

C语言基础上的一种目标化高级程序设计语言

ZortechC++ 用户参考手册

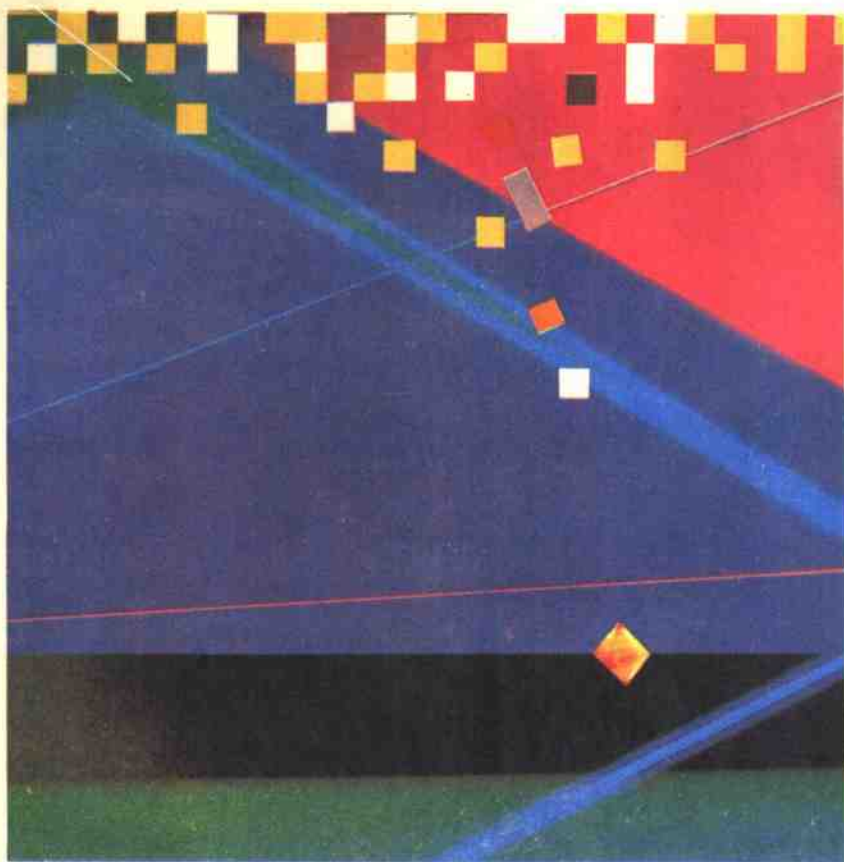
(V2.0)

Walter Bright 著

冯广聚

译

王玉萍



HOPE

中国科学院希望高级电脑技术公司

1

185x255

Zortech C++ V2.0

用户参考手册

冯广聚 译
王玉萍

中国科学院希望高级电脑技术公司
一九九一年二月

15.00

版权所有
翻印必究

■ 北京市新闻出版局

准印证号：3206—90206

■ 订购单位：北京8721信箱资料部

■ 邮 码：100080

■ 电 话：2562329

■ 传 真：01—2561057

■ 乘 车：320、332、302路
车至海淀黄庄下车

■ 办公地点：希望公司大楼一楼
往里走101房间

译 者 的 话

ZortechC++V2.0终于与读者见面了,它可用于MSDOS, OS/2, Microsoft windows系统,在C++1.07版本推出后,经历了很长的一段时间,在这期间, 有两次想推出中间版本, 但最后认为用户所需要的是一套完整的C++, 所以, 一直在C++2.0版准备好后,才正式推出。

作为程序设计语言, C++2.0是一套成熟的语言。在其中,加入了很多新功能,删除了其中的 余部分,其中最著名的新功能是多重继承 (multi-inheritance)和类型安全联接 (type-safe linkage)。

C++2.0可以在多用户工作台上操作。由于C++2.0已趋完善,所以在以后两年内不会有太多的改变。同时,对此语言进行标准化的ANSI委员会也正式成立。总之,对此语言的开发已告一段落,它支持多售主, (Multi-Vendor) 和多用户工作台 (Multi-Platform)。

越来越多的用户正在使用此语言,由此可以保证对C++编译器进行不断地完善和提高。

对于MSDOS系统, ZortechC++2.0根据MSDOS系统的实际情况,包会几个扩充。如:近程、远程、栈、和句柄指针类型,还支持近程、远程、C、C++和PASCAL函数类型。这些扩充允许在扩展内存中直接使用和处理数据。

1989年, Microsoft Windows作为一个编程环境非常流行。Windows编程强烈要求C++的面向目标功能,用以管理其复杂性。所以在C++2.0编译器中,增加了支持Windows的功能。在OS/2和Presentation Manager上,亦可以使用此编译器,它提供了一个分类库以保证在Windows和Presentation Manager之间非常容易地进行代码传送。

Walter Bright

前 言

如果你是C语言的使用者,那么你一定会喜欢Zortech C++ V2.0!

Zortech C++是在C语言基础上开发的一种目标化高级程序设计语言.

本书是Zortech C++一套四本中的一本, (还有用户手册, 安装和调试手册, C++ 工具C++ Tools), 详细介绍了C++的库函数及其使用; 如何编写内存驻留(TSR)程序; 如何联接鼠标, 等等. 为了便于你学习掌握, 你可以把书中的例子输入计算机中运行.

本书由冯广聚, 王玉萍合译, 同时得到了李昆等同志的大力帮助, 在此表示感谢. 由于译者水平有限, 加之时间仓促, 会有许多疏忽和错误之处, 欢迎读者指正.

目 录

1 函数库介绍.....	1
1.1 zortech 库摘要.....	1
1.1.1 C++标准库.....	1
1.1.2 C标准库.....	1
1.1.3 快速图形库.....	2
1.2 印刷规范.....	2
2 Include 文件, 全局变量和类型.....	3
2.1 Include 文件.....	3
2.2 全局变量.....	11
2.3 标准类型.....	12
3 TSR函数包.....	14
3.1 介绍.....	14
3.1.1 TSR的回顾.....	14
3.1.2 TSR如何工作.....	15
3.1.3 TSR能作什么, 不能作什么.....	15
3.1.4 驻留中的驻留.....	16
3.2 已提供的文件.....	16
3.3 写你的TSR驻留程序.....	17
3.3.1 使你的程序驻留.....	17
3.3.2 调试(DEBUGGING).....	18
3.3.3 消除你的程序.....	19
3.3.4 全局变量.....	20
3.3.5 特殊函数 popmain.....	22
3.4 限制.....	22
3.4.1 一个完整的例子.....	22
3.5 问题.....	25
4 库函数.....	26

第一章 函数库介绍

*Zortech C++*包含三个实时程序库集. 其中两个集是必须存在且必须与所有的*c++*程序联接. 这些库由*Zortech*编程系统自动处理, 而且不必对*ZTC*, *ZED*, *BLINK*说明, 每一个库集由四个库构成, 每一个对应一种模式, 微模式除外, 它与小模式使用相同的库. 这些必存库集是*c++*标准库(*p1s.lib*等等)和*c*标准库(*z1s.lib*等等). 还有一个附加库*fg.lib*. 它提供快速图形例程, 且存贮模式独立. 这一个库必须分别向*ZTC*, *ZED*或*BLINK*指明.

开发者版本包含一个附加的实时程序库集, 它包含*C++TOOLS*. 同样, 每一个库对应一种存贮模式, 微型模式除外, 如果使用由*C++TOOLS*提供的工具, 这些库就必须向编译器指明.

*Zortech*库(不包含*C++Tools*)总共提供了超过400个函数为程序员所使用. 由于*C++*程序员依赖标准库来提供很多功能, 而它本身语言中不具有. 所以这些函数尤其重要.

一个特别的标准库已编译版本对于在*OS/2*上运行作为*OS/2*的功能提高是可行有效的.

1.1 Zortech库摘要

1.1.1 C++标准库:

这个库必须与每一个*C++*程序连接, 它由*Zortech*编程系统自动处理, *C++*标准包含*C++*流的执行, 及很多*C++*编译器帮助例程, 对于*C++*流库更详细的说明, 请参阅用户手册中*C++*流一章.

1.1.2 C标准库

*C*标准库是为了与*MSDOS*, *UNIX*系统及*ANSI*标准*C*编译程序兼容而设计的, 这个标准库包含很多函数, 这些函数允许直接被访问到*MSDOS*的环境中, 其中, 很多函数通过特别程序包提供. 下面是已提供的大多数程序包的概况:

BIOS函数包:

这个函数包提供一个接口与由*IBMPC BIOS*提供的功能相连接. 包括磁盘功能, 时间日期控制, 串并口连接和其它例程.

