

Petzold, 1999; Simon, 1997; Rector 和Newcomer, 1997)。显然,在这一小节中,我们只可能浅尝其表面并且介绍少许基本的概念。为了使讨论具体化,我们将描述Win32 API,它被Windows的所有32位版本所支持。在一般意义上,其他GUI的输出软件大体上是相似的,但是细节迥然不同。

屏幕上的基本项目是一个矩形区域，称为窗口（window）。窗口的位置和大小通过给定两个斜对角的坐标（以像素为单位）惟一地决定。窗口可以包含一个标题条、一个菜单条、一个工具条、一个垂直滚动条和一个水平滚动条。典型的窗口如图5-39所示。注意，Windows的坐标系将原点置于左上角并且y向下增长，这不同于数学中使用的笛卡儿坐标。

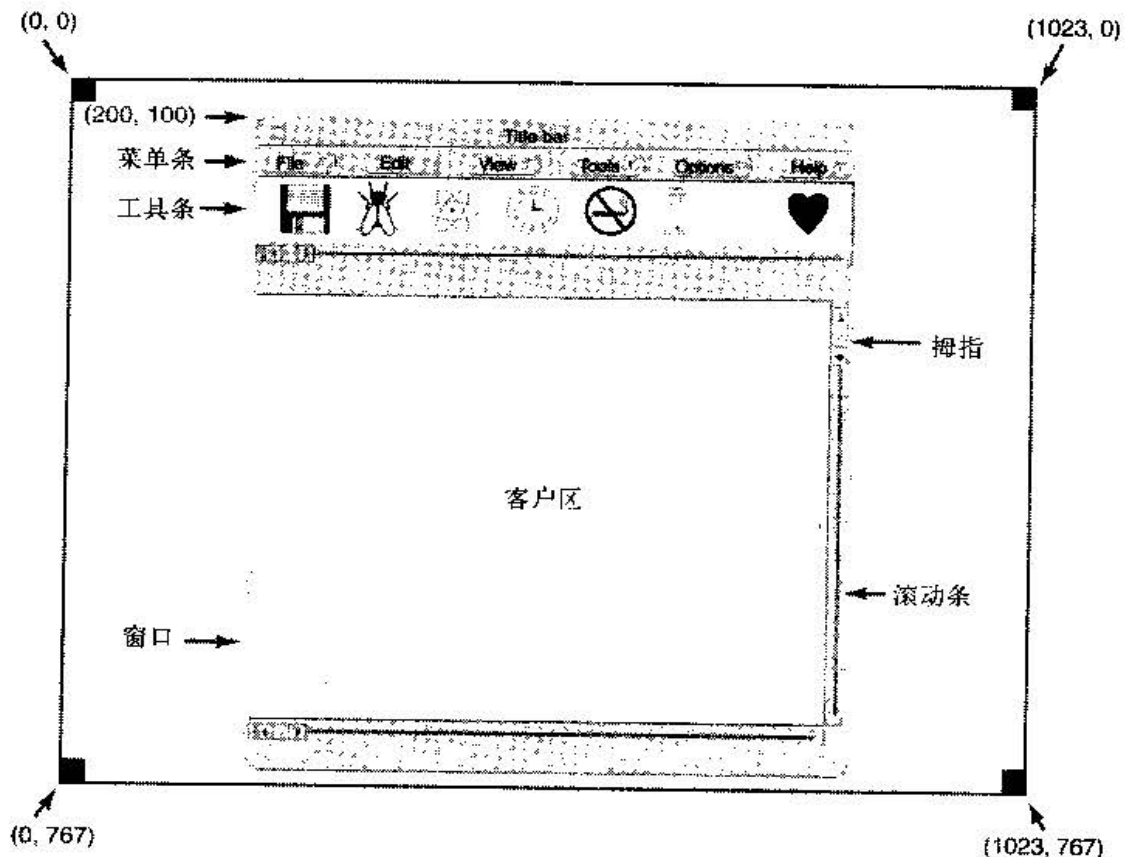


图5-39 XGA显示器上位于(200, 100)处的一个窗口样例

当窗口被创建时，有一些参数可以设定窗口是否可以被用户移动，是否可以被用户调整大小，或者是否可以被用户滚动（通过拖动滚动条上的拇指）。大多数程序产生的主窗口可以被移动、调整大小和滚动，这对于Windows程序的编写方式具有重大的意义。特别地，程序必须被告知关于其窗口大小的改变，并且必须准备在任何时刻重画其窗口的内容，即使在程序最不期望的时候。

因此，Windows程序是面向消息的。涉及键盘和鼠标的用户操作被Windows所捕获，并且转换成消息，送到正在被访问的窗口所属于的程序。每个程序都有一个消息队列，与程序的所有窗口相关的消息都被发送到该队列中。程序的主循环包括提取下一条消息，并且通过调用针对该消息类型的内部过程对其进行处理。在某些情况下，Windows本身可以绕过消息队列而直接调用这些过程。这一模型与UNIX的过程化代码模型完全不同，UNIX模型是提请系统调用与操作系统相互作用的。然而，X是面向事件的。

为了使这一编程模型更加清晰，请考虑图5-40的例子。在这里我们看到的是Windows主程序的框架，它并不完整并且没有做错误检查，但是对于我们的意图而言它显示了足够的细节。程序的开头包含一个头文件windows.h，它包含许多宏、数据类型、常数、函数原型，以及Windows程序所需要的其他信息。

主程序以一个声明开始，该声明给出了它的名字和参数。WINAPI宏是一条给编译器的指令，让编译器使用一定的参数传递约定并且不需要我们进一步关心。第一个参数h是一个实例句柄，用来向系统的其他部分标识程序。在某种程度上，Win32是面向对象的，这意味着系统包含对象（例如程序、文件