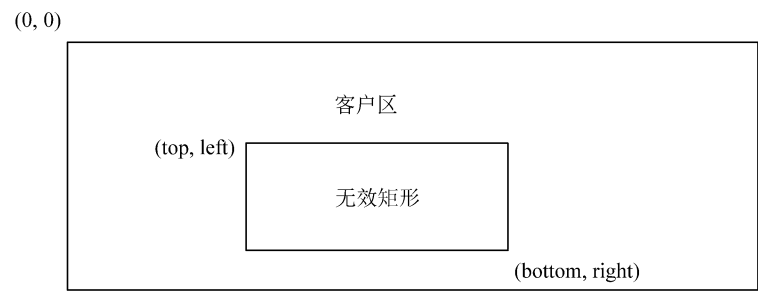


```
    BOOL    fIncUpdate ;           //操作系统用
    BYTE    rgbReserved[32] ;     //操作系统用
} PAINTSTRUCT ;
```

通过“PAINTSTRUCT ps ;”定义变量,程序调用 BeginPaint 时,返回值是 hDC,Windows 会把它填入 PAINTSTRUCT 中。用户程序只使用画图信息结构的前三个条款,其他条款由 Windows 操作系统内部使用。hdc 条款是 BeginPaint 的返回值。在大多数情况下,fErase 被标志为 FALSE(0),这意味着 Windows 已经擦除了无效矩形的背景。在早期 Windows 版本中,如果要在窗口函数中自己定义一些显示背景擦除行为,可以自行处理 WM_ERASEBKGD 消息。但是,现在 Windows 使用 WNDCLASS 结构的 hbrBackground 条款指定的画刷来擦除背景,这个 WNDCLASS 结构是程序在 WinMain 初始化期间登录窗口类时使用的。许多 Windows 程序使用白色画刷。以下就是窗口类背景条款的设定方法:

```
wndclass.hbrBackground = (HBRUSH) GetStockObject(WHITE_BRUSH) ;
```

不过,如果程序通过调用 InvalidateRect 函数使客户区中的矩形失效,则该函数的最后一个参数会指示是否擦除背景。如果这个参数为 FALSE(即 0),则执行 InvalidateRect 时 Windows 将不擦除背景,反之,则会擦除背景。在调用 BeginPaint 后 PAINTSTRUCT 结构的 fErase 条款将成为 TRUE(非零),PAINTSTRUCT 结构的 rcPaint 条款是 RECT 类型的结构,RECT 结构定义了一个矩形,其四个条款为 left、top、right 和 bottom,分别表示 PAINTSTRUCT 结构中 rcPaint 条款定义无效矩形的边界坐标,如图 6-1 所示。这些值均以像素为单位,并相对于客户区的左上角。无效矩形就是应该重画的区域。



(0, 0)是客户区坐标原点;
(top, left)是无效矩形左上角, (bottom, right)是无效矩形右下角

图 6-1 无效矩形的边界

PAINTSTRUCT 中的 rcPaint 矩形不仅是无效矩形,它还是一个“剪取”矩形。这意味着 Windows 将绘图操作限制在剪取矩形内(更确切地说,如果无效矩形区域不为矩形,则 Windows 将绘图操作限制在这个区域内)。

在处理 WM_PAINT 消息时,为了在更新的矩形外绘图,可以使用如下调用: