绘图系统设计及实现

目 录

绘图系统设计及实现

摘　要

自从上世纪计算机系统向绘画界的延伸以来，绘图技术已成为计算机系统不可分割的一部分。而图形编辑器又是图形编辑软件的基础，几乎所有的图形编辑软件，都是在拥有基本图形编辑功能的基础上实现更复杂功能的。故在图形应用非常广泛的今天，研究开发绘图系统是非常有意义的。本文主要经过对绘图系统软硬件配置的分析，就绘图系统的设计及实现进行了合理的阐述。在基于 MFC 的操作环境中，通过对 CD、CDC 类以及 CDI、CGdiobject 类的应用，在绘图方面实现绘制直线、矩形、圆形等基本功能，在界面设计方面实现了对图形的操作，从而从底层对绘图系统进行简单的实现。

关键词：硬件系统组成　软件系统构架　图形操作及绘画技术　基本绘图功能　图形界面

Design and implementation of drawing system

Abstract

Since the last century, the computer system has been extended to the painting industry, computer graphics technology has become an integral part of the system. The graphics editor graphics editing software is the foundation of almost all of the graphics editing software, are in possession of basic graphics editing capabilities based on the more complex functions. It is widely used in graphics today, research and development of graphics systems is very meaningful. This article focuses on the mapping system through the analysis of hardware and software configurations, the mapping system design and implementation of a reasonable set. MFC-based operating environment, through the CD, CDC class, and CDI, CGdiobject class applications, to achieve in the drawing to draw lines, rectangles, circles and other basic functions, implemented in the interface design of graphics operations, and thus from the underlying graphics system for simple implementation.

Key words: Hardware system drawing technology　The software system structure draw tools　Graphic operation and GUI

一、绘图系统概述

计算机绘图系统是基于计算机的系统，由软件系统和硬件系统组成。其中，软件是计算机绘图系统的核心，而相应的系统硬件设备则为软件的正常运行提供了基础保障和运行环境。另外，任何功能强大的计算机绘图系统都只是一个辅助工具，系统的运行离不开系统使用人员的创造性思维活动。因此，使用计算机绘图系统的技术人员也属于系统组成的一部分，将软件、硬件及人这三者有效地融合在一起，是发挥计算机系统强大功能的前提。从上世纪计算机系统向绘画界得引入到现在，计算机绘图已然成为计算机图形学的一个重要分支，其主要特点是向计算机输入非图形信息，由计算机处理后生成图形输出。

二、硬件系统组成

通常，将用户进行计算机绘图作业的独立硬件环境称作计算机绘图的硬件系统。计算机绘图的硬件主要由主机、输入设备（键盘、鼠标、扫描仪等）、输出设备（显示器、绘图仪、打印机等）、信息存储设备（主要指外存，如硬盘、软盘、光盘等）、以及网络设备、多媒体设备等组成[1]。如图 1-1 所示

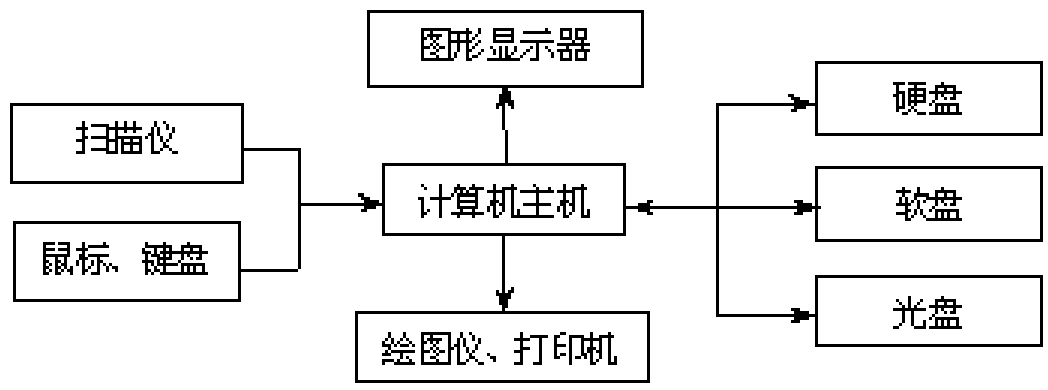

图 2.1 计算机绘图系统的基本硬件组成

（一）主机

主机由中央处理器（CPU）和内存储器（简称内存）组成，是整个计算机绘图系统的核心。衡量一个主机性能的指标中相互要有两项：CPU性能和内存容量。

1．CPU性能

CPU 的性能决定着计算机的数据处理能力、运算精度和速度。CPU 的性能通常用每秒可执行的指令数目或进行浮点运算的速度指标来衡量，其单位符号为 MI/S(每秒处理 1 百万条指令)和 GI/S(每秒处理 10 亿条指令)。目前，CPU 的速度已达到 160GI/S 以上。一般情况下，用芯片的时钟频率来表示运算速度更为普遍，时钟频率越高，运算速度越快。