显示函数包:

在这个包中提供一个丰富的高性能屏幕显示函数集, 这个包对于文本属性控制, 画框, 屏幕存贮及恢复功能和其它很多屏幕处理功能采取直接屏幕写技术.

DOS函数包:

除了一个与*DOS*操作系统调用的接口以外, 它与*BIOS*函数包相似. 这些函数在运行*MSDOS*的机器上(*OS/2*及*OS/2*升级版本中)是可以移植的. 还提供很大范围的功能, 特别适于磁盘和文件存取功能.

EMM函数包:

此函数包含一个函数集, 用于分配和扩展(*EMS*或*LIMS*)内存. 它作为一种相对句柄指针类型内部编译器的选择. 对于更详细的说明请参阅用户手册中句柄指针类型一章.

远程函数包:

此函数包提供一个函数集合, 函数用于处理远程堆栈, 特别在小模式和中模式状态下使用. 这样数据能够分配在常规64K数据段的外部.

句柄函数包:

此函数包允许句柄存储器分配(现流行EMS存储器)处理指针, 查阅用户手册中句柄指针一章可得到更详细的内容.

中断函数包:

在装载和消除中断处理程序中, 这个函数包提供给程序员很大的方便. 处理程序包括准备内部栈处理和链连已存中断.

鼠标函数包:

此函数包提供一个与Microsoft相兼容的鼠标驱动程序接口. 鼠标驱动程序允许鼠标控制, 位置报告, 按钮状态和计数, 灵敏度, 光标状态和控制, 及装入一个鼠标中断处理程序的功能.

页函数包:

此程序包允许程序员从任意源存储区中取出一存储块转为栈, 在其中存储区能被自动状态地分配和释放.

声音函数包:

函数用于产生声音, 可以通过IBM PC内部发音器控制声音的持续时间和节距.

时间函数包:

函数用于获取和处理以任何方式表示的时间和日期.

TSR驻留函数包:

在小和微模式状态下程序可以简单容易地转为驻留程序, 通过使用这个库上的例行程序, (大多数并不全部), 标准库函数能在驻留程序中使用.

1.1.3 快速图形库:

库中函数用于实现Zortech图形功能. Zortech快速图形是一高性能图形系统, 提供一套丰富的图元集. 使用这些例程可建立完善的图形系统, 而且在很大范围内支持不同的显示器.

1.2 印刷规范

在本书中, 采用了很多方式来表达内容. 符号"#"用于强调重点.

第二章 Include文件 全局变量和类型

2.1 Include文件

这一节详述Zortech C++标准头文件的内容. 属于ANSI说明中的头文件亦有标记, 但这并不意味着在此文件中说明的函数是ANSI函数, 参考各自的描述来确定是否包含在ANSI C说明中.

2.1.1 assert.h

定义assert宏, assert的定义依赖于ANSI标识符NDEBUG是否已被定义. 如果已定义, 则assert宏作为空文本定义.

2.1.2 bios.h

包含对BIOS函数包的结构定义和函数说明.

下列结构定义与_bios_disk函数一起使用:

struct diskinfo_t

bios.h文件包含下列函数:

<i>_bios_keybrd</i>	<i>bioskey</i>	<i>_bois_equiplist</i>
<i>_bios_memsize</i>	<i>_bios_printer</i>	<i>_bios_serialcom</i>
<i>_bios_timeofday</i>	<i>_bios_disk</i>	

2.1.3 conio.h #ANSI

此文件包含控制台和I/O端口例程函数:

getche getch kbhit
inp outpi npw
outpw

2.1.4 ctype.h #ANSI

此文件包含有关字符分类及转换的宏定义. 这些宏是:

isalnum isalpha iscntr
isdigit isgraph islower
isprint ispunct isspace
isupper isxdigit isascii
toascii

ctype.h还包含下列函数原型:

toupper tolower

2.1.5 direct.h

此文件包含下列函数:

chdir *mkdir* *rmdir*
getcwd

2.1.6 disp.h

这个文件说明全局变量和组成DISP函数包的关联函数.

