

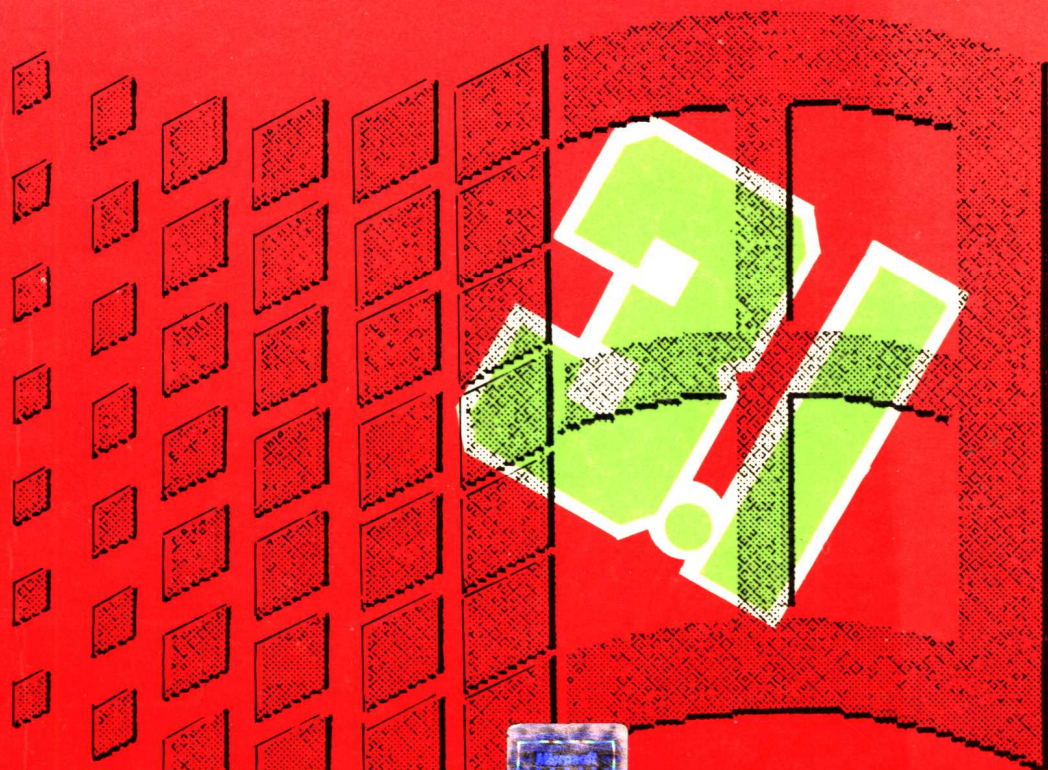
Microsoft Microsoft Microsoft

Microsoft Windows 3.1

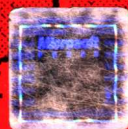
程序员参考大全(四)

—资源

赵先铭 曾明 邵毓琴 王旭 译



清华大学出版社



Microsoft

Windows 3.1

程序员参考大全(四) —资源

微软公司

[美] Microsoft 公司

赵先铭 曾 明 邵毓琴 王 旭 译

清华大学出版社

Microsoft Windows 3.1 Programmer's Reference

Volume 4 Resources

本书英文版由 Microsoft Press 出版。版权为 Microsoft 公司所有。

© Microsoft Corporation 1992.

