

目 录

第一篇 NDP 工具及编译器

第一章 概论	(3)
1.1 微型电子计算机系统简介	(3)
1.2 FORTRAN 程序设计语言简介	(3)
1.3 NDP FORTRAN 486/386 简介	(3)
第二章 几种最新 NDP 版本介绍	(5)
2.1 概述	(5)
2.2 3.2.0 版用于 DOS 系统的 NDP 486/386 编译器	(6)
2.3 4.0.2 版用于 DOS 系统的 NDP 486/386 编译器	(22)
2.4 4.2.0 Beta 版用于 DOS 系统的 NDP 486/386 编译器	(28)
第三章 用于 NDP 486/386 编译器的 NDP 工具	(30)
3.1 概述	(30)
3.2 NDP Link	(31)
3.3 NDP Run 和 VM	(40)
3.4 NDP Lib	(42)
3.5 NDP 工具实用程序	(45)
3.6 使用 NDP Tools 的警告和错误信息	(46)
第四章 NDP FORTRAN 486/386 编译器	(50)
4.1 概述	(50)
4.2 运行编译程序	(53)
4.3 使用编译器驱动程序—mf486、386	(62)
第五章 NDP FORTRAN 486/386 的优化处理	(69)
5.1 一般优化	(69)
5.2 速度优化	(71)
5.3 FORTRAN 应用程序的优化	(72)
第六章 NDP 编译程序的接口	(81)
6.1 概述	(81)
6.2 NDP FORTRAN 与其它 NDP 语言之间的接口	(82)
6.3 NDP 语言与汇编语言之间的接口	(87)
6.4 实模式程序的接口	(91)
第七章 移植程序到 NDP FORTRAN 中	(94)
7.1 与其它 Microway NDP 编译器的兼容性	(94)
7.2 字长问题	(94)
7.3 字节顺序问题	(94)

7.4	对齐要求	(95)
7.5	浮点范围和精度	(95)
7.6	汇编语言接口	(95)
7.7	表达式的求值顺序	(96)
7.8	编译优化的非法假设	(96)
7.9	隐含寄存器的使用	(96)
7.10	内存分配假定	(96)
7.11	-OM 和 -OLM 选择项	(97)
7.12	有关源程序级调试程序的问题	(97)
7.13	编译器内存大小问题	(97)
第八章	数字协处理器支持	(98)
8.1	概述	(98)
8.2	编译器的浮点选项—"n"开关	(99)
8.3	数值异常—主动与被动处理(仅限于 DOS)	(100)
8.4	协处理器结构	(102)
8.5	对 IEEE 数值系统的介绍	(105)
8.6	内部 NDP 异常处理函数(仅限于 DOS)	(114)
8.7	定制或编写自己的 NDP 异常处理程序	(117)
8.8	舍入、精度及无穷大控制	(118)
8.9	改进无浮点数开关(-n6)	(119)
第九章	跟踪/调试功能	(121)
9.1	概述	(121)
9.2	调试程序	(122)
9.3	内存保护错误	(122)
9.4	编写一个快速 user_debug 程序	(123)
第十章	错误信息及恢复	(125)
10.1	前端错误	(125)
10.2	后端错误	(125)

第二篇 NDP FORTRAN 库

第十一章	内部函数	(129)
abs		(129)
acos allocated and		(130)
asin atan,atand atan2,atan2d		(131)
btest char cmplx		(132)
conjg cos,cosd cosh		(133)
cotan dble dcmplx		(134)
dfloat dim dprod eof		(135)
epsilon exp		(136)

float hfix huge	(137)
ibchng ibclr ibits	(138)
ibset ichar ifix imag	(139)
index int intl	(140)
int2 int4	(141)
intc ior isha	(142)
ishc ishft	(143)
ishftc ishl jfix	(144)
len len_trim loc,locnear,locfar	(145)
log log10 lshift max	(146)
maxexponent min	(147)
minexponent mod nearest	(148)
nint not or	(149)
precision ran real rshift	(150)
scan sign sin	(151)
sinh sqrt tan,tand	(152)
tanh tiny verify	(153)
xor zext	(154)
第十二章 FORTRAN 库例程	(156)
accept backspace bdos	(158)
beep _bios_disk	(159)
_bios_equiplist _bios_keybrd _bios_memsizes _bios_printer	(160)
_bios_serialcom _bios_timeofday	(161)
_blk_bm _blk_mb	(162)
cgets chdir chmod chsize clear87	(163)
_clock_ticks close clrndpex	(164)
_control87 cputs _cstring	(165)
cvid	(166)
cvif date decode	(167)
dosdat _dos_allocmem _dos_close _dos_creat	(168)
_dos_creatnew _dos_exterr _dos_findfirst _dos_findnext	(169)
_dos_freemem _dos_getdate _dos_getdiskfree _dos_getdrive	(170)
_dos_getfileattr _dos_getftime _dos_gettime	
_dos_keep _dos_open _dos_read	(171)
_dos_setblock _dos_setdate _dos_setdrive	
_dos_setfileattr _dos_setftime _dos_settime	(172)
dostim _dos_write dsab_ex	(173)
dup dup2 enab_ex encode	(174)
endfile erf erfc	(175)
errsns exit fflush_buf	(176)

filelength _freect ftime gamma	(177)
get_addr getarg	(178)
getch getche getcwd getdat get_dos_dta getenv	(179)
geterrno gethost gettim _heapchk	(180)
_heaprun _heapset _heapwalk _hostfree _hostmalloc _hypot	(181)
largc iclrer idate	(182)
igeter _ignore_breaks	(183)
inf_ctl init_ndp inkey\$	(184)
inp,inpd,inpw inquire	(185)
install_pm_isr	(187)
int386 int386x	(188)
int86 int86x	(189)
intdos intdosx	(190)
itoa _itoh,_itoo,_itob j0,j1,jn	(191)
kbhit idndpcw	(192)
ldndpenv lfind locking	(193)
lrotl lrotr lsearch ltoa _makepath mapdev	(194)
mapref memavl memcpy	(195)
memmax _memory_address	(196)
mkdir _msize mvbits nargs ndptype	(197)
note nstndpenv	(198)
num_files open os386_rpc os386_rp_load	(199)
outp,outpd,outpw pause	(200)
pauseb peek,peekb,peekw poke,pokeb,pokew precis	(201)
print printererror putch pxerror	(202)
random read _read_typeahead	(203)
restore_isr ret_ex_hdl rewind	(204)
rm_alloc rmdir rm_free rm_int386	(205)
rm_resize rotl rotr round	(206)
save_isr _searchenv sec_100	(207)
secnds seed segread	(208)
set_buffering setdat seterrno	(209)
set_ex_hdl set_io_defaults settim	(210)
shiftstate	(211)
show_registers sign_of_gamma sopen spawn	(212)
spawnv spawnve spawnvp spawnvpe	(213)
_splitpath stndpcw stndpenv	(214)
stndpsw strdup _strerror	(215)
_strtime system tell tempnam	(216)
time timedate	(217)

type tzset ultoa ungetch _user_break	(218)
utime _watch_breaks write xexit y0,y1,yn	(219)

第三篇 NDP 扩展图形库 GREX

第十三章 GREX 库概述	(223)
13.1 设计思想	(223)
13.2 库的范围	(227)
13.3 库的内容	(230)
第十四章 GREX 库例程	(234)
14.1 图形例程	(234)
address aspect_ratio	(234)
clear	(235)
define_color	(236)
draw filled_ellipse	(237)
filled_rectangle	(238)
flood_fill	(239)
get_active_page get_clip_limits get_color	(240)
get_dash get_device_limits	(241)
get_fill_mask get_display_page get_pixel	(242)
graphic_page_count	(244)
graphic_text	(246)
graphics_mode horizontal_line	(247)
magnify_text	(248)
move move_cursor	(249)
open_ellipse	(251)
physical_screen	(252)
polyfill	(253)
restore_window	(256)
save_window set_active_page	(259)
set_bios_mode	(260)
set_clip_limits set_color set_dash	(261)
set_display_page set_fill_mask	(263)
set_font_8×8 set_font_8×14 set_font_8×16 set_palette	(264)
set_pixel	(267)
set_pixel_function	(268)
set_user_font	(269)
set_xor	(270)
size_window	(271)
super_vga	(272)

