

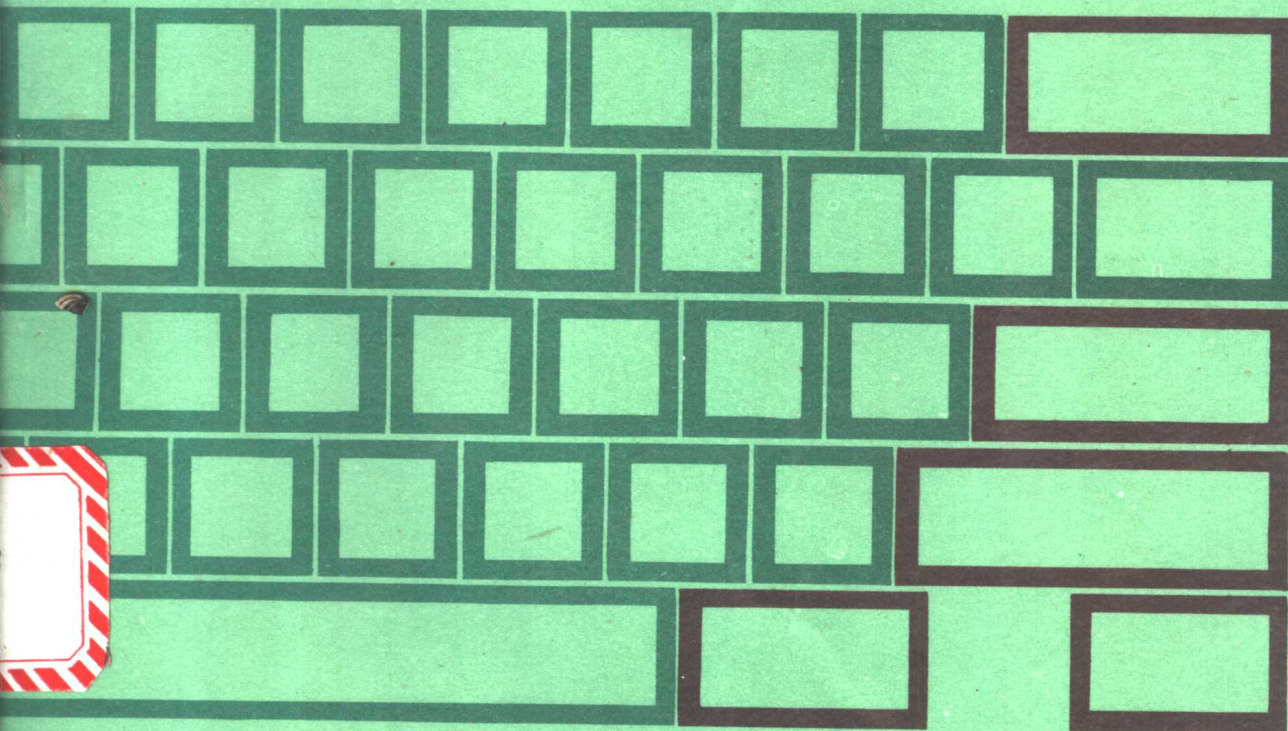
高等学校教材

图形核心系统

GKS 及其应用

魏朋三 司 岩 主编

蒋本铁 李 荆 董伟光 编著



东北工学院出版社

内容简介

本书对计算机图形的国际标准ISO-7942图形核心系统(GKS)的原理和基本概念作了系统介绍,主要包括:输出原语,坐标系及变换,工作站,图段,输入功能,GKS元文件及GKS环境,最后一章给出了在微机上应用GKS绘图的方法、步骤及应用实例,附录中给出了全部GKS功能函数。

本书可供从事CAD(计算机辅助设计)工作的工程技术人员阅读,也可作为高等院校师生和研究生的教学参考书。

图形核心系统 GKS 及其应用

魏朋三 司岩 主编

蒋本铁 李荆 董伟光 编著

东北工学院出版社出版 辽宁省新华书店发行

(沈阳·南湖) 大连海运学院出版社印刷厂印刷

开本: 787×1092 1/16 印张: 13 字数: 324 千字

1991年5月第1版 1991年5月第1次印刷

印数: 1~2 000 册

责任编辑: 长 仁

责任校对: 张德喜

封面设计: 唐敏智

描 图: 何建刚

ISBN 7-81006-275-1/TP·11 定价: 3.81 元

前 言

计算机图形是计算机科学中十分活跃和迅速发展的领域之一。我们不仅已经知道图形作为一种有效的通讯方法的价值，而且也认识到用计算机进行图形对话的能力正在变革着计算机在各个领域中使用的方法。

随着新的交互图形设备和工作站的不断问世，各生产厂家又为自己的设备配备了自己的支持软件。这样，图形应用程序的可移植性问题就摆在了从事计算机图形设计的科技人员面前，因而促进了计算机图形软件的标准化的研究。计算机图形标准的制定花费了很长时间。由图形学专家组成的专门国际组织工作了多年，直到 1985 年 8 月国际标准化组织(ISO)才公布了以图形核心系统(GKS)作为计算机图形标准。

GKS 是一个为应用程序服务的基本的图形系统，它提供了一组用于计算机图形程序设计的功能函数，定义了 6 种图形输出元素和 6 种逻辑输入设备，引进了工作站的概念，设置了 3 种坐标系，可进行窗口和视口间的转换，允许存贮和动态地修改画面。由于 GKS 对具体设备是独立的，因此使用 GKS 编制的图形软件具有很好的可移植性。GKS 定义了一个独立于语言的图形系统的内核，为了和程序语言集成在一起，GKS 将嵌入到一个服从于某个语言专有的惯例的语言依赖层中。

诚然，由于计算机图形所涉及的领域越来越广泛，现有的图形标准已经满足不了实际发展的需要，这些标准有待于重新估价，进一步完善和扩充，第二代图形标准不久将会问世。

自从 GKS 成为国际标准以来，许多国家实现了自己的 GKS 系统，并与 CAD 结合在一起设计出面向应用的交互图形系统。在一些国家还成立了有关 GKS 的实现、测试的专门机构以推进研究和探讨工作。

我国的许多部门在 GKS 的研究和应用方面也进行了卓有成效的工作，许多高等学校也把 GKS 纳入图形学的教学。由浅入深、全面系统地介绍 GKS 的基本概念及其应用是本书的任务，相信将会对 GKS 的研究、应用和教学起到一定的推动作用。

本书在编写过程中考虑到了不同层次的人员的需要，1 中介绍 GKS 的基本概念，2, 3 说明输出原语、坐标系及变换，4, 5 说明工作站和图段的概念，6 介绍输入功能，7, 8 介绍 GKS 的元文件和环境，9 通过实例给出了在 IBM-PC 机上应用 GKS 与 FORTRAN 语言和 C 语言的联编而实现的图形应用的方法。

本书由魏朋三、司岩主编，参加编写的还有蒋本铁、李荆、董伟光等同志。在编写过程中，得到东北工学院计算中心龙光、高禹、李东黎同志以及王延红同志的热情帮助，受到王泳嘉教授的大力支持和鼓励。在此仅向他们致以衷心感谢。

由于时间仓促，水平有限，书中一定存在许多错误和不当之处，恳请有关专家、同行和广大读者批评指教。

编者

1990.3 于东北工学院

目 录

前 言

1. 概 述	(1)
2. 输出原语	(10)
2.1 基本概念.....	(10)
2.2 多段线及其属性.....	(15)
2.3 多点记号及其属性.....	(22)
2.4 填充区域及其属性.....	(24)
2.5 正文及其属性 正文输出原语 正文的属性	(30)
2.6 象元阵列.....	(38)
2.7 广义绘图原语.....	(39)
2.8 与各原语有关的属性设置.....	(40)
3. 坐标系和变换	(43)
3.1 世界坐标系.....	(43)
3.2 规格化设备坐标系.....	(43)
3.3 规格化变换.....	(44)
3.4 裁 剪.....	(47)
4. 工作站	(50)
4.1 基本概念.....	(50)
4.2 工作站描述表.....	(52)
4.3 工作站状态表.....	(57)
4.4 应用到工作站上的 GKS 功能函数	(61)
4.5 工作站控制.....	(67)
4.6 工作站变换.....	(70)
4.7 延迟画面的变化 延迟方式 隐含再生成方式	(73)
5. 图 段	(77)
5.1 图段的建立和删除.....	(77)
5.2 检取标识符与图段.....	(79)
5.3 与图段有关的其它功能函数.....	(79)
5.4 图段的属性.....	(80)
可见性 醒目性 可检测性 图段的优先级 图段变换 图段变换与裁剪	
5.5 图段存储器.....	(86)
6. 输入功能	(95)

6.1	逻辑输入设备的分类和标识	(95)
6.2	逻辑输入设备的运行方式 请求方式 采样方式 事件方式	(96)
6.3	逻辑输入设备的模型	(97)
	交互操作 度量 触发器 提示和应答 承认	
6.4	逻辑输入设备的进一步叙述	(99)
	定位器和笔划器 定值器 选择器 检取器	
6.5	逻辑输入设备的初始化	(105)
	初始值 提示 / 应答类型 应答区域 输入数据记录	
6.6	设置逻辑输入设备的运行方式	(109)
6.7	请求输入	(110)
6.8	采样输入	(112)
6.9	事件输入	(115)
	输入队列和当前事件报告 等待和删除事件队入口的功能函数 读取输入功能函数	
6.10	FORTRAN 程序实例	(119)
7.	GKS 元文件	(124)
7.1	GKS 元文件的内容	(124)
7.2	GKS 元文件的接口	(124)
7.3	元文件工作站	(125)
7.4	GKS 文件的输出	(126)
	图形的直接输出 借助图段的输出 用户数据的输出	
7.5	GKS 元文件的输入	(128)
8.	GKS 的环境	(131)
8.1	GKS 的初始化	(131)
8.2	GKS 的级	(131)
8.3	错误处理 错误的处理过程 错误状态表 错误信息表	(132)
8.4	GKS 的运行状态	(138)
8.5	GKS 状态表	(139)
8.6	询问功能函数	(142)
9.	在 IBM-PC 微型机上应用 GKS 实例	(144)
9.1	系统构成	(144)
9.2	实现 GKS 应用的一般步骤	(145)
9.3	与 FORTRAN 语言联编的 GKS 过程	(145)
9.4	与 C 语言联编的 GKS 过程	(166)
附录	功能函数简表	(171)
参考文献		(201)

1. 概 述

计算机绘图是一门崭新的学科,早在 30 多年前第一台计算机图形显示装置(在 MIT'S "whirl wind" 计算机上)就问世了,但是计算机绘图系统所必备的设备(如绘图仪、图形显示器)还相当昂贵,一些中小用户根本无力购买。

随着计算机技术的发展,到了 60 年代,新的交互图形设备的价格不断降低,成为人们能够购买的商品。但不同厂家提供的设备配备了各自的软件,而这些软件又是相互独立的,若要增加或更换一种新的设备就需要修改原来编制的软件,造成人力资源的巨大浪费。在计算机绘图设备变得越来越廉价的年代,是不能允许这种浪费的。人们希望在不修改源程序的情况下使用性能更好的绘图设备,因而推进计算机图形软件标准化的进程。

1.1 SEILLACI 会议

鉴于这种情况,IFIP(International Federation for Information Processing)的图形分会开始着手计算机图形软件的标准化研究工作。欧美国家从事计算机绘图研究的 25 名专家 1976 年应邀在法国的 SEILLACI 组成了专门的研究计算机绘图方法论的工作组。这个工作组集中讨论了计算机绘图标准化的核心问题和标准化应遵循的原则。

专家们认为标准化的主要目的是要明确地定义“可移植性”,它包括四个方面:

- ① 应用程序的可移植性;
- ② 设备的独立性;
- ③ 图形数据的可移植性;
- ④ 教学的可移植性。

应用程序的可移植性是标准化所致力实现的目标。建立核心系统与应用程序之间的标准接口可以实现这一目标。把实际物理设备的功能抽象定义成逻辑输入/输出设备,以此作为核心系统与设备驱动器之间的接口。这样,应用程序只与逻辑设备打交道,给应用程序的移植提供了极大的方便。设备驱动器使得核心系统与物理设备成为两个独立的部分,使得应用程序可以在不同厂家生产的硬设备之间移植。

图 1-1 表示的是图形系统的各个接口。

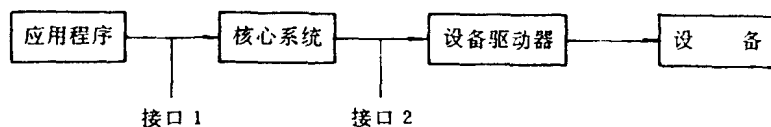


图 1-1 图形系统的接口

工作组讨论的另一个问题是应该把哪些功能放到图形核心系统来完成,而哪些功能应该属于应用程序的范围,并提出了图形系统与模型化系统两个新的概念。

模型化系统允许人们把客观物体定义在自定的坐标系里,物体的观察模型可以产生在一个称作世界坐标系(WC)的参考坐标系里,而图形系统的工作就是把物体的模型映射到图形

显示器上, 这是由图形核心系统完成的。

SEILLACI 工作组对计算机图形软件的标准化工作起到了巨大的推动作用。

1.2 CORE 系统和 GKS 系统的发展

在此之前美国的 CORE 系统的发展比较缓慢, SEILLACI 工作组的工作促进了它的发展, 使它的设计规范与 SEILLACI 的目标趋于一致。CORE 系统是美国 GSPC(Graphics Standards Planning Committee)所提出的一个全三维图形软件系统。在进行了大量的工作以后, 1977 年第一次出版了该系统的草案。以后又不断地更改了它的版本, 增加了新的内容。尽管这样, CORE 系统与 GKS 系统相比还是呈现出许多不足之处。所以现在人们不再热衷于把 CORE 系统作为国际标准。

受到 SEILLACI 工作的激励, 联邦德国国家标准化组织(DIN)也加速了图形软件的标准化工作, 并于 1977 年第一次出版了二维 GKS 系统草案。接着又进行了系统的再设计, 扩充以及再检查。其工作吸收了以往各种图形软件包的优点, 特别是继承了美国 CORE 系统的优点, 并在图形工作站、输出原语、输出属性、逻辑输入设备和图段等概念上有了较大的发展。最后于 1985 年被国际标准化组织(ISO)接受为正式国际标准。

GKS 是一个不断发展的系统, 经过了几次功能扩充和性能改进, 先后推出了许多不同的版本:

1977 年 7 月 GKS Version 1 它仅包含输出。

1977 年 8 月 GKS Version 2 它包含了输出、事件和输入。

1978 年 9 月 GKS Version 3 它包含了输出、询问和输入。

1979 年 3 月 GKS Version 4.8 它是在 Version 3 的基础上设计的, 又增加了级(Level)、询问功能(Inquire Function)、光栅属性、二级变换、元文件(Metafile)、采样输入这些新的概念。

1980 年初 GKS Version 6.0

1980 年秋 GKS Version 6.2

1981 年 5 月 GKS Version 6.6

1982 年 3 月 GKS Version 7.0 联邦德国国家标准化组织提出了 GKS 7.0 与 FORTRAN 语言之间的接口建议。1982 年 11 月美国国家标准化委员会投票通过把 GKS 7.0 作为美国国家标准。

1983 年 GKS Version 7.2 GKS 被国际标准化组织接受为正式草案, 即 ISO/DIS7942。

1985 年 8 月 GKS 被接受为正式国际标准, 即 ISO-7942。

1.3 GKS 设计的原则和目标

GKS 提供了一个应用程序可以调用的功能函数(function)集合, 供绘图编程使用。它能够满足大部分应用所提出的要求。根据用户和设备的要求不同, 可把这个功能函数集合分成 9 个向下兼容的级(Level)。最低级仅仅是 GKS 功能函数的一个最小集合。级越高集合的内

容越丰富。推荐 GKS 作为计算机绘图软件标准的主要原因是:

- ① 使得包含图形内容的应用程序可以方便地在不同的设备上移植;
- ② 借助于应用去更好地了解整个图形系统的使用;
- ③ 可作为计算机图形设备制造商的指南,以便提供具有多种性能的设备。

为了达到上述目的, GKS 系统的设计必须要满足下面这些要求:

① GKS 必须包含绘图领域所提出的高性能要求和从简单的输出到高水平的交互应用。

② GKS 必须采用统一的方法控制所有的图形设备,其中包括:矢量和光栅设备、胶片摄影机、存储管显示器和彩色显示器等。

③ GKS 在满足应用的前提下,必须使系统的使用简单易行。

这些要求已经形成了 GKS 任何设计中都必须遵循的如下一系列准则,以用于判断和确定系统的不同设计方案:

- 一致性(Consistency) —— GKS 所提出的要求一定不是相互矛盾的。
- 兼容性(Compatibility) —— 一定不破坏任何其它的标准和实际准则。
- 正交性(Orthogonality) —— GKS 中的功能函数之间或模式之间一定是相互独立的,否则一定进行明确定义。

图形设备的设计准则:

• 设备的独立性(Device Independence) —— GKS 功能的设计必须允许使用这些功能的应用程序能够选择不同类型的输入/输出设备的性能。

• 设备的丰富性(Device Richness) —— 各种不同的输入/输出设备所具有的丰富功能都应该由 GKS 功能函数调用实现。

这些准则导致了一个新概念的产生,即多个独立图形工作站的概念。这一概念首先是由 GKS 提出并付诸实现的。应用程序可以调用询问功能函数,询问每一个工作站的状态。

用户接口设计的准则:

- 透明度(Clarity) —— 用户能够容易地了解 GKS 里的概念和各个功能的含义。
- 错误处理能力(Error handling) —— 一些应用程序或系统本身发生故障,系统一定要清楚地把信息反馈给用户,同时在对系统和应用程序的影响尽可能小的情况下及时地排除故障。

透明度和错误处理能力是能否维持系统与用户良好关系的重要因素。保持系统具有良好的透明度,不但是系统设计的要求同时也是系统描述的要求。GKS 的标准可以分成两部分,其一是逻辑数据结构的描述,由此确定系统的状态。其二是功能函数及其作用的描述。错误处理规范是 GKS 标准的一个组成部分。

以上几个设计准则之间有些是相互抵触的,这就需要采用折衷的方法满足设计要求。GKS 应该有一个较容易理解的结构和功能函数集合。使广大用户设计出可移植的独立于设备的应用程序。

1.4 GKS 的实现以及在图形软件中的地位

GKS 的实现是靠操作系统和硬件支持的, 其实现方法可以有多种。现在多数采用的是 FORTRAN, C 和 PASCAL 语言。虽然 GKS 的实现方法是灵活的, 但必须遵守下面这些准则:

- 可实现性(Implementability) —— 在大多数操作系统和设备上能够使其主语言(host language)支持 GKS 的功能。
- 语言的独立性(Language - independence) —— 能够使大多数标准程序设计语言实现 GKS 标准。
- 有效性(Efficiency) —— 在没有延时算法的前提下, 实现 GKS 标准。这一点在交互系统里考虑到响应速度时显得尤为重要。
- 坚固性(Robustness) —— 操作人员必须能够尽可能地避免系统的硬件和软件故障。

在 GKS 标准里定义了一个功能函数集合。在系统实现的过程中, 这些功能函数是靠某种程序设计语言写成的子程序实现的, 被称为语言层。应用程序可以采用语言层所提供的功能函数完成不同的任务。

图形核心系统作为通用的基础软件为计算机绘图提供各种服务。图 1-2 的 GKS 层次模型表示了图形核心系统在整个图形系统中的地位。

每一层可以调用相邻底层的功能, 这样应用程序可以存取下面所有层的信息, 而所有的绘图操作都需要调用 GKS 的功能函数完成。核心系统与语言接口层之间的接口是独立于语言的应用接口。它是由 GKS 标准所定义的。在语言的接口层和面向应用层之间的接口是依

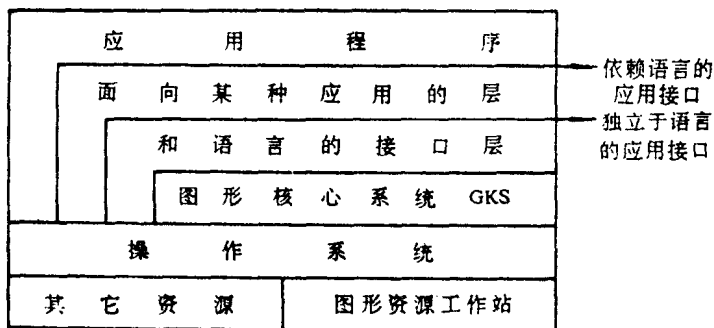


图 1-2 GKS 的层次模型

赖语言的应用接口如 FORTRAN, PASCAL 等。

一个软件功能实现的优劣不是随意评价的, 需要进行一系列的论证测试。论证就是通过测试系统, 证明系统的实现是否正确, 与其标准相一致。其中包括模式的论证, 程序结构的论证, 接口测试等。在 GKS 研究中, 令人鼓舞的发展是在世界各地建立了许多 GKS 测试研究室, 专门研究测试已经实现的 GKS 软件系统, 证明它们是否符合发表的 GKS 标准。

1.5 GKS-3D

通常计算机二维绘图系统能够满足应用需要的 90% 左右, 这也是首先把 GKS 设计成二维系统的重要原因之一。现在 GKS-2D 已经成为国际标准。人们自然希望能够直接把 GKS-2D 扩充到 GKS-3D。对此, 国际标准化组织成立了专门的 3D 工作组。讨论和研究

3D 图形标准的基本模式、范围、作用和目的等。其中需要解决的最重要问题是 GKS-3D 的流程以及与 GKS-2D 之间的关系。

目前, GKS-3D 已成为国际标准化图形软件的候选之一, 而美国的 PHIGS(分层结构交互式图形标准) 也作为候选之一而引起人们的注视。PHIGS 不仅提供了三维图形功能, 更重要的是具有多级分层的图段数据结构。由于是相同的 3D 委员会在 GKS-3D 和 PHIGS 上工作, 所以这两个系统可能会溶为一体。

1.6 GKS 的用户

对于不同层次的用户, 使用 GKS 的含义是不同的。图 1-3 表示了不同类型的 GKS 与用户之间的关系。

GKS 实现者的任务是根据 GKS 的标准规范实现不同级别的 GKS 系统。他们使用如程序设计语言一样的基本软件, 以及连接和载入程序或输入/输出子程序, 编写各种图形设备的驱动程序, 为 GKS 的应用程序员提供方便。

根据 GKS 标准实现了具体的 GKS 系统后, 可使用 GKS 的各种功能编写各种应用图形程序。

操作员使用由应用程序员编写完的应用图形程序, 在图形工作站上操纵各种图形设备。

本书为应用程序员和操作员提供了一本学习和掌握 GKS 系统的参考资料。

一个新兴的技术——计算机绘图正在标准化的进程中。这一工作缩短了用户和厂商的距离, 兼顾了两者的利益。如今在计算机绘图系统中硬件设备的价格不再占有主要地位, 人们已经把主要精力移到软件上, 去组成一个解决实际问题的系统。这就是图形标准软件的重要意义所在。