本书中文版由 Microsoft Press 授权清华大学出版社独家出版,1993。未经出版者书面允许,不得用任何手段复制或抄袭本书部分或全部内容。

Adobe®和 PostScript®是 Adobe 系统公司的注册商标。

TrueType®是 Apple 计算机公司的注册商标。

Epson®是 Epson 美国公司的注册商标。

Hewlett-Packard®, HP®和 LaserJet®是 Hewlett-Packard 公司的注册商标。

ITC Zapf Chancery®和 ITC Zapf Dingbats®是国际 Typeface 公司的注册商标。

Code View®, Microsoft®, MS®, MS-DOS®是 Microsoft 公司的注册商标。

Windows™是 Microsoft 公司的商标。

Arial®和 Times New Roman®是 Monotype 公司 PLC 的注册商标。

(京)新登字158号

Microsoft Windows 3.1

程序员参考大全(四)——资源

赵先铭 曾 明 邵毓琴 王 旭 译

☆

清华大学出版社出版

北京 清华园

清华大学印刷厂印刷

新华书店总店科技发行所发行

☆

开本:787×1092 1/16 印张:14.75 字数:346千字

1993年5月第1版 1993年5月第1次印刷

印数:0001—5000

ISBN 7-302-01224-5/TP·462

定价:25.00元

引 言

Microsoft Windows 操作系统是一种采用图形化用户界面的单用户个人电脑操作系统。图形化界面带有许多形式各一的资源。本手册,《Microsoft Windows 3.1程序员参考大全(四)》,介绍这些资源的格式及可执行文件首部格式。

第一部分“文件格式”描述 Windows 应用程序使用的主要文件的格式。这部分中的各章详细描述了各种文件格式,以及 MS-DOS 和 Windows 可执行文件首部格式和可执行文件中的资源格式。具体包括下列文件格式:图形,裁剪板,字体,组,日历,对象模块,库,符号和元文件。

第二部分“工具参考”详细介绍用于创建和维护 Windows 资源的工具的语句、命令、和宏。其内容包括资源定义语句,汇编语言宏,及帮助(help)语句和宏。其中每项都列出:命令或宏的目的;完整的语法和参数及返回值;相关命令或宏的交叉参考。许多项还提供了关于命令或宏的用法的扩充说明。

如何使用这个手册

本手册描述 Windows 资源文件的格式。每章介绍与某个特定资源或用途有关的文件的格式。例如,关于图形文件格式的章介绍用于位图、图符和光标资源文件的格式。

每章有两部分:文件类型的一般描述和格式的详表示。第二部分“工具参考”中的章节只描述文件格式,并不介绍工具。有关工具的进一步介绍参见《Microsoft Windows 3.1程序员参考大全(五)——编程工具》。

文本规则

本手册用下列规则定义语法:

规 则	意 义
正体字	指示逐字输入一个词或一个字符,如资源定义语句或函数名(MENU 或 Create Window),MS-DOS 命令或命令行选择项(/nod)。编程时须严格按所示内容编写。
斜体字	指示一个占位符或变量;编程时提供实际值。例如,语句 Set Cursor Pos(X, Y)要求用值替代 X 和 Y 参数。
[]	括起选择项。
	分隔“二选一”或者“或”选项。
...	指示重复前面内容。

(续表)

规 则	意 义
BEGIN	
⋮	表示范例程序中的省略内容。
END	

此外,为帮助学习本书,书中采用下列文字规则:

规 则	意 义
“小号”大写字符	表示键名,键序列或键组合——例如,ALT+SPACEBAR。
“大号”大写字符	表示文件名和路径,类型和结构名(也可能是黑体字),常量。
单空格	分隔代码及语法。

