

**МАШИННАЯ
ГРАФИКА
И ЕЁ
ПРИМЕНЕНИЕ**

Академия наук СССР Сибирское отделение
Вычислительный центр

МАШИННАЯ ГРАФИКА И ЕЕ ПРИМЕНЕНИЕ

Сборник научных трудов

Под редакцией А.М. Мацокина



Новосибирск 1979

СОДЕРЖАНИЕ

	Стр.
ПРЕДИСЛОВИЕ	4
ДВОРЖЕЦ В.И. Принципы построения и реализации модуль- ной графической системы СИГАМ	5
ДЕБЕЛОВ В.А. Система управления графическим выво- дом	41
ДЕБЕЛОВ В.А. Диалоговый графический канал	66
ДВОРЖЕЦ В.И., ТОРШИН В.И., УПОЛЫНИКОВ С.А. СМОГ для ЭВМ ЕС и АСВТ	91
ТОРШИН В.И., ТРУФАНОВ О.Д. СМОД для ЭВМ М-4030	98
ТОРШИН В.И., ПУЗАТКИН В.С., НАСИБУЛЛИН М.Х. СМОГ - МАГИСТРАЛЬ	102
ВИШНЕВСКИЙ Е.В., ДВОРЖЕЦ В.И., НИККЕЛЬ И.Г. Растро- вый графический канал в системе СМОГ- БЭСМ-4М	107
ВАСИЛЬЕВА Л.Ф. Операционные средства, обеспечиваю- щие работу ЭВМ БЭСМ-6 с унифицированной магистральной системой обмена (УМСО)	116
КУЗНЕЦОВ С.Б., УСОВ В.Н. Объемное изображение по- верхностей вращения	125
РИВИН Г.С., КУЛИКОВ А.И. Адаптация СМОГ к задачам вывода метеорологической информации	132

ПРЕДИСЛОВИЕ

Большая часть работ, представленных в настоящем сборнике, посвящена проблемам разработки систем машинной графики и их адаптации к различным машинно-приборным конфигурациям. В первой статье предлагается вниманию читателя новая графическая система СИГАМ, построенная на модульных принципах и обладающая развитой системой графических архивов и библиотек. Во второй работе описывается система управления графическим выводом, осуществляющая непосредственный вывод на графические устройства информации, подготовленной графическими системами. Разработка диалогового графического канала, обладающего тесной связью с первыми двумя работами, посвящена третья статья сборника.

Широкое использование системы математического обеспечения графических устройств (СМОГ), разработанной в Вычислительном центре СО АН СССР, вызвало необходимость в создании ее версий для различных ЭВМ. Следующие пять статей сборника посвящены вопросам адаптации системы СМОГ на ЕС-ЭВМ, М-4030 и БЭСМ-4М.

В последних двух работах сборника описывается алгоритм изображения поверхностей вращения и методы использования системы СМОГ для вывода метеорологической информации.

А.М.Мацокин

В.И.Дворжец

ПРИНЦИПЫ ПОСТРОЕНИЯ И РЕАЛИЗАЦИИ
МОДУЛЬНОЙ ГРАФИЧЕСКОЙ СИСТЕМЫ СИГАМ

В проектировании, конструировании, в научных исследованиях широко используется графическая форма представления информации. Большое распространение получили устройства графического отображения информации из ЭВМ и соответствующие системы математического обеспечения [1, 2, 8, 9]. Увеличивается также число диалоговых графических устройств (дисплеев), для которых разработаны и разрабатываются специальные диалоговые системы [3, 7, 8].

Важное значение приобретают задачи хранения и обработки графической информации в ЭВМ [12, 13]. Весьма часто возникают потребности в расширениях и модификациях программных систем машинной графики, а также в генерации некоторых специальных версий таких систем. Во многих случаях это позволяет существенно повысить скорость обработки и уменьшить объем занимаемой памяти.

Для решения этих задач, на базе проведенного исследования графических систем и алгоритмов, была разработана новая графическая система СИГАМ (система графических архивов и модулей). Основные принципы СИГАМ приведены в [4]. В настоящей статье дается развернутое изложение принципов построения и некоторых методов реализации системы.

I. Цель создания

Одна из первых отечественных графических систем общего назначения – система СМОГ [10, II]. Разработки различных версий, модификаций и расширений [3, 6] СМОГ дали весьма ценный опыт. Был также проведен анализ работ по машинной графике в СССР и за рубежом. Стало ясно, что путем простого расширения СМОГ, во-первых, всех новых задач не решить, и во-вторых, система при расширениях теряет свои основные качества – простоту и технологичность.