text_mode vertical_text	(273)
vga_palette	(274)
video_configuration	(275)
virtual_screen	(276)
_where	(281)
14.2 文本例程	(282)
box_text_window clear_text clear_text_window	(282)
cr create_text_subwindow create_text_window	
crlf display_text_page	(283)
dn_cursor free_text_window get_active_attribute	(285)
get_active_text_page get_cursor get_video_mode	(286)
hide_cursor home lf	(287)
locate	(288)
lt_cursor place_string_at	(289)
read_attr read_char read_text_window	
reshape_text_window restore_text_window	(290)
rt_cursor save_text_window	(291)
scroll_active_page_down	(295)
scroll_active_page_up scroll_text_window	
set25line set43line set_active_attribute	(296)
set_active_text_page set_cursor_type	(297)
set_video_mode up_cursor write_char	(298)
write_char_attr write_one_char	
write_string write_test_screen	(299)
write_text_window	(300)
14.3 端口、块和中断	(300)
inkey \$	(300)
pause	(301)
pauseb	(302)
shiftstate	(303)
beep note	(304)
blk_bm	(305)
blk_mb inp	(306)
inpw int386	(307)
save_isr	(308)
restore_isr	(309)
install_pm_isr mapdev	(310)
outp	(311)
outpw peek peekw peekb poke	(314)
pokew pokeb pxerror	(315)

第一篇

NDP 工具及编译器

第一章 概 论

1.1 微型电子计算机系统简介

一个微型电子计算机系统是由计算机硬件和软件两大部分组成的。计算机硬件包括所有组成计算机系统的设备,最基本的有三部分:(1)输入设备,包括键盘、磁盘机等;(2)中央处理机,包括内存、运算器和控制器等;(3)输出设备,包括显示器、打印机和磁盘机等。计算机软件是指能使计算机系统运行的各种程序和指令,一般地有三类:应用程序、编译程序和操作系统。应用程序是为了解决某个具体问题而编写的程序,比如用 FORTRAN 语言就可以编写一些解决科学和工程问题的应用程序;编译程序是指能把用高级语言编写的应用程序翻译成机器能直接识别的机器语言的程序;操作系统是指导和管理工作运行的一系列程序,例如 DOS 系统。

本书是一本关于软件的书籍,与本书有关的是应用程序和编译程序。

1.2 FORTRAN 程序设计语言简介

FORTRAN 这个词是 FORMula TRANslation (公式翻译)的缩写。FORTRAN 语言是最早出现的程序设计语言。50 年代初期,美国国际商用机器公司(IBM)着手发展高级程序设计语言,形成了 FORTRAN 语言。1958 年,IBM 704 型计算机就可以使用 FORTRAN 语言了,这是最初的 FORTRAN 语言,称为 FORTRAN I。60 年代初期,FORTRAN 语言作为科学和工程计算用的程序设计语言得到了广泛的应用,同时 FORTRAN 语言本身也得到了迅速的发展,逐渐形成了 FORTRAN IV。为了实现通用性,对 FORTRAN 语言进行了标准化,1966 年公布了 FORTRAN 语言的第一个标准文本(ANSI X 3.9-1966),这就是标准化的 FORTRAN 66。随着计算机使用范围的扩大,要求程序设计语言有一定的字符数据处理功能和文件操作功能,因此 1978 年又出现了新的 FORTRAN 标准(ANSI X 3.9-1978),称作 FORTRAN 77。

随着计算机技术的高速发展,1980 年代出现了 32 位的微机 386,486,为了充分利用这些高性能的微机,美国 Microway 公司开发了自己的 NDP 系列语言,其中 NDP FORTRAN 全面实现了 FORTRAN 77 标准,有比较宽的兼容性。由于 FORTRAN 77 已为广大读者所熟知,本书没有详细列出 NDP FORTRAN 语言。

1.3 NDP FORTRAN 486/386 简介

NDP FORTRAN 是 Microway 公司专门为 Intel 80386、80486 微处理器开发的,是 32 位的全局优化编译器。NDP FORTRAN 全面实现了 ANSI X 3.9-1978 (即 FORTRAN 77)标准,还包括了移植程序所需的许多扩展(例 FORTRAN 66、VAX BSD 4.0 等)。它使