目 录

引言	■	5.2.2 项目数据	(43)
第一部分 文件格式		5.2.3 标签数据	(44)
第1章 图形文件格式	(3)	第6章 可执行文件首部格式	(45)
1.1 位图文件格式	(3)	6.1 MS-DOS 首部	(45)
1.1.1 位图文件结构	(3)	6.2 Windows 首部	(45)
1.1.2 位图的压缩	(4)	6.2.1 信息块	(46)
1.1.3 位图示例	(6)	6.2.2 段表	(48)
1.2 图符资源文件格式	(7)	6.2.3 资源表	(49)
1.2.1 图符目录	(7)	6.2.4 驻留名表	(51)
1.2.2 图符图象	(8)	6.2.5 模块引用表	(51)
1.2.3 Windows 图符的选择	(9)	6.2.6 输入名表	(51)
1.3 光标资源文件格式	(9)	6.2.7 入口表	(51)
1.3.1 光标目录	(10)	6.2.8 非驻留名表	(52)
1.3.2 光标图象	(11)	6.3 代码段和重定位数据	(53)
1.3.3 Windows 光标的选择	(12)	第7章 可执行文件内的资源格式	(54)
第2章 裁剪板文件格式	(13)	7.1 图符资源	(54)
2.1 裁剪板文件首部	(13)	7.2 图符目录资源	(54)
2.2 裁剪板文件结构	(13)	7.3 光标资源	(55)
第3章 元文件格式	(14)	7.4 光标目录资源	(55)
3.1 元文件首部	(14)	7.5 菜单资源	(56)
3.2 典型元文件记录	(15)	7.5.1 菜单首部	(56)
3.3 可定位 Windows 元文件	(17)	7.5.2 弹出式菜单项	(56)
3.4 关于 Windows 元文件的若干准则	(18)	7.5.3 标准菜单项	(57)
3.5 元文件程序输出示例	(18)	7.5.4 组合菜单项	(57)
3.6 特定函数元文件记录	(19)	7.6 对话框资源	(58)
第4章 字体文件格式	(33)	7.6.1 对话框首部	(58)
4.1 字体文件的组织	(33)	7.6.2 控制数据	(60)
4.2 字体文件结构	(33)	7.7 位图资源	(61)
4.3 版本专用的字符图形表	(39)	7.8 字体资源	(61)
第5章 组文件格式	(41)	7.8.1 字体目录数据	(61)
5.1 组文件的组织	(41)	7.8.2 字体组分数据	(62)
5.2 组文件结构	(41)	7.9 字符串表资源	(63)
5.2.1 组文件首部	(41)	7.10 加速键资源	(63)
		7.11 名表资源	(64)
		7.11.1 名表项	(64)

7.12 版本信息资源	(65)	12.5.2 LINEINF 结构	(97)
7.12.1 根块	(66)	第二部分 工具参考	
7.12.2 变量信息块	(66)	第13章 资源描述语句	(101)
7.12.3 字符串信息块	(68)	13.1 资源语句目录	(101)
7.12.4 语言专用块	(68)	第14章 汇编语言宏指令	(151)
第8章 字处理器(Write)文件格式		14.1 创建汇编语言 Windows 应用程序	
.....	(70)		(151)
8.1 Write 文件首部	(70)	14.1.1 指定内存模式	(151)
8.2 正文与插图	(71)	14.1.2 选择调用约定	(152)
8.2.1 正文	(71)	14.1.3 使 Windows 的前缀及	
8.2.2 插图	(71)	后缀选项有效	(153)
8.3 格式化	(72)	14.1.4 包含 CMACROS.INC	
8.3.1 字符与段落	(72)	文件	(153)
8.3.2 脚注	(75)	14.1.5 创建应用程序入口点	
8.3.3 段	(75)		(153)
8.3.4 字体表	(76)	14.1.6 申明回调函数	(153)
第9章 日历文件格式	(78)	14.1.7 库连接	(154)
9.1 Calendar 文件首部	(78)	14.1.8 使栈检查有效	(154)
9.2 日期描述符	(79)	14.2 Cmacros 宏指令组	(154)
9.3 日志专用信息	(79)	14.2.1 段宏指令	(155)
9.4 约会安排专用信息	(80)	14.2.2 存储分配宏指令	(155)
第10章 Windows 目标模块格式	(82)	14.2.3 函数宏指令	(155)
10.1 目标模块格式记录	(82)	14.2.4 调用宏指令	(156)
10.2 记录参考	(83)	14.2.5 特殊定义宏指令	(156)
第11章 库和调入库格式	(88)	14.2.6 错误处理宏指令	(156)
11.1 库的组织	(88)	14.3 Cmacros 的使用	(156)
11.2 字典	(88)	14.3.1 超越类型	(156)
11.2.1 冲突的解决	(89)	14.3.2 符号重定义	(157)
11.3 记录参考	(89)	14.3.3 Cmacros 函数示例	(157)
第12章 符号文件格式	(92)	14.4 汇编语言宏指令目录	(158)
12.1 映象定义	(92)	第15章 Windows 的 Help 语句	
12.2 段定义	(93)	和宏指令	(172)
12.3 符号定义	(95)	15.1 Help 语句的语法	(172)
12.4 常量定义	(96)	15.2 Help 宏指令的语法	(173)
12.5 行定义	(96)	15.3 Help 语句目录	(173)
12.5.1 LINEDEF 结构	(96)	15.4 Help 宏指令目录	(208)