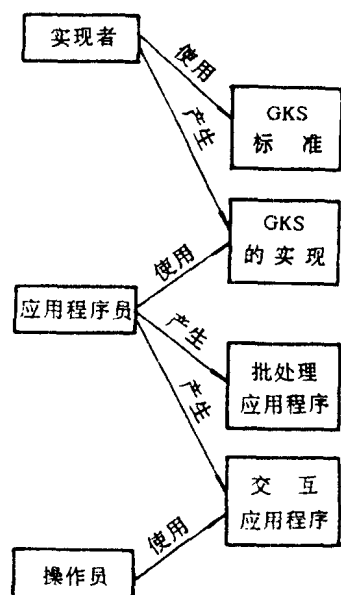


图 1-3 GKS 与用户间的关系

1.7 GKS 重要术语的定义

(1) 收悉, 承认(acknowledgement) 向逻辑输入设备上的操作员发出的指示, 是一个引发器已经击发的输出。

(2) 宽高比 (aspect ratio) 一个用来描述在某一个坐标系中矩形形状的 X 对 Y 的比值 (例如工作站窗口或工作站的视口)。

(3) 输出原语的外表(aspect of primitives) 可以改变输出原语外观的方法。某些外表由输出原语属性直接控制, 某些则通过束表间接进行控制。图段内的输出原语有一个通过包含它们的图段进行控制的外表, 例如醒目; 而图段外的输出原语就不能这样做。

(4) 属性(attribute) 一种能附加到显示元素(输出原语)或者图段上的特殊性质。例如醒目, 字符高度。在 GKS 中, 某些工作站的性质称为工作站属性。

(5) 基线(baseline) 一条字符框内的水平线, 对很多字符的定义, 在外观上是字符外形

的下界。字符的下伸部分要伸到该线下面。同一字体的所有基线处于字符框内的同一位置。

(6) 束索引值(bundle index) 某个输出原语束表中的一个索引值。它定义了该图原和工作站有关的外表。

(7) 束表(bundle table) 一个和工作站有关并和某一个输出原语相关联的表。表中的项规定了一个图原所有和工作站有关的外表。在 GKS 中, 下述输出原语有束表: 折线、多点记号、正文和填充区。

(8) 字形上线(capline) 一条字符框内的水平线, 对很多字符的定义来说, 它在外观上是字符外形的上界。字符的上伸部分要伸到该线的上面。在某些语言中, 在字符上面的附加记号(例如重音记号)可以定义在该线的上面。同一字体的所有字形上线处于字符框内的同一位置。

(9) 象元阵列(cell array) 一个由相同尺寸矩形单元的矩形网格组成的 GKS 输出原语, 其中的每个矩形小单元有同一颜色。这些小单元不需要一对一地映到图素上去。

(10) 中心线(centreline) 一条垂直的平分了字符框的线(见图 3)。

(11) 字符框(character body) 一个用于字体设计以定义字符外形的矩形。同一种字体的所有字符框有相同的高度。

(12) 选择设备, 选择器(choise device) 一个 GKS 的逻辑输入设备, 它提供了一个用于确定一组选择中某一个的非负整数。

(13) 剪取(clipping) 去掉显示元素中处于给定边界外面的部分, 通常的边界是窗口或视口。

(14) 颜色表(colour table) 一个和工作站有关的表, 其中的表项指定了定义某一颜色的红、绿、蓝强度值。

(15) 坐标图形(coordinate graphics), 线图形(line graphics) 由显示命令和坐标数据来产生显示图象的计算机图形。

(16) 设备坐标(DC)(device coordinate) 用一个和设备有关的坐标系来表示的坐标。在 GKS 中, 在那些能够产生精确尺寸图象的设备上, DC 的单位是米, 而在其他设备上, 则用适当的依赖于工作站的单位。

(17) 设备驱动程序(device driver) GKS 实现中用于支持图形设备的与设备有关的部分。设备驱动程序产生和设备有关的输出, 处理和设备有关的交互。

(18) 设备空间(device space) 由显示设备上可寻址的点所定义的空间。

(19) 显示设备(display device), 图形设备(graphics device) 一个能在上面表示显示图象的设备(例如刷新显示, 存储管显示, 绘图机)。

(20) 显示图象(display image) 图画, 图片, 画面(picture) 在任意一个时刻, 表示在显示面上的一组输出原语或图段。

(21) 显示空间(display space) ①设备空间中对应用于显示图象区域的部分; ②诸如数字化仪表一类输入设备上的工作空间。

(22) 显示面(display surface), 观察面(view surface) 在显示设备中, 显示图象出现在其上的介质。

(23) 应答(echo) 在显示控制台上, 给操作员的一个有关输入设备提供的最新值的即时特性通知。

(24) 逸出(escape) GKS 中用于接触与实现或设备有关特性的功能, 它不同于产生图形

输出的功能，也未由 GKS 在其他地方提及。

(25)反馈(feedback) 应用程序解释了逻辑输入值后指给操作员看的输出。

(26)填充区域，填充区(fill area) 由多边形(封闭边界)组成的一个 GKS 输出原语，它可以是空洞，或用单一颜色、图案或阴影线进行填充。

(27)填充区束表(fill area bundle table) 一张使填充区图原的所有和工作站有关的外表的一组值和一个填充区束索引值相关的表。在 GKS 中，这张表包含了由内部样式、样式索引值和颜色索引值组成的表项。

(28)广义绘图原语(GDP)(Generalized Drawing Primitive) 一种用于涉及诸如曲线绘图等工作站专有的几何能力的输出原语。

(29)GKS 级(GKS Level) 由值域 0 到 2 和 a 到 c 中两个值在一起定义的一个 GKS 功能性能能力的集合。一个 GKS 的实现精确地提供了某一级的功能。

(30)GKS 元文件(GKSM)(GKS Metafile) 一个能由 GKS 读写的顺序文件，它用来长期储存(传输和传送)图形信息。

(31)半高线(halfline) 在字符框内处于字形上线和基线之间的水平线(见图 3)，对同一字体的一个水平字符串，它将出现在垂直方向的正中间。同一字体中的所有半高线都处于字符框中同一位置。