用全局数据流的方法，在保护模式下工作，最大限度地利用了 386、486 的 32 位数值寄存器，它产生的 80386、80486 代码长度短而运行速度快，可以在 MS DOS、UNIX V、SCO XENIX 系统下正常运行。

NDP FORTRAN 486/386 的一个突出的优点是能够管理虚拟内存，因此可以突破 640K 内存的限制，只要硬盘许可，就可以编译任意大的源程序，给科学研究工作者提供了极大的便利。

NDP FORTRAN 486/386 有一个内容丰富的图形库 GREX，读者可以仿照源盘提供的演示程序编制自己的图形显示程序。利用扩展图形库 GREX，读者可以编出友好的图形界面，使应用程序更加完善。也可利用图形跟踪计算结果，使科学计算更加直观、更省机时。

本书主要对 3.2.0 版和 4.0.2 版的 NDP FORTRAN 编译器作详细的介绍。各种版本稍微详细一点的情况以及它们之间的区别请参见下章的内容。

第二章 几种最新 NDP 版本介绍

2.1 概述

带有 NDP Tools 的 NDP 编译器可以用来编译、连接以及运行 80486 和 80386 保护模式代码,而不需要其它的 DOS 扩展程序或工具。当然,也可以继续使用 Phar Lap 开发工具。

使用 Phar Lap Tools 的开发过程包括编译、连接和运行(以及调试,如果必要的话)。将源程序编译成目标文件需要合适的编译器:NDP C、NDP C++、NDP FORTRAN 或者 NDP Pascal。用编译器驱动程序就会自动地编译和连接源代码。

使用 NDP Link 可以连接编译器生成的目标文件;如果所有的符号都已得到解释,连接过程将产生一个扩展名为 .ltd 的文件。用 NDP Run 可以运行这个 .ltd 文件。要进行 .ltd 文件的源程序级调试需要选用 ClearView-386。

2.1.1 基本特点

- 在 DOS、UNIX 和 XENIX 系统上都可以使用这些编译器。若用 DOS 版本则需要 NDP Tools 或 Phar Lap 开发工具。在使用 Microway 工具时,使用 NDP VMEM 就可以支持虚拟内存;在使用 Phar Lap 开发工具中,选择 VMM 就可以支持虚拟内存。
- 几种编译器共享一个公共代码生成程序和后端;特定语言的特点仅仅在前端的寻址上有所不同。
- 编译器包含 NDP Tools。NDP Tools 又包含 NDP Link、NDP Run 和 NDP Lib。如果你想形成一个 .exe 文件,则需要联结一个第三种 DOS 扩展程序,这个 DOS 扩展程序需要运行许可证。想了解详细的情况可以与 Microway 的技术支持部门联系。
- 目标模块生成程序(OME—Object Module Emitter)包含在每一个编译程序中。OME 直接从源代码产生目标代码,而不需要第三种汇编程序。编译程序也能产生汇编代码。
- 编译器有两个系列库,一个是对应 NDP Tools 的,另一个是对应 Phar Lap tools 的。这些库按层次排序以便连接程序一次通过每个库。
- 所有的编译程序包含协处理器 Intel、Cyrix(Intel 模式)和 Weitek 的支持程序。

2.1.2 386 编译器特点

NDP 386 编译器产生 80386DX 代码,这些代码也能在 80386SX 和 80486 上运行。NDP 386 编译器不能生成 80486 的优化代码,产生的 386 代码在 80486 上运行也不会象 486 编译器产生的代码那么快。

2.1.3 486 编译器特点

486 编译器是一个完整开发系统的主要部分,它们能帮助用户开发各种平台程序。NDP

486 编译器能够产生 486 特定的代码,这种代码被排列以优化 486 使用流程(使之满负荷)。NDP 486 编译器也能够产生 386DX/386SX 的优化代码,这种代码与 NDP 386 编译器产生的代码相同。

2.1.4 486 代码优化

Intel 80486 是 100% 的二进制代码,与 Intel 80386 兼容。任何的 8086/8088、80286 和 80386 应用程序都可以不经修改地在 80486 上运行,而且也不需要重编译、重连接或重链接。

80486 性能胜过 80386,许多 80386 指令在 80486 上运行得更快,典型的是在一个时钟周期内。然而,有一些使程序在 486 上运行更快的优化当在 386 上运行时会有少量的性能损失,这些优化包括 486 的交替指令、指令序列选择以及充分利用 80486 内部管路单元的软件指令调度。

在缺省情况下,NDP 486 编译器产生 80486 CPU 的目标代码,这种代码能在 386 上运行,但是可能会有代码大小和速度的损失。编译程序也支持 16 字节校准的 80486 优化和避免预取队列缺乏的 NOPS。

80486 有一个 8KB 的 16 字节线的超高速缓冲存储器。如果一个转移的目标接近超高速缓冲存储器线的尾端,超高速缓冲存储器故障就可以避免。因此程序是 16 字节定位。如果允许第一个操作(作为一个单元)跨在段的边界,标号就是段定位的,调用指令被移到段的第 11 字节以避免下一个指令跨在一个 16 字节的边界。

为了防止预取队列的缺乏,仅有单个时钟存储器(或寄存器)移动指令的段组将尽可能早地使 NOP 放入每一组中。

2.2 3.2.0 版用于 DOS 系统的 NDP 486/386 编译器

2.2.1 新的特点

- 新的环境变量: 可以设置环境变量 NDPSW,将经常要在驱动程序命令行输入的任何开关包含在内。例如,如果总是使用 Weitek 协处理器进行编译和使用 Phar Lap 工具,可以设置 NDPSW=-phar1 -n4。
- 若同时使用 Phar Lap 汇编程序和 NDP Tools,而它们又不在同一目录下时,可以设置环境变量 386ASM 以指示二者的目录。
- 选择 Cyrix EMC 支持: 与 Intel 兼容的 Cyrix EMC 协处理器芯片有一个特别的模式,在这个模式中它要求速度要与 Weitek 协处理器芯片在同一个范围。Microway 公司是第一家提供支持 EMC 协处理器的编译程序的公司。为了使用这个模式,可以把开关-emc 加到命令行中,还必须有选择库和启动模块。
- 联机帮助: 敲入编译器驱动程序的名字(例如 mf486),然后按回车键时,所有开关的简单描述就会显示在屏幕上。
- Phar Lap 兼容性: 现在的 NDP 编译器既支持 3.0 版的 Phar Lap Development Tools 也支持先前的版本。在此以前的 NDP 编译器与 Phar Lap 2.2D 以及更早的版本是兼容的。

- 驱动程序开关： 驱动程序开关—time 可以在屏幕上给出编译过程的时间。
- 压缩的 .lrl 文件： NDP Tools 可以产生压缩的和不压缩的可执行文件。压缩的文件占比较少的磁盘空间。
- 虚拟内存： NDP Tools 在缺省的情况下就包含了使用虚拟内存的能力。
- 驱动程序开关： 若选择了驱动程序开关—run，则当可执行文件—产生驱动程序就执行它。

2.2.2 变化

- SX 编译程序： SX 版本的 NDP 编译器已经停止使用，用户拥有的 SX 版本可以被提升到 386 版本。
- 连接速度： NDP Tools 已经被改进，其速度可以达到与 Phar Lap Development Tools 相比拟的程度。
- Plum Hall： NDP C 全部通过了 Plum Hall Validation Suite。
- 包含文件： 在 NDP C 中，ANSI 和 non-ANSI 的包含文件被合并在一起，仅有一个包含目录。
- 速度： C 语言库中一些字符串处理函数得到了改进，这改善了 Dhrystone 基准程序，从每秒 1,8000 Dhrystone 提高到每秒 30,000 Dhrystone，这个变化影响所有的 NDP 编译程序。
- 驱动程序名字： 下表列出编译器驱动程序的名字，从中可以找出驱动程序的新名字。

旧名字	NDP386	NDP486
mf77	mf386	mf486
mcc	mc386	mc486
mpc	mp386	mp486

- NDP Lib 命令行： NDP Lib 不再是交互式的，它带有命令行参数和间接文件，同 Phar Lap 库管理程序 386LIB 相似。NDP Lib 也支持通配符(*.obj)。
- 子目录层次结构： 安装程序自动地在用户指定的 NDP 目录下建立下列的子目录结构：

文件	所在目录
编译器和驱动程序	\NDPx86\BIN
库文件	\NDPx86\LIB
包含文件	\NDPx86\INC
例子	\NDPx86\SRC
工具	\NDPx86\TOOLS

- 环境变量： 当某个环境变量没有设置时，子目录的寻找会发生变化。下表列出了先前的和现在的缺省情况。仅仅当问题中的环境变量没有设置时，这种变化才有影响。

环境变量	旧的缺省	新的缺省
NDP	NDP	错误信息
TOOLS	NDP 目录	NDP 目录下的 TOOLS 子目录
LIBP	NDP 目录	NDP 目录下的 LIB 子目录
INCLUDE	NDP 目录	NDP 目录下的 INC 子目录
SRC	当前目录	当前目录

- 例程作了改动以反映编译器的改进。
- Microsoft 的与 C 兼容的库 LIBD*.LIB 被合并入别的库,在编译或连接时不再必要为了得到 Microsoft C 函数而包含某个库的名字。
- Easy-Driver 已经不再继续使用。
- 在一个程序执行的末端报告下列 NDP 异常情况:

下溢(Underflow)

不正常的操作数(Denormalized Operand)

这两种情况都与单精度数和 $1.0E-307$ 有关,当一个数非常接近零时,在协处理器中这个数就失去精度。对于大多数用户来说,这并不重要,可以通过在程序末端调用 Clrndpex()来移走这种情况。

2.2.3 故障处理

- 以前的 NDP VMEM 遇到在硬盘上分配额外内存空间的困难,这一点现在已经得到改进。
- 内部编译器错误: NDP 编译器不再容易出现内部编译器错误 1034、2020、2030 或 2100。
- FORTRAN 函数: FORTRAN 函数 STATEMENT、NAMELIST 以及 INQUIRE 的问题已经得到修正。
- MIKE: NDP Tools 提供的保护模式 MIKE 版、宏汇编编辑程序(macro-EMACs editor)现在可以工作。
- 改变了驱动程序,以便当编译过程出现问题时使汇编和连接不再进行下去。
- NDP FORTRAN 的 STOP 和 CALL EXIT 不再将 DOS 错误级别设置为 1。
- NDP FORTRAN 的 LOGICAL 变量的缺省格式得到了改进。

2.2.4 在最近一些版本中的变化

3.1.0 版本

- 源程序级的调试程序支持: 在 1990 年 10 月份,可选的调试程序 ClearView-386 就已经有售。使用 ClearView-386 需要 NDP Tools。这个调试程序可以处理 80386 和 80486 代码,它是为 FORTRAN 和 C 设计的。
- 简表文件(Profiler)支持: 和 ClearView-386 在一起的简表文件与 UNIX 的应用程序 PROF 相似。
- 虚拟内存支持: NDP VMEM 允许在使用 NDP Tools 进行编译和运行时支持虚拟内存。Phar Lap tools 的 1.4f 以上版本的编译器也支持虚拟内存。
- 驱动程序控制: 如果加上开关 -keep (不删除任何临时文件)、-keepobj (不删除 .obj 文件)和 -keeps (不删除汇编文件),在编译过程中不论发生了什么错误都不再删去临时文件。
- 开关: 关于开关 -g 和 -i2 的问题得到了解决。
- 函数: 对函数 EXIT、rename、remove 和 time 作了一些修正。
- _NDPC_: 符号 _NDPC_ 现在在缺省情况下是一个定义的宏功能(仅用于 NDP C)。

- NDP 库的名字： 下列命名结构被 NDP 编译器使用的库所采用，以便在同一子目录中允许多平台的存在：

LIB +	所有库的开头
{ <i>language</i> } +	F(ortran), C, M(ath), G(raphics), D(MS-C)
{ <i>platform</i> } +	3(80386), 2(80287), 4(80486), 8(80860)
[{ <i>linker</i> } +]	H 是 Phar Lap Linker, 空缺是 NDP Tools
[{ <i>coprocessor</i> } +]	W 是 Weitek 协处理器, 空缺是 Intel 协处理器, C 是 EMC
.LIB	所有库的扩展名