下列为全局变量说明,并认为是只读的:

disp_mono *disp_snowycga* *disp_mode*
disp_initd *disp_ega* *disp_base*

下列函数包含在disp.h中:

disp_printf *disp_getmode* *disp_getattr*
disp_putc *disp_open* *disp_box*
disp_close *disp_usebios* *disp_move*
disp_flush *disp_eeol* *disp_eeop*
disp_startstand *disp_endstand* *disp_setattr*
disp_setcursortype *disp_pokew* *disp_scroll*
disp_setmode *disp_peekbox* *disp_pokebox*
disp_set43 *disp_reset43* *disp_fillbox*
disp_hidecursor *disp_showcursor* *disp_peekw*

2.1.7 dos.h

此文件包含用在MSDOS接口函数中的全局变量说明,类型和宏定义,函数说明,还包含*inp*, *inpw*, *outp*和*outpw*的内部版本.

全局变量说明如下:

_psp *_osmajor* *_osminor*
_osmode *_doseerrno*

此文件还包含下列结构说明:

struct WORDREGS
struct BYTEREGS
union REGS
struct SREGS
struct FIND
struct DOSError
struct dos_date_t
struct dos_time_t

宏定义是:

FP_SEG *FP_OFF* *MK_FP*

*dos.h*还包含下列函数说明:

int86 *int86x* *intdos*

<i>intdosx</i>	<i>segread</i>	<i>findfirst</i>
<i>_farptr_norm</i>	<i>_farptr_fromlong</i>	<i>_farptr_tolong</i>
<i>farmalloc</i>	<i>farcalloc</i>	<i>farrealloc</i>
<i>farfree</i>	<i>farcoreleft</i>	<i>dos_set_verify</i>
<i>dos_set_ctrl_brea</i>	<i>dos_get_verify</i>	<i>dos_get_ctrl_break</i>
<i>dos_abs_disk_read</i>	<i>dos_abs_disk_write</i>	<i>dos_getdiskreespace</i>
<i>dos_setblock</i>	<i>dos_setdrive</i>	<i>dos_getdrive</i>
<i>dos_setftime</i>	<i>dos_getftime</i>	<i>dos_setfileattr</i>
<i>dos_getfileattr</i>	<i>dos_exterror</i>	<i>dosexterror</i>
<i>dos_setdate</i>	<i>dos_getdate</i>	<i>dos_settime</i>
<i>dos_gettime</i>	<i>_chkstack</i>	<i>peek</i>
<i>poke</i>	<i>bdos</i>	<i>bdosx</i>
<i>dos_alloc</i>	<i>dos_calloc</i>	<i>dos_free</i>
<i>dos_avail</i>		

2.1.8 emm.h

这个文件包含对EMM函数包的结构定义和函数说明:

结构定义如下:

```
struct emm_handle_s
```

下列函数被说明:

<i>emm_init</i>	<i>emm_getunalloc</i>	<i>emm_gettotal</i>
<i>emm_allopages</i>	<i>emm_maphandle</i>	<i>emm_savepagemap</i>
<i>emm_restorepagemap</i>	<i>emm_physpage</i>	<i>emm_term</i>
<i>emm_gethandlecount</i>	<i>emm_deallopages</i>	<i>emm_gethandlespages</i>
<i>emm_getversion</i>	<i>emm_getpagemap</i>	<i>emm_setpagemap</i>
<i>emm_getsetpagemap</i>		

2.1.9 errno.h #ANSI

这个文件包含对全局变量*errno*的说明和定义值.

2.1.10 fcntl.h

这个文件包含与Microsoft相兼容的用于*open*函数的读写方式定义.只有需要使用附加方式时才需要此头文件, 否则*io.h*就足够了.

2.1.11 fg.h

此文件定义类型和宏, 为快速图形函数包说明全局变量和函数.

全局变量说明如下:

<i>fg_displaybox</i>	<i>fg_charbox</i>	<i>fg_ncolormap</i>
<i>fg_nsimulcolor</i>	<i>fg_pixelx</i>	<i>fg_pixely</i>
<i>fg_activemap</i>	<i>fg_displaypage</i>	<i>fg_numpages</i>

fg_display fg_color[]

另外,类型定义如下:

fg_coord_t fg_box_t[] fg_pbox_t
fg_const_pbox_t fg_line_t[] fg_pline_t
fg_const_pline fg_color_t fg_handle_t

还有一些宏定义如下:

fg_coord_midpoint fg_box_height fg_box_width
fg_matrix_size fg_box_area fg_make_line
fg_line_horiz fg_line_vert fg_make_line
fg_box_cpy fg_setcolomum