第一部分

文 件 格 式

第 1 章

图形文件格式

本章介绍 Microsoft Windows 操作系统使用的图形文件格式。图形文件包括位图文件、图符资源文件和光标资源文件。

1.1 位图文件格式

Windows 的位图文件以设备无关位图(DIB)格式存储,使得 Windows 可以在任何设备上显示这个位图。“设备无关”一词在这里指的是位图指定象素颜色的方式与任何显示设备表示颜色所用的方式无关。Windows 设备无关位图(DIB)文件的缺省扩展名为 .BMP。

1.1.1 位图文件结构

每个位图文件包括一个位图文件首部、一个位图信息首部、一个颜色表和一个定义位图各位的字节数组。位图文件具有如下形式:

```
BITMAPFILEHEADER  bmfh;  
BITMAPINFOHEADER  bmih;  
RGBQUAD           aColors [ ];  
BYTE               aBitmapBits [ ];
```

位图文件首部内容包括该设备无关位图文件的类型、大小和布局信息。位图文件首部定义成 BITMAPFILEHEADER 结构。

位图信息首部定义成 BITMAPINFOHEADER 结构,用来确定位图的尺寸、压缩类型和颜色格式信息。

颜色表定义成 RGBQUAD 结构类型的数组,其元素个数与位图中的颜色数目相同。由于 24 位彩色位图在位图数据域中每个象素均由 24 位红-绿-蓝(RGB)值来表示,因此这种位图无颜色表。表中的颜色必须按其重要性依次排列,当显示设备能够显示的颜色数目少于位图文件中包含的颜色数目时,颜色表的排列方式可以帮助设备驱动器把位图重现在该设备上。如果设备无关位图采用 Windows 3.0 版本及更高版本的格式,驱动器可以用 BITMAPINFOHEADER 结构中的 biClrImportant 成分来确定哪些颜色是重要的。

BITMAPINFO 结构可以用来表示位图信息首部和颜色表的合成。

位图中的各个位由一个字节数组组成,紧跟在颜色表之后,这个字节数组的值用来表示位图中相继的一行行象素(或称为扫描线)。从左到右排列的若干连续字节表示扫描线上的象素,组成一条条扫描线。表示一条扫描线所用到的字节数目由位图的颜色格式和以象素为单位的宽度决定。如果需要的话,一条扫描线必须用零填充直到行结束的 32 位边

界处。但是,段边界可以出现在位图中任何地方。位图中的扫描线是从下到上存储的。这意味着数组中的第一个字节表示位图的左下角,最后一个字节表示位图的右上角。

BITMAPINFOHEADER 结构中的 biBitCount 成分确定了位图中定义每个像素所需的位数和可用的最大颜色数目。这个成分可以取下列值之一:

值	意 义
1	位图是单色的,则颜色表中包含两项。位图数组中的每一位代表一个像素。如果某一位被置为 0,则相应的像素以颜色表中第一项的颜色显示。如果该位被设置为 1,则相应的像素具有颜色表中第二项的颜色。
4	位图最多有 16 种颜色。位图中的每个像素由指向颜色表的 4 位索引来表示。例如,若位图中的第一个字节是 0x1F,则该字节表示两个像素。第一个像素具有位于第 2 表项中的颜色,第二个像素具有第 16 表项中的颜色。
8	位图最多有 256 种颜色。位图中每个像素由指向颜色表的 1 字节索引来表示。例如,如果位图中的第一字节是 0x1F,则第一个像素具有第 32 表项中的颜色。
24	位图最多有 2^{24} 种颜色。bmi Colors(或 bmci Colors)成分是 NULL,并且位图数组中每 3 个字节分别表示一个像素的红、绿、蓝三种颜色的相对强度。

BITMAPINFOHEADER 结构中的 biClrUsed 成分指定了颜色表中由位图实际使用的颜色索引数目。如果 biClrUsed 成分设为 0,则位图使用与 biBitCount 成分值一致的最大颜色数目。

位图还可选另一种方式来表示,即采用 BITMAPCOREINFO, BITMAPCOREHEADER 和 RGBTRIPLE 结构。

有关位图结构的完整描述,参见《Microsoft Windows 3.1 程序员参考大全(三)》。

1.1.2 位图的压缩

Windows 3.0 版本及更高版本支持行程编码格式来压缩位图,这些位图的像素是用 4 个比特或 8 个比特来表示的。压缩可以减少位图存储所需的磁盘和内存空间。