(32)阴影线(hatch) 填充由填充区图原指定的多边形内部的一种方法。它的内部用一组或多组平行线进行填充。

(33)醒目，高光，增辉(high lighting) 一种通过改变图段的视觉属性来强调图段不依赖于设备的方法。例如，内烁。

(34)实现必备(implemption mandatory) 描述需要在所有 GKS 实现的所有工作站上一致地实现的特性。

(35)输入类，输入类别(input class) 一组在逻辑上功能等价的输入设备。在 GKS 中，输入类是：LOCATOR(定位器)，STROKE(笔划器)，VALUATOR(定值器)，CHOICE(选择器)，PICK(捡取)和 STRING(字符串)。

(36)询问功能(inquiry function) 一种目的在于根据 GKS 的当前状态或者 GKS 实现的某些固定特性来返回值的 GKS 功能。它们对 GKS 状态或显示图象没有影响。

(37)定位设备，定位器(locator device) 一种提供用世界坐标表示的位置和规格化变换号的 GKS 逻辑输入设备。

(38)逻辑输入设备(logical input device) 一个逻辑输入设备是给程序以逻辑输入值的一个或多个物理设备的抽象。GKS 中的逻辑输入设备可以是下述类型：LOCATOR，STROKE，VALUATOR CHOICE，PICK 和 STRING。

(39)逻辑输入值(logical input value) 由一个逻辑输入设备给出的值。

(40)记号，标号(marker) 用于标记某个位值，具有某种外表的小图形。

(41)量度(measure) 由一个或多个物理输入设备决定，由物理设备给出值映射出来的(与一个逻辑输入设备相关的)值。由逻辑输入设备发出的逻辑输入值是该量度的当前值。

(42)MI GKS 元文件输入的缩写，一个工作站类别。

(43)MO GKS 元文件输出的缩写，一个工作站类别。

(44)规格化变换(normalization transformation)，观察变换(viewing transformation)，窗口

到视口变换(window-to-viewport transformation) 一个把窗口的边界和内部映射到视口的边界和内部的变换。在 GKS 中, 该变换把一个位置的世界坐标映射到规格化坐标。

(45)规格化设备坐标(NDC)(normalized device coordinates) 一个规定在独立于设备的中间坐标系中, 并规格化到某个范围(典型是 0 到 1 之间)的坐标。在 GKS 中间状态时, 坐标可能在定义范围之外, 但有关的剪取信息将保证输出不超出坐标值域 $[0,1] \times [0,1]$ 。

(46)操作员(operator) 操纵物理输入设备以改变逻辑输入设备的量度和击发引发器的人。

(47)输出原语(output primitive), 图原(graphic primitive), 显示元素(display element) 能用来构造显示图象的基本图形元素。GKS 中的输出原语是: 折线、多点记号、正文、填充区域、象元阵列和广义绘图原语。

(48)捡取设备、捡取器(pick device) 一种提供附在一个输出原语上的捡取标识符和相关联的图段名的 GKS 逻辑输入设备。

(49)捡取标识符(pick identifier) 一个附在图段内个别的输出原语上, 并能由捡取设备返回的名字。同一个捡取名可以赋予不同的输出原语。

(50)图素(pixel), 图元素(picture element) 显示面上能独立地赋予颜色或亮度的最小元素。

(51)折线(polyline) 一个由一组相连的线组成的 GKS 输出原语。

(52)折线束表(polyline bundle table) 一张使折线图原的全部依赖于工作站的外表的一组值和一个折线束索引值相关联的表。在 GKS 中, 这张表包含了由线型、线宽比例因子和颜色索引值组成的表项。

(53)多点记号(polymarker) 一个由一组记号标志的位置所组成的 GKS 输出原语。

(54)多点记号束表(polymarkere table) 一张使多点记号图原的全部依赖于工作站的外表的一组值和一个多点记号束索引值相关联的表。在 GKS 中, 这张表包含了由信号类型、记号大小比例因子和颜色索引值组成的表项。

(55)图原属性(primitive attribute) (输出原语的)图原属性值由应用程序以与工作站无关的方式选择, 但可以有与工作站有关的效果。

(56)提示(prompt) 向操作员指示某个逻辑输入设备已可用的输出。

(57)光栅图形(raster graphics) 显示图象是由按行列排列的图素阵列组成的计算机图形。

(58)旋转(rotation) 使全部或部分显示图象绕一个轴转动。在 GKS 中, 这种能力只局限于图段。

(59)定比(scaling), 缩放(zooming) 通过把显示元素的坐标乘一个常数值的方法放大或缩小全部或部分显示图象。在 GKS 中, 这种能力只局限于图段。在两个不同方向取不同比例时, 需要用两个常数值。

(60)图段(segment) 一组能够做为一个整体进行操纵的显示元素。

(61)图段属性(segment attributes) 只能用于图段的属性。在 GKS 中, 图段属性是可见性、醒目性、可检测性、图段优先级和图段变换。

(62)图段优先级(segment priority) 用来决定几个重叠的图段在进行输出和输入时谁优先的图段属性。

(63)图段变换(segment transformation) 一个使得由图段定义的显示元素以不同的位置(平移),大小(比例)和/或方向(旋转)出现在显示面上的变换。

(64)字符串设备(string device) 提供字符串作为其结果的 GKS 逻辑输入设备。

(65)笔划设备(stroke device) 提供用世界坐标表示的一串点和一个规格化变换号的 GKS 逻辑输入设备。

(66)正文(text) 由字符串组成的 GKS 输出原语。

(67)正文束表(text bundle table) 一张使正文图原的全部依赖于工作站的外表的一组值和一个正文束索引值相关联的表。在 GKS 中,这张表包含了由正文字体和准确度、字符扩展因子、字符间隔和颜色索引值所组成的表项。

(68)正文字体和准确度(text font and precision) 由字体和准确度两个成分组成的 GKS 正文的外表,它们一起决定了输出到显示面上字符的外形。此外,准确度还描述了应用程序所要求的其他正文外表匹配的准确程度。按准确性增加的次序,准确度分为 STRING, CHAR 和 STRKE。