最后,包含在*fg.h*中的函数说明如下:

fg_assert fg_save fg_init_all
fg_init_herc fg_init_egaecd fg_init_egamono
fg_init_egacolor fg_init_vga11 fg_init_vga12
fg_init_vga13 fg_init_vegahires fg_init_cga
fg_init_cgamedres fg_init_toshiba fg_init_8514a
fg_box_close fg_pt_inbox fg_getfillside
fg_linepixels fg_line_cpy fg_llineclip
fg_setenv_variable fg_setpalette fg_fillpolygon
fg_restore fg_growbox fg_drawlineclip
fg_drawlinepclip fg_drawpolygon fg_filloutline
fg_puts fg_adjustxy fg_drawellipse
fg_fill fg_traveseline fg_drawthickline

2.1.12 float.h #ANSI

此文件包含常量的定义. 这些常量用于指明浮点类型的处理范围.

2.1.13 handle.h

此文件包含与句柄指针处理相关联的函数说明.

函数说明如下:

handle_malloc handlecalloc handle_realloc
handle_strdup handle_free

2.1.14 int.h

此文件包含用于中断函数包的函数说明和用于传送信息到中断服务程序中的结构定义.

结构是: *struct INT_DATA*

函数说明如下:

int_getvector *int_setvector* *int_setvecsize*
int_restore *int_off* *int_on*

2.1.15 io.h

此文件包含大多数低级文件处理和I/O函数的函数说明,说明如下:

read *write* *getDS*
open *creat* *close*
locking *unlink* *remove*
access *lseek* *filesize*
filelength *isatty*

2.1.16 limits.h #ANSI

此文件包含常量定义, 这此常量用于规定数据类型处理范围

2.1.17 locate.h #ANSI

包含与国家(地区)有关的信息.

2.1.18 math.h #ANSI

此文件包含被函数 *matherr*所使用的, *exception*结构定义, 数学函数说明, 及与显型 (*manifest*)常量相关的数学定义. 函数说明如下:

acos *asin* *atan*
atan2 *cos* *sin*
tan *cosh* *sinh*
tanh *frexp* *ldexp*
log *log10* *modf*
pow *sqrt* *ceil*
fabs *floor* *fmod*
matherr *atof* *hypot*
poly

2.1.19 msmouse.h

此文件包含用于鼠标函数包的函数说明, 说明如下:

msm_init *msm_term* *msm_showcursor*
msm_hidecursor *msm_getstatus* *msm_setcurpos*
msm_getpress *msm_getrelease* *msm_setareax*
msm_setareay *msm_setgraphcur* *msm_settextcur*
msm_readcounters *msm_signal* *msm_lightpenon*
msm_setthreshold

2.1.20 page.h

此文件包含用于页存贮分配函数包的宏定义和函数说明.

宏定义如下:

`page_size` `page_toptr`

函数定义如下:

`page_calloc` `page_malloc` `page_realloc`
`page_free` `page_maxfree` `page_initialize`

2.1.21 process.h

此文件说明过程控制函数. 函数说明如下:

`_exit` `spawnl` `spawnr`
`spawnlp` `spawnvp` `execi`
`execv` `execvp` `execvp`

2.1.22 setjmp.h #ANSI

此文件定义一个类型 `jmp_buf`, 它是用于函数 `setjmp` 和 `longjmp` 的缓冲区. 它还包含这两个函数的说明.

2.1.23 signal.h #ANSI

此文件说明信号处理函数, 和与其相关的显型常量, 定义 `sigatomic_t` 类型, 函数说明如下:

`signal` `raise`

2.1.24 sound.h

定义关于声音函数包的函数说明:

`sound_tone` `sound_beep` `sound_click`

2.1.25 stdarg.h

此文件定义类型 `va_list` 和一些宏. 这些宏使用户可以在函数中使用带可变长度的参数. 如 `vprintf`. 宏定义如下:

`va_start` `va_arg` `va_end`

2.1.26 stddef.h #ANSI

此文件包含一个宏定义, 和那些通常使用的类型. 对全局变量 `errno` 有说明.

(`errno` 还在 `errno.h` 中有说明)

类型定义如下:

`ptrdiff_t` `size_t` `wchar_t`

宏定义是:

`offsetof`

2.1.27 stdio.h #ANSI

此文件包含常量定义, 宏定义和类型定义及缓冲区I/O函数说明. 一些显型常量及FILE结构类型也被定义.

其它类型定义如下:

`size_t fpos_t;`

宏定义如下:

`getchar putchar getc
putc ferror feof
clearerr fileno`

函数说明如下:

`tmpnam fopen freopen
fseek ftell fgetc
fgetc fflush fclose
fputs gets fputc
puts ungetc rewind
fread fwrite printf
fprintf vfprintf vprintf
sprintf vsprintf fscanf
fscanf sscanf setbuf
setvbuf remove rename
perror fcloseall filesize
flushall getch getche`

`unlink`

2.1.28 stdlib.h

此文件包含一些类型定义, 及常用的库函数说明:

类型定义如下:

`size_t wchar_t
div_t ldiv_t`

函数说明如下:

`atof atoi atol
strtod strtol strtoul
rand srand calloc
free malloc realloc
abort atexit exit
getenv system bsaarch
qsort abs div
labs ldiv itoa
ecvt fcvt _rotl
_rotr _lrotl _lrotr`

2.1.29 string.h #ANSI

此文件包含字符串和内存操作函数的说明:

memcpy memmove strcpy
strncpy strcat strncat
memcmp strcmp strcoll
strncmp strxfrm memchr
strchr strcspn strpbrk
strchr strspn strstr
strtok memset strerror
strlen memcmp strcmpi
strdup strlwrstrupr
strset strrev strset
swab movedata

2.1.30 sys\locking.h

此文件包含用在*locking*函数中的方式定义, *locking*函数要求MSDOS3.0或更高版本.

2.1.31 sys\stat.h

此文件定义一个结构类型, 说明两个对确定文件状态信息有关的函数.

结构类型如下:

struct stat

函数说明为: *stat fstat*

2.1.32 time.h #ANSI

此文件定义两个宏, 结构*tm*, 和一些用于时间函数包的其它类型, 说明所有于时间有关的函数.

类型定义是: *sig_t clock_t time_t*

宏定义是: *diffime gmtime*

函数说明如下:

clock time mktime
asctime ctime localtime
strftime sleep usleep
msleep

2.1.33 tsr.h

此文件包含关于(TSR)驻留程序包的函数说明:

tsr_install tsr_uninstall tsr_service

2.1.34 varargs.h

此文件包含关于unix c格式的 参数变量的宏, 及类型定义 `va_list`.

宏定义为:

```
va_dcl      va_sta      va_arg  
va_end
```

2.2 全局变量:

有几个全局变量, 定义程序中某部分的状态, 变量名和使用方法描述如下:

2.2.1 _8087

每一个程序开始执行, 先确定8087协处理器是否存在. 如果存在, 则所有浮点运算都用它处理, 如果不存在, 则所有浮点运算通过软件仿真来处理. 通过测试定义在C.ASM中的 `_8087` 全局变量, 能够确定协处理器是否存在.

可能值是:

- | | |
|------------|-------------|
| 0 没有协处理器在场 | 1. 8087在场 |
| 2. 80287在场 | 3. 80387在场. |

这对于那些对协处理器之间区别有影响的程序, 或用80387有更高效率的程序是非常有用的. Zortech库还体现在能够被这些标记控制.

2.2.2 _okbigbuf(仅适用于微,小模式)

这个变量定义使用哪一种内存分配方式. 有两种方式:

1. 在程序开始时, 分配所有的64可用内存到堆中. 这是一个缺省方式.
2. 仅仅分配所需内存到堆中, 这种方式当需要spawn函数时使用. 使用方法见 `_okbigbuf` 说明. 更详细的内容请参阅用户手册中 *Zortech Implementation* 一章.

用法:

```
int _okbigbuf=0;          /*分配所需内存*/  
_okbigbuf!=0 意为大磁盘缓冲区用于数据段外的流I/O, 这都由c.asm控制.
```

<注> 堆式分配: 即把内存分成数个长度可变内存, 需要存入数据时从其中选择出一个足够大的自由区, 把数据存入, 当该数据不需要时, 就释放这部分内存.

2.2.3 _osmajor

这个变量定义正在执行的MSDOS的版本号左部分. MSDOS3.21运行时, `_osmajor` 的值是3; MSDOS2.11运行时, 值为2.