1.1.2.1 8 位位图的压缩

当 BITMAPINFOHEADER 结构中的 biCompression 成分设置为 BI_RLE8 时,用 256 色位图行程编码格式将该设备无关位图进行压缩。这种格式使用两种方式:编码方式和绝对方式。两种方式都可以出现在位图的任何地方。

编码方式 在编码方式中一个信息单位由两个字节组成。第一个字节给出要画出的相连像素的个数,这些像素所用的颜色索引由第二个字节给出。这对字节中第一个字节也可以设置为 0,以表示一个控制符、标志线的端点、位图的结束或一个增量值。该控制符的解释与第二个字节的值有关,该值必须在 0x00 至 0x02 之间。下面给出第二字节中可以使用的控制符值的意义:

第二字节值	意 义
0	线结束。
1	位图结束。

(续表)

第二字节值	意 义
2	增量。紧跟控制符之后的两个字节中所包含的无符号值指定了下个像素相对于当前位置的水平和垂直偏移量。

绝对方式 绝对方式的标志是第一个字节为 0, 第二个字节在 0x03 至 0xFF 之间设置值的一对字节。第二字节的值表示跟随其后的字节数目, 其中每个字节都含有一个像素的颜色索引。每个行程编码都必须补齐到字的边界上。

下面是一个 8 位行程编码压缩位图的示例(其中第二列项的两个 16 进位数值表示一个像素的颜色索引):

压 缩 数 据	展 开 数 据
03 04	04 04 04
05 06	06 06 06 06 06
00 03 45 56 67 00	45 56 67
02 78	78 78
00 02 05 01	从当前位置右移 5 个像素, 下移 1 个像素。
02 78	78 78
00 00	线结束。
09 1E	1E 1E 1E 1E 1E 1E 1E 1E 1E
00 01	RLE 位图结束。

1.1.2.2 4 位位图的压缩

当 BITMAPINFOHEADER 结构中的 biCompression 成分被设置为 BI_RLE4 时, 用 16 色位图的行程编码格式对设备无关位图进行压缩。这种格式也使用两种方式: 编码方式和绝对方式。

编码方式 在编码方式中, 一个信息单位由两个字节组成。第一个字节给出要画出的像素的个数, 这些像素所用的颜色索引由第二个字节给出。

第二个字节包含两个颜色索引, 一个在上半部分(即字节的高 4 位), 另一个在下半部分。第一个像素用上半部分指定的颜色索引画出, 第二个像素用下半部分指定的颜色索引画出, 第三个像素再用上半部分指定的颜色索引画出, 如此类推, 直到由第一个字节指定的所有像素全部画出为止。

第一个字节可以设置为 0 以表示一个控制符、标志线的端点、位图的结束或一个增量值。控制符的解释与第二个字节的值有关。在编码方式中, 第二个字节必须取 0x00 至 0x02 之间的值。这些值的意义与 8 位设备无关位图中的意义相同。

绝对方式 在绝对方式中, 第一个字节为 0, 第二个字节的值表示紧跟其后的颜色索引的个数。随后的字节分别在其上半部分和下半部分中包含颜色索引, 一个颜色索引对应一个像素。每个行程编码必须补齐到字边界上。

下面是一个 4 位行程编码压缩位图的示例(其中第二列项中的一个 16 进位数值代表一个像素的颜色索引):

压 缩 数 据	展 开 数 据
03 04	040
05 06	06060
00 06 45 56 67 00	455667
04 78	7878
00 02 05 01	从当前位置左移 5 个像素,下移 1 个像素。
04 78	7878
00 00	线结束。
09 1E	1E1E1E1E1
00 01	RLE 位图结束。

1.1.3 位图示例

下面给出的例子是一个 16 色位图(每个像素 4 位)的正文打印内容:

Win3DIBFile

BitmapFileHeader

Type	19778
Size	3118
Reserved1	0
Reserved2	0
OffsetBits	118

BitmapInfoHeader

Size	40
Width	80
Height	75
Planes	1
BitCount	4
Compression	0
SizeImage	3000
XPelsPerMeter	0
YPelsPerMeter	0
ColorsUsed	16
ColorsImportant	16

Win3ColorTable

	Blue	Green	Red	Unused
[00000000]	84	252	84	0
[00000001]	252	252	84	0
[00000002]	84	84	252	0
[00000003]	252	84	252	0
[00000004]	84	252	252	0
[00000005]	252	252	252	0
[00000006]	0	0	0	0
[00000007]	168	0	0	0
[00000008]	0	168	0	0
[00000009]	168	168	0	0
[0000000A]	0	0	168	0

[0000000B]	168	0	168	0
[0000000C]	0	168	168	0
[0000000D]	168	168	168	0
[0000000E]	84	84	84	0
[0000000F]	252	84	84	0

Image

•
•
•

Bitmap data

1.2 图符资源文件格式

图符资源文件包含了 Windows 应用程序所用图符的图象数据。文件由一个标志文件中图符图象数目和类型的图符目录及一个或多个图符图象组成。图符资源文件的缺省扩展名为 .ICO。

1.2.1 图符目录

每个图符资源文件都从一个图符目录开始。图符目录定义成一个 ICONDIR 类型的结构,指定了资源中图符的数目以及每个图符图象的尺寸和颜色格式。ICONDIR 结构具有如下形式:

```
typedef struct ICONDIR {
    WORD        idReserved;
    WORD        idType;
    WORD        idCount;
    ICONDIRENTRY idEntries [1];
} ICONHEADER;
```

下面说明 ICONPIR 结构中的各个成分:

idReserved

留用;必须设置为 0。

idType

指定资源文件的类型。这个成分设置为 1。

idCount

指定目录中的项数。

idEntries

指定一个 ICONDIRENTRY 结构数组,其中包含有关每个单独图符的信息。idCount 成分指定了数组的大小。

ICONDIRENTRY 结构指定了一个图符的尺寸和颜色格式。这个结构具有如下形式:

```
struct IconDirectoryEntry {
    BYTE    bWidth;
    BYTE    bHeight;
```

```

        BYTE      bColorCount;
        BYTE      bReserved;
        WORD       wPlanes;
        WORD       wBitCount;
        DWORD      dwBytesInRes;
        DWORD      dwImageOffset;
    };

```

下面说明 ICONDIRENTRY 结构中的各个成分：

bWidth

以像素为单位指定图符的宽度。允许值为 16、32 和 64。

bHeight

以像素为单位指定图符的高度。允许值为 16、32 和 64。

bColorCount

指定图符中的颜色数目。允许值为 2、8 和 16。

bReserved

留用；必须设置为 0。

wPlanes

指定图符位图中的颜色面数。

wBitCount

指定图符位图中的字节数。

dwBytesInRes

以字节为单位指定资源文件的大小。

dwImageOffset

以字节为单位指定从文件起始到图符图象的偏移量。

1.2.2 图符图象

每个图符资源文件包含与图符目录中标识的每个图象相对应的图符图象。图符图象由图符图象首部、颜色表、XOR 掩码和 AND 掩码组成。图符图象具有如下形式：

```

    BITMAPINFOHEADER  icHeader;
    RGBQUAD            icColors [ ];
    BYTE               icXOR [ ];
    BYTE               icAND [ ];

```

图符图象首部定义成 BITMAPINFOHEADER 结构，指定了图符位图的尺寸和颜色格式。这里只用到了 biSize 到 biBitCount 之间的成分和 biSizeImage 成分。其它成分（如 biCompression 和 biClrImportant 等）必须设置为 0。

颜色表定义成 RGBQUAD 结构数组，指定了 XOR 掩码用到的颜色。与位图文件中的颜色表一样，图符图象首部的 biBitCount 成分确定了颜色数组元素的数目。有关颜色表的进一步信息，参见 1.1 节，“位图文件格式”。

有关图符资源结构的完整说明，参见《Microsoft Windows 3.1 程序员参考大全

(三)》。

XOR 掩码是一个 BYTE(字节)数组,紧跟在颜色表之后。数组值表示位图中相继的行。这个位图定义了图符图象的基本形状和颜色。如同位图文件中的位图位,图符资源文件中的位图数据也是以扫描线方式组织的,根据颜色格式的定义,每个字节表示一个或多个象素。有关位图位的进一步信息,参见 1.1 节“位图文件格式”。

XOR 掩码之后是 AND 掩码,也是一个字节数组,其值表示一个单色的位图,该位图的宽和高与 XOR 掩码一致。这个数组也是以扫描线方式组织的,其中每个字节表示 8 个象素。

当 Windows 系统绘制一个图符时,系统用 AND 掩码和 XOR 掩码把图符图象与已存在于显示面上的象素组合起来。Windows 首先通过位 AND 操作来实现 AND 掩码,保留或删除已有的象素颜色。然后通过位 XOR 操作实现 XOR 掩码。这样就形成了每个象素的最终颜色。

下图显示了创建一个具有大写字母 K 形式的单色图符的 XOR 和 AND 掩码(图符的尺寸为 8×8 象素):

AND 掩码								XOR 掩码								产生的图符								
0	0	1	1	1	0	0	1	1	1	0	0	0	1	1	0	K	K				K	K		
0	0	1	1	0	0	1	1	1	1	0	0	1	1	0	0	K	K			K	K			
0	0	1	0	0	1	1	1	1	1	0	1	1	0	0	0	K	K			K	K			
0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	K	K	K	K					
0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	K	K	K	K					
0	0	1	0	0	1	1	1	1	1	1	0	1	1	0	0	K	K			K	K			
0	0	1	1	0	0	1	1	1	1	1	0	0	1	1	0	K	K			K	K			
0	0	1	1	1	0	0	1	1	1	1	0	0	0	1	0	K	K				K	K		

1.2.3 Windows 图符的选择

Windows 检测当前显示设备的分辨率并且根据每个图符图象的宽度和高度与设备进行匹配。如果 Windows 确定存在一个位图图象与当前设备严格匹配,则系统采用其匹配图象。否则选择最接近的匹配值并且把图象调整到合适的尺寸。

如果图符资源文件中含有多个图象与特定的分辨率相匹配,那么 Windows 选择其中与显示器显示颜色的能力最匹配的图符图象。如果没有一个图象与设备显示颜色的能力相匹配,那么 Windows 将选择具有颜色数最多且不超出显示器显示颜色能力的图象。如果所有图象的颜色数都超出了当前显示设备显示颜色的能力,Windows 将采用包含颜色数目最少的图符图象。

1.3 光标资源文件格式

光标资源文件包含了 Windows 应用程序所用光标的图象数据。文件由一个标志资源

中光标图象数目和类型的光标目录及一个或多个光标图象组成。光标资源文件的缺省扩展名为.CUR。

1.3.1 光标目录

每个光标资源文件都以一个光标目录开始。光标目录定义为 CURSORDIR 结构,指定了文件中含有光标的数目及光标图象的尺寸和颜色格式。CURSORDIR 结构具有如下形式:

```
typedef struct _CURSORDIR {  
    WORD        cdReserved;  
    WORD        cdType;  
    WORD        cdCount;  
    CURSORDIRENTRY cdEntries [ ];  
} CURSORDIR;
```

下面说明 CURSORDIR 结构中的各个成分:

cdReserved

留用;必须设置为 0。

cdType

指定资源文件类型。这个成分必须设置为 2。

cdCount

指定文件中光标的数目。

cdEntries

指定一个 CURSORDIRENTRY 结构数组,其中包含有关每个单独光标的信息。cdCount 成分指定了该数组的大小。

一个 CURSORDIRENTRY 结构指定了光标图象的尺寸和颜色格式。这个结构具有如下形式:

```
typedef struct _CURSORDIRENTRY {  
    BYTE        bWidth;  
    BYTE        bHeight;  
    BYTE        bColorCount;  
    BYTE        bReserved;  
    WORD        wXHotspot;  
    WORD        wYHotspot;  
    DWORD        lBytesInRes;  
    DWORD        dwImagOffset;  
} CURSORDIRENTRY;
```

下面说明 CURSORDIRENTRY 结构中的各个成分:

bWidth

以像素为单位指定光标的宽度。

bHeight

以像素为单位指定光标的高度。