2. 内存容量

内存是存放运算程序、原始数据、计算结果等内容的记忆装置。如果内存容量过小，将直接影响计算机绘图软件系统的运行效果。因为，内存容量越大，主机能容纳和处理的信息量也就越大。

（二）外存储器

外存储器简称为外存，虽然内存储器可以直接和运算器、控制器交换信息，存取速度很快，但内存储器成本较高，且其容量受到CPU直接寻址能力的限制。外存作为内存的后援，是计算机绘图系统将大量程序、数据库、图形库存放在外存器中，待需要时再调入内存进行处理。外存储器通常包括硬盘、软盘、光盘等。

（三）图形输入设备

在计算机绘图作业过程中，不仅要求用户能够快速输入图形，而且还要求能够将输入的图形以人机交互方式进行修改，以及对输入的图形进行图形变换(如缩放、平移、旋转)等操作。因此，图形输入设备在计算机绘图硬件系统中占有重要的地位。目前，计算机绘图系统常用的输入设备有键盘、鼠标、扫描仪等。

（四）图形输出设备

图形输出设备包括图形显示器、绘图仪、打印机等。图形显示器是计算机绘图系统中最为重要的硬件设备之一，主要用于图形图像的显示和人机交互操作，是一种交互式的图形显示设备，其主要部件是阴极射线管(CRT)。它有 3 种类型：直接存储管式显示器、射线刷新式显示器、光栅扫描式显示器。目前，交互式图形系统采用的主流显示器是基于 CRT 的光栅扫描式显示器。其工作原理与电视机相似，不同之处在于电视机利用摄像机产生的模拟信号构成屏幕上的图像，而光栅扫描式显示器则利用计算机产生的数字信号构成屏幕上的图像。衡量显示器性能的主要指标是分辨率和显示速度。对于光栅扫描式显示器而言，沿水平和垂直方向单位长度上所能识别的最大光点数称为分辨率(光点也称为像素)。对于相同尺寸的屏幕，点数越多，距离越小，分辨率就越高，显示的图形也越精细。显示速度同显示器在输出图形时采用的分辨率以及计算机本身处理图形的速度有关。从人机工程学的角度来看，通常应满足人眼观察图形时不出现闪烁这一基本要求，图形屏幕的刷新速度应不低于 30 帧/秒。随着人们对显示器轻型化、薄型化以及大尺寸的要求，目前，液晶显示器和等离子显示器的应用越来越多。由于这些显示器的制造成本逐渐降低，已呈现出取代基于 CRT 的光栅扫描式显示器的趋势。绘图仪、打印机等也是目前常用的图形输出设备。目前，常用的绘图仪为滚筒式绘图仪，这种绘图仪具有结构简单紧凑、图纸长度不受限制、价格便宜、占用工作面积小等优点。常用的打印机主要有针式、喷墨、激光打印机等。

三、软件系统概述

计算机软件是指控制计算机运行，并使计算机发挥最大功效的各种程序、数据及文档的集合。在计算机绘图系统中，软件配置水平决定着整个计算机绘图系统的性能优劣。因此可以说硬件是计算机绘图系统的物质基础，而软件则是计算机绘图系统的核心。从计算机绘图系统的发展趋势来看，软件占据着愈来愈重要的地位，目前，系统配置中的软件成本已经超过了硬件。目前而言，计算机绘图系统的软件可以分为 3 个层次，即系统软件、支撑软件和应用软件。系统软件是与计算机硬件直接关联的软件，一般由专业的软件开发人员研制，它起着扩充计算机的功能以及合理调度与使用计算机的作用。系统软件有 2 个特点：一是公用性，无论哪个应用领域都要用到它；二是基础性，各种支撑软件及应用软件都需要在系统软件的支撑下运行。支撑软件是在系统软件的基础上研制的，它包括进行计算机绘图作业时所需的各种通用软件。应用软件则是在系统软件及支撑软件支持下，为实现某个应用领域内的特定任务而开发的软件。下面分别对这 3 类软件进行具体介绍。

（一）系统软件

系统软件主要用于计算机的管理、维护、控制、运行，以及计算机程序的编译、装载和运行。系统软件包括操作系统和编译系统。操作系统主要承担对计算机的管理工作，其主要功能包括文件管理(建立、存储、删除、检索文件)、外部设备管理(管理计算机的输入、输出等外部硬件设备)、内存分配管理、作业管理和中断管理。操作系统的种类很多，在工作站上主要采用 UNIX、Windows 2000/NT/XP 等；在微机上主要采用 UNIX 的变种 XENIX、ONIX、VENIX，以及 Windows 系列操作系统。编译系统的作用是将用高级语言编写的程序翻译成计算机能够直接执行的机器指令。有了编译系统，用户就可以用接近于人类自然语言和数学语言的方式编写程序，而翻译成机器指令的工作则由编译系统完成。这样就可以使非计算机专业的各类工程技术人员很容易地用计算机来实现其绘图目的。目前，国内外广泛应用的高级语言 FORTRAN、PASCAL、C/C++、Visual Basic、LISP 等均有相应的编译系统[2]。

（二）支撑软件

支撑软件是计算机绘图软件系统中的核心，是为满足计算机绘图工作中一些用户的共同需要而开发的通用软件。近 30 多年来，由于计算机应用领域迅速扩大，支撑软件的开发研制有了很大的进展，推出了种类繁多的商品化支撑软件。

（三）计算机绘图应用软件

应用软件是在系统软件、支撑软件的基础上，针对某一专门应用领域而开发的软件。这类软件通常由用户结合当前绘图工作的需要自行研究开发或委托开发商进行开发，此项工作又称为“二次开发”。能否充分发挥已有计算机绘图系统的功能，应用软件的技术开发工作是很重要的，也是计算机绘图从业人员的主要任务之一

四、图形操作的基本知识介绍

（一）CD和 CDC类的介绍

CDC是设备环境类的基类直接由 CObject 派生。是 GDI的关键元素，它代表了物理设备。每一个 C++设备环境对象都有相对应 Windows设备环境，并通过一个 32 位类型的 HDC句柄来标识。CDC类的虚拟性使我们可以很容易的做到编写同时适用于多种设备的代码。例如 OnDraw函数的 既可以适用于显示器、还可以适用于打印预览和打印，只需要在 CView::OnDraw函数的 pDC参数指向不同的对象类[3]。

CClientDC 和 CWindowDC是显示设备环境类，都是由 CDC派生而来，区别在于 CClientDC 是窗口的客户区不包括边框、标题栏和菜单栏，（0，0）指客户区域的左上角。CWindowDC 的（0，0）指整个屏幕的左上角，这意味着我们可以在显示器的任意地方绘图，包括窗口边框、标题栏和菜单栏等等。CWindowDC一般应用在框架窗口，而不是视图窗口。

CDC对象被创建后一定要在合适的时候将它删除掉，如果忘记了删除设备环境对象则会造成内存丢失。如何做才能避免出现这个问题呢，我们应该在堆栈中构造对象。

看例子

```
// 例子
void CMyView::OnLButtonDown(UINT nFlags,CPoint point)
{
CRect rect;
CClientDC dc(this); //      在堆栈中构造设备环境对象，用一个窗口指针 this  作参数。
dc.GetClipBox(rect);      //GetClipBox  函数是一个虚函数，作用是可以获得选定区域的尺寸
}
//析构函数在函数返回时自动调用，也就完成对设备环境对象的删除。
书上还给出了另一种写法：
void CMyView::OnLButtonDown(UINT nFlags,CPoint point)
{
CRect rect;
CDC * pDC=GetDC(); // 通过调用 CWnd的 GetDC()函数获得设备环境指针
pDC->GetClipBox(rect); //      可以获得选定区域的尺寸
ReleaseDC(pDC); //  一定不能忘记，释放设备环境。（书上写错了）
}
```

创建的设备环境对象具有一些默认的特性，通过 CDC 类的成员函数可以设定这些特性。例如前一篇笔记用到的刷子、映射模式等等。我们还可以通过重载 SelectObject 函数将 GDI 对象选进设备环境中。

（二）CDI和 CGdiobject 类

GDI 对象是通过 CGdiObject 派生类的 C++对象来表示的[4]。

CBrush 是一个 GDI的派生类，它在 MFC中的层次结构是这样的：CObject 派生 CGdiObject 派生 CBrush，明白了吧。CGdiObject 是所有 GDI对象的抽象基类。下面列出的是 GDI派生类的列表：

CBitmap：位图是一种位矩阵，每一个显示象素都对应于其中的一个或多个位，可以用来表示图象，也可以用来创建刷子

CBrush：刷子定义了一种位图形式的象素，可以用来对区域内部填充颜色。

些依赖某种设备。
CPalette ：调色板是一种颜色映射接口，它允许应用程序在不影响其他应用程序的前提下，可以充分利用输出设备的颜色描绘能力。
CPen: 笔是一种用来画线及绘制有形边框的工具，可以指定它的颜色及宽度，并可以指定画虚线、点线还是实线。
CRgn 区域是由多边形、椭圆二者组合形成的一种范围，可以用来进行填充、裁剪、鼠标点中测试等等。
以上很容易理解，可以用 WINDOWS 的画图帮助我们理解。
CGdiObject 类很眼生，看过很多代码就没有看到过它，原因是由于 CGdiObject 类是所有 GDI 对象类的虚拟基类，所以我们不必创建 CGdiObject 类的对象，可以直接构造它的派生类的对象，例如这样
CPen newPen(PS_DASHDOTDOT,2,(COLORREF) 0);黑色的笔宽度为 2
但需要注意的是 CFont 和 CRgn的对象建立需要先调用默认的构造函数来构造 C++对象，然后再调用相应的创建函数如：
CreateFont 或 CreatePolygonRgn 等。
CGdiObject 类有一个虚拟的析构函数，它派生类的析构函数需要将与 C++对象相关联的 GDI 对象删除掉，一定要在退出程序之前把构造的 CGdiObject 派生类对象干掉。因为一个没有释放的 GDI对象会占用很多的内存。
让我们用一个例子跟踪一下 GDI对象

```
void CMy10View::OnDraw(CDC* pDC)
{
pDC->MoveTo (10,10);
pDC->LineTo (110,10);
CPen newPen(PS_DASHDOTDOT,10,(COLORREF) 192);红色的笔宽度为 10
CPen * pOldPen=pDC->SelectObject (&newPen);
//在将新对象选进设备环境的同时返回指向前一次被选对象的指针。作用保存原来的对象，以便完成任务时恢复它。
pDC->MoveTo (10,20);
pDC->LineTo (110,20);
pDC->SelectObject (pOldPen);//      把原来的对象恢复
pDC->MoveTo (10,30);
pDC->LineTo (110,30);
}
```

屏幕上应该显示三条线，第一条和第三条一样颜色和粗细因为他们都是用的设备环境默认的 CPen对象，第二条是一条用我们自己设定的 CPen对象。我们可以看出在将新对象选进设备环境的同时返回指向前一次被选对象的指针。作用保存原来的对

Windows 还包含有一些可以利用的库存对象，它们不会被删除，因为 Windows对企图删除它们的动作不予理睬。我们可以用 SelectStockObject　函数将它们选进设备环境。下面列出的是所有的有关刷子、笔、字体和调色板的库存对象。

由于 SelectObject　函数返回的 GDI C++对象指针具有临时性，当程序的空闲处理阶段或者控制函数返回时应用程序框架会将临时的 C++对象删除，我们不能简单的把这一指针保存在类的数据成员中，而应该借助 GetSafeHdc 函数将它转化为 Windows的句柄，以便持久的保存 GDI的标识。

```
void CMy10View::OnDraw(CDC* pDC)
{
HPEN m_hPen; // 一个指向 CPen对象的指针
pDC->MoveTo (10,10);
pDC->LineTo (110,10);
COLORREF) 192); // 红色的笔宽度为 10
CPen *  pOldPen=pDC->SelectObject　(&newPen); //　在将新对象选进设备环境的同时返回指向前一次被选对象的指针。作用保存原来的对象，以便完成任务时恢复它。
                                                    获得并保存原来对象的句柄
pDC->MoveTo (10,20);
pDC->LineTo (110,20);
pDC->SelectObject (CPen::FromHandle (m_hPen));//　　把原来的对象恢复，和例子 7-2不同的是通过句柄
pDC->MoveTo (10,30);
pDC->LineTo (110,30);
}
```

补充一下 IGDI 派生类的 Windows handle type　列表

CPen HPEN

CBrush HBRUSH

CFont HFONT

CBitmap HBITMA

CRgn HRGN