(69)平移(translation),移动(shift) 对全部或部分显示图象的位置附加一个常量的位移。在 GKS 中,该能力只限于图段。

(70)引发器,触发器(trigger) 操作员用来指出重要时刻的一个或一组物理输入设备。

(71)定值设备,定值器(valuator device) 一个提供实数的 GKS 逻辑输入设备。

(72)视口(view port) 由应用程序指定的规格化设备坐标空间中的一部分。在 GKS 中,本定义只限于规格化设备空间中的矩形区域,用于定义规格化变换。

(73)窗口(window) 虚拟空间中一个预定义部分。在 GKS 中,该定义部分只限于世界坐标空间中的矩形区域,它用定义规格化变换。

(74)工作站(work station) GKS 是基于抽象图形工作站的概念的,它提供了应用程序用以控制物理设备的逻辑界面。

(75)依赖于工作站的图段存储器(WDSS)(Workstation Dependent Segment storage) 用于图形输出的工作站上的图段存储器,图段不能从 WDSS 转送到另一工作站上。

(76)独立于工作站的图段存储器(workstation independent segment storage) 一种图段能在上面储存,并在以后转送到其他工程站上的特殊工作站类型。

(77)工作站必备(workstation mandatory) 工作站必备描述了必须在 GKS 实现的所有工作站上一致地实现的特性。

(78)工作站变换(workstation transformation) 一个把工作站窗口的边界和内部映射到工作站视口(显示空间的一部分)的边界和内部,并保持了宽高比的变换。在 GKS 中,本变换把用规格化设备坐标表示的位置映射到设备坐标。保持宽高比的结果是工作站窗口的内部可能不映射到工作站视口的全部。

(79)工作站视口(workstation viewport) 当前选为输出图形用的部分显示空间。

(80)工作站窗口(workstation window) 要呈现在显示空间上的规格化设备坐标系中的一个矩形区域。

(81)世界坐标(WC)(world coordinate) 一个应用程序用来规定图形输入和输出的独立于设备的笛卡尔坐标系。

2. 输出原语

2.1 基本概念

由 GKS 产生的图形输出建立在两组基本要素上：输出原语 (output primitives) 和输出原语的属性 (output primitive attributes)。

输出原语是图形输出设备可以完成的基本动作的抽象。GKS 提供 6 种输出原语：

一个线原语 多段线 (POLYLINE) 即一组依次连结给出点列的直线，也称为折线。

一个点原语 多点记号 (POLYMARKER) 即在给出的位置上生成某种类型的符号，这些符号统称为记号。

一个文字原语 正文 (TEXT) 即在指定的位置上生成字符串。

两个光栅原语

填充区域 (FILL AREA) 即根据给出的点列生成多边形区域。多边形区域内可能是中空的，也可能用统一的颜色、图案或者阴影线充填。

象元阵列 (CELL ARRAY) 即生成具有各自颜色的矩形象元的阵列，这是光栅设备上象素阵列的一种概括。然而这种原语的象元不必与显示软件设备上定义的象素一一对应。

一个广义目的的原语 广义绘图原语 (GENERALIZED DRAWING PRIMITIVE) 即 GKS 增加了工作站特殊的几何输出能力，例如画轮廓线、圆弧、椭圆弧等。这些图形由标识符、一组点和附加的数据描述。简称为 GDP。

图 2-1 给出 6 种输出原语的几种情况。

原语的定义只包括了该原语几何形状的信息，而更多的有关原语外表 (appearance) 的几何细节和信息则是通过输出原语属性增加到原语的定义中去的。所谓原语的属性是一种能附加到输出原语上的特殊性质，例如线宽、记号大小、颜色等。在 GKS 里属性可以定义和以不同的方式成束地结合到原语里。本章将详细介绍各原语的属性。

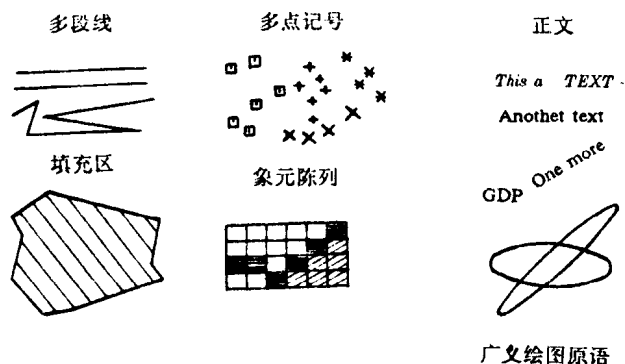


图 2-1 输出原语

GKS 将属性分为几何属性 (geometric attributes) 和非几何属性 (Non-geometric attributes)。几何属性影响原语的外形和尺寸大小，它由世界坐标系定义，或者以世界坐标应用到原语里。所谓世界坐标系 (World Coordinate System) 是指 GKS 在定义图形输入或输出时使用的依赖于设备的笛卡尔坐标系，简称为 WC。几何属性服从于所有的变换，但设置不同的变换将产生不同的结果。

表 2-1 列出了 GKS 中的全部几何属性。由表中可以看出, 仅正文原语和填充区域原语有几何属性, 其它原语没有几何属性。

非几何属性控制原语的外形 (aspect), 它们不能进行变换, 如颜色、线宽、记号大小等。GKS 不把线宽作为几何属性, 是因为绝大多数图形设备上根本没有可变换的线宽。在国际标准中如果不作出这样的规定显然是不合理的。记号大小也是不能变换的, 因为记号指

表 2-1 GKS 中几何属性			
	字符高度	填	图案大小
正	字符竖向	充	图案参考点
文	正文路径	区	
	正文对准	域	

定的点的范围为零。若要改变线宽 (或记号大小) 可以通过选择不同的线宽比例因子 (或记号大小比例因子) 实现, 而不依赖于变换。字符间隔和字符扩展因子影响原语的几何形状, 但 GKS 也将它们作为非几何属性处理, 这是因为它们的取值是字符高度的相对值, 是通过字符高度定义的。如果要改变字符间隔和字符的宽度需要根据字符高度, 而不需要经过变换。也可以认为这两种属性是特殊正文字体的一部分, 而不是独立的可变换的几何属性。

属性可以通过给出属性值直接指定, 或者通过为各自不同属性提供索引值间接 (用索引) 指定。GKS 中几何属性总是直接指定。颜色是一种不能被直接指定的属性, 而是由一个指向颜色表的索引间接指定。有些属性可以用直接和索引两种方式定义。表 2-5 列出这些属性。

索引属性可以分成两部分, 一部分是实现必备的静态表中的索引属性, 一部分是可调整的工作站表中的索引属性。表 2-2 列出全部实现必备的静态表, 这些值的范围由实现者确定, 实现者必须提供各项的最小集合。

表 2-2 指向 (实现必备的) 静态表的索引

线型	→ 线型表 (最小长度 4)
记号类型	→ 记号类型表 (最小长度 5)
正文字体和精度	→ 正文字体和精度表 (最小要求取决于级)
填充区域式样索引	→ 阴影线表 (最小长度 3)

可调整的工作站表包括:

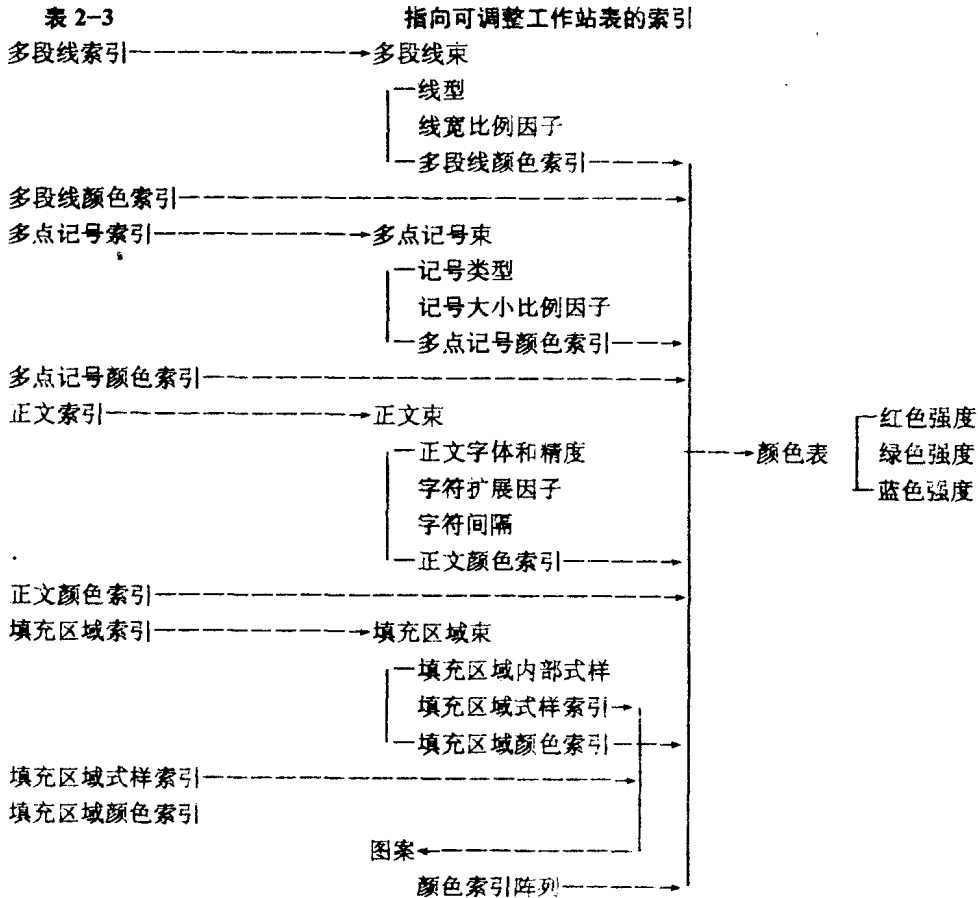
- 多线段束表;
- 多点记号束表;
- 正文束表;
- 填充区域束表;
- 图案表;
- 颜色表。

表 2-3 列出指向这些表的全部索引和在各自的束表中可被设置的全部外形。设置索引可发生在多层。例如, 填充区域索引指向包含填充区域内部式样索引的束表, 而填充区域内部式样索引又指向图案阵列, 而图案阵列里又包含指向颜色表的颜色索引。

工作站表的主要目的是利用每个工作站特有的能力, 在不同类型的工作站上显示图形。GKS 允许在任何时刻重新设置工作站表, 这增加了强有力的动态功能。例如, 光栅显示运用这种技术可以实现颜色表的动态能力。

属性可以是静态的 (static) 和动态的 (dynamic)。在生成原语时, 静态属性的值与原语结合在一起。在该原语使用的整个时间里其值保持不变。属性本身可以重新定义, 这样静

态属性将使随后产生的原语按新的属性值生成。如果动态属性发生变化，那么这种变化将影响先前和随后生成的原语。由于先前生成的原语属性的改变需要图形信息的存贮，所以仅能对图段里的原语起这种作用。



工作站表的动态能力可应用在任何工作站上。但是，由于工作站的能力不同，它们对重新设置工作站表的反应速度也不一样，有的可以立即作出反应，有的需一段延迟，以达到理想的显示。

索引与动态属性有联系，而索引本身不一定是动态属性。颜色是典型的动态属性。例如：

```

SET POLYLINE COLOUR-INDEX (1)
SET COLOUR REPRESENTATION (WKID, 1,1.0,0.0,0.0)
POLYLINE (N,POINT1)
SET COLOUR REPRESENTATION (WKID, 1,0.0,1.0,0.0)
POLYLINE (N,POINT2)
  
```

按着这个序列的定义，首先生成一条红色的多段线，重新设置颜色的表示之后，生成第二条多段线就是绿色的。在这里颜色索引本身是静态的，而颜色是动态的。

属性可能是全局的 (global)，或者是依赖于工作站的 (workstation-dependent)。如果是全局属性，其值对所有的工作站都有效，如果是依赖于工作站的属性，每个工作站都需要