例如, LIBM3H.lib 是 NDP 编译程序的数学库, 3 是 386 中的 3, H 表示 Phar Lap 连接程序。

- 直接插入汇编语言(仅限于 C 语言)。
- Weitek 支持： 检查是否有这样的问题，一些计算机的 BIOS 只返回 16 位的关于 Weitek 协处理器的信息，同时 NDP Tools 却希望找到 32 位的信息。在 Tools 的磁盘上提供了两个新的文件 TEST11.EXE 和 WEITEK11.EXE, TEST11.EXE 确定你的计算机是否有这样的问题，如果有这样的问题，在 AUTOEXEC.BAT 中加入 WEITEK11.EXE 将解决这个问题。
- 内部编译程序错误： ICE 1739 已经作了修正。
- 目标文件： 文件 LIBF.OBJ 和 LIBF1167.OBJ 不再在分配磁盘上提供，取而代之的是提供能构造这些文件的批处理文件，Utilities 和 Examples 磁盘上的文件 LIB2OBJ.BAT 可以用于这个用途。

3.0.0~3.0.4 版本

- 运行许可证的选择： Microway 引入对 ErgoDos extender 的支持，从而允许有意创建.exe 文件的用户在 Phar Lap 和 Ergo 之间选择一种运行许可证。
- NDP Tools： Microway 开始将 NDP Compilers 和一个连接程序(NDP Link)、库管理程序(NDP Lib)、Phar Lap 的 Intel 库转换程序(PLI)以及加载程序/DOS extender(NDP Run)结合在一起，以便使 NDP Compiler 独立于第三类工具。Microway 将继续支持 Phar Lap Development Tools。
- MiniMake： 这个菜单程序已经停止了继续使用。
- 486 支持： 在 NDP486 编译程序的工作中引入了充分利用 Intel80486 处理器的优化处理。拥有 386 编译器的用户可以得到版本升级，要了解详细情况可以与 Microway 的技术支持部门联系。

2.0.0~2.1.0 版本

- 图形：LIBFGREX 和 LIBCGREX 被 LIBGREX 代替。重编了所有的图形程序。
- Pascal：在 NDP 产品序列中加入了 NDP Pascal。
- 标准： NDP FORTRAN 的 VAX/VMS 标准支持改进到 99% 的 VAX/VMS 扩展。NDP C 加入了对推荐的 ANSI 标准的支持。
- 开关 -g 以前有一个跟踪调试路数的问题，这个问题已经得到解决。
- 开关 -i2 在 FORTRAN 中以前有一个与 INTEGER * 2 有关的问题，这个问题也已

经得到解决。

2.2.5 连接文件说明

如果使用自己的连接文件,应该注意这样的问题,开关、模块和库的顺序是重要的。为了搞清连接文件的正确格式,用 `-v` 开关编译测试程序, `-v` 开关可以保留连接文件 (NDPTMP.LNK),而这个连接文件通常是被删除掉的。用这个连接文件作为你自己连接文件的样板。

2.2.6 对系统的要求

用 NDP 编译器时,编译和运行代码需要:

- 安装在与 IBM 兼容的 PC 机上的一个 Intel80486、80386 或 80386SX 处理器。
- 为了进行浮点数运算,需要一个与 CPU 兼容的数值协处理器,例如:结合在 80486 上的 FPU、Intel 80387、80387SX 和 80287;Weitek4167、3167 和 1167;Cyrix CX-83D87 或 CX-EMC887;或者任何与此兼容的协处理器。在编译时,不需要任何协处理器。如果你的代码不进行浮点运算,使用 `-co0` 开关。
- 一个 1.2MB 5.25 英寸的磁盘驱动器(如果需要可以使用 3.5 英寸磁盘)。
- 有自由空间的硬盘,自由空间的具体数量要在安装时确定,并且要根据你安装的是编译器的什么部分。
- 至少 2MB 的扩展内存,推荐为 4MB,这是超出 DOS 限制(640KB~1MB)的。
- MS-DOS v3.2 或者更高版本。

2.2.7 磁盘内容

编译程序总共有 7 张盘: 编译程序和包含文件(3 张盘)、NDP 应用程序和例子、NDP 库(3 张盘)、带虚拟内存的 NDP Tools 和 MIKE。

编译器的磁盘标明所用的语言(C, C++, FORTRAN 或 Pascal) 和平台(486 或 386), 每个编译器磁盘包含下列文件:

DISK.ID	安装程序使用的磁盘标记。
INSTALL.DAT	安装数据文件(只在第一张盘上)。
INSTALL.EXE	解开压缩文件 NDP.001 的安装程序(只在第一张盘上)。
NDP.00x	压缩状态的编译程序文件。

NDP Examples 盘包含了许多各种语言的源程序例子,这些文件包含了数值异常处理程序、不同语言的混合代码以及 GREX 图形库的例子。它安装下列文件:

bin\pmem.*	报告可用的保护内存的数量。
bin\checkndp.*	报告安装的是何种数值协处理器。
bin\msmw.exe	把 Microsoft FORTRAN 文件转换成能被 NDP FORTRAN 读的格式。
bin\87r.com	显示 80x87 寄存器内容的一个 TSR。
bin\lib2obj.*	转化一个库成为它的组件.OBJ 文件的程序。
bin\test11.exe	报告调用设备状态的 BIOS 返回的位数。
bin\weitek11.exe	一个补充支持 weitek 的 BIOS 的 TSR。
src\examples1*.*	在 NDP 编译程序中包括错误处理在内的各种特点的例子。
src\grex*.*	在 GREX 图形库中可用的大部分功能的例子。

NDP Tools 磁盘安装的文件：

tools\ndplib.exe	库管理程序。
tools\ndplink.exe	保护模式连接程序。
tools\ndprun.exe	加载程序。
tools\ndprunvm.exe	带虚拟内存的加载程序。
tools\test11.exe	诊断应用程序,确定使用 Weitek 协处理器是否需要程序 weitek11.exe。
tools\weitek11.exe	在一些计算机上使用 Weitek 协处理器需要的 TSR 程序。
tools\LOM.exe	“目标模块列表”程序,给出关于目标模块的信息并且检查它们是否出错。
tools\PLI.exe	“Phar Lap—Intel”转换程序,把库从 Phar Lap 格式转化成 Microway 格式。

MIKE 是一个象 MicroEMACS 一样的文本编辑程序,有 16 位和 32 位两种版本,32 位版本允许在保护模式下编辑文件。磁盘安装下列文件：

bin*.obj	重新创建 MIKE.LTL 文件所需的目标文件。
bin\MIKE.LTL	编辑程序的可执行代码。
doc*. *	在重新创建 MIKE.LTL 文件时要用到的文件。
src*. *	重新创建 MIKE.LTL 文件所需的源文件。

2.2.8 安装

为了编译程序,必须安装编译器、库文件和 NDP Tools (包括在内)或者 Phar Lap Tools 磁盘(不包括在内)。对于实现编译器的功能来说,不需要安装 Utilities 和 Examples 以及编辑器 MIKE。

下列的指令假设使用的软盘驱动器为 A 驱动器。如果你使用的是一个不同的软盘驱动器,可以用这个驱动器的符号取代指令中的“A:”。

将第 1 号编译程序盘插入 A 驱动器并且键入 A:INSTALL

然后按屏幕的提示进行。编译程序的三个拷贝将被安装在 NDP 目录下的 bin 子目录中。当编译时不使用虚拟内存时,使用 .exe 版本;当使用虚拟内存,而且用 Phar Lap Tools 编译时,使用 .exp 版本;当要使用虚拟内存,而且用 NDP Tools 编译时,使用 .ltl 版本。如果其中的哪一个不适合你,可以移开编译程序的那个版本以节省硬盘空间。

重复这个过程来安装库文件,如果需要的话,还可以安装 NDP Tools ,Utilities ,Examples 和 MIKE 。Examples 包含 FORTRAN 、C 和 Pascal 的源代码,为了节约硬盘空间,可以删除不适合你的语言的例子。

2.2.9 设置

在运行编译器之前,必须用 DOS 的 SET 命令设置 NDP 环境变量：

SET NDP = *path*

这里 *path* 是完整的路径名,包括安装的编译程序所在目录的磁盘驱动器。如果你使用的工具不在 INSTALL 程序建立的目录下,应当也设置 TOOLS 环境变量以便指出那个目录。

SET TOOLS = *drive:\tools_directory*

可以用不带任何变元的 SET 命令检查环境变量的设置。SET 命令可以放到批处理文件中以简化这个过程。