Это происходило в основном из-за недостаточно модульного характера СМОГ, а также из-за того, что некоторые решения, принятые на начальном этапе работ (в 1969 – 71 гг.), оказались неприменимыми в новых ситуациях. Это касалось как внешнего оформления некоторых процедур первого уровня [10] (таких как КЛЮЧ, КАНАЛ, ЛИСТ, СИМ), так и принятых соглашений о связях между процедурами и частями системы. Кроме того, в СМОГ практически отсутствуют архивные и диалоговые средства, средства обработки рисунков и работы с трехмерными объектами. Необходимо было также расширить библиотеку символов (сделав ее сменяемой) и буферную часть (добавить диалоговые каналы и усовершенствовать каналы графического вывода). Решить все эти задачи, оставаясь в рамках СМОГ, не представлялось возможным. В то же время, в связи с довольно широким распространением СМОГ и наличием в этой системе достаточно удачного второго уровня [II], система, призванная пройти ей на смену, должна обеспечить преемственность.

Итак, новая система должна предоставить пользователю диалоговые, архивные средства, средства построения и обработки графических объектов, средства генерации и модификации версий самой системы. Она должна быть существенно модульной, а также обеспечить преемственность по отношению к СМОГ, причем по второму уровню должна быть обеспечена совместимость со СМОГ.

2. Возможности

Система СИГАМ оперирует с объектами трех типов: графический объект общего типа (графоб), рисунок и символ.

Графобы строятся из простых графобов. Например, точка определяется как совокупность двух или трех чисел (координат), ребро – как совокупность двух точек, ломаная – как набор ребер и т.д. Общий тип графоба – структура, которую можно образовать путем объединения, пересечения (или иным образом) других графобов.

Рисунок строится как последовательность отрезков (приращений). Его можно получить и путем специальных преобразований графобов. Для хранения графобов и рисунков можно использовать оперативный (ОГА) или внешний (ВГА) графический архив.

Символы – это стандартные (наиболее часто используемые) рисунки. Они отличаются от рисунков способами кодирования и употребления, а также тем, что хранятся в специальных библиотеках символов. Символ можно создать только путем выполнения преобразования рисунок → символ.

В СИГАМ имеются средства создания и уничтожения архивов и библиотек символов. Графобы и рисунки, хранящиеся в архивах, могут редактироваться и подвергаться преобразованиям. Рисунки и символы могут выводиться на графические устройства. Имеются также средства прямого вывода (без записи в архив).

Для удобства обращения к графобам, рисункам, архивам и библиотекам символов им могут быть присвоены имена. Обращение к символам осуществляется по их номерам в библиотеке. В одной программе может использоваться несколько архивов и библиотек символов.

Средства построения рисунков включают средства СМОГ плюс некоторые дополнительные процедуры (в частности, рисование пунктиров).

В системе реализована параллельная выдача в каналы, совмещенная с построением рисунков, что позволяет строить одновременно несколько рисунков и вести вывод по нескольким каналам. Так же как и СМОГ, СИГАМ не ориентируется на конкретную ЭВМ или систему программирования. Система может быть использована в задачах проектирования, конструирования, в научных иссле-

дований и т.д.

СИГАМ включает развитой сервисный программный аппарат, который обеспечивает удобства отладки, надежный контроль ошибок и может выдавать пользователю информацию о состоянии и работе системы.

Модульный характер СИГАМ позволил создать развитую систему генерации, которая может из имеющегося набора модулей сконструировать специализированную систему, выполняющую нужные данному пользователю (или группе пользователей) работы с большей эффективностью.

3. Структура и состав

СИГАМ состоит из набора специальных комплектов, базового монитора и набора базовых комплектов. Любой комплект ориентирован на решение специальной или базовой задачи и состоит из модулей. Каждый модуль реализует одну или несколько базовых либо специальных функций.

При генерации версии системы задается необходимый набор базовых и специальных комплектов. Пользователь может вызвать и использовать отдельно любую часть (некоторый набор комплектов плюс базовый монитор) сгенерированной версии. Базовый монитор управляет размещением в оперативной памяти модулей базовых комплектов. Размещением специальных комплектов занимается используемая система программирования либо сам пользователь.

В настоящее время на ЭВМ БЭСМ-6 реализована первая версия системы, состоящая из базового монитора, шести специальных, восьми базовых комплектов и набора вспомогательных программ, включающих генератор системы и тестовые задачи.

Общая структура системы показана на рис. I.

Специальные комплекты (кроме заново написанного лифтового комплекта) с небольшими доработками взяты из второго уровня СМОГ [II] .

Базовые комплекты реализуют основные (базовые) возможности системы. Модули базовых комплектов написаны на автокоде и оттранслированы в машинные команды в виде перемещаемых блоков,

длина которых выбрана кратной сектору (256 ячеек). Базовый монитор также написан на автокоде, оттранслирован и оформлен в виде СП (по правилам библиотеки СП БЭСМ-6).

