程序将使用环境变量 LIBP 指定的目录(见 2.3 节)。

编译驱动程序使用下列目录之一作为库文件的位置:环境变量 LIBP 指定的目录、环境变量 NDP 指定的目录和 \ndp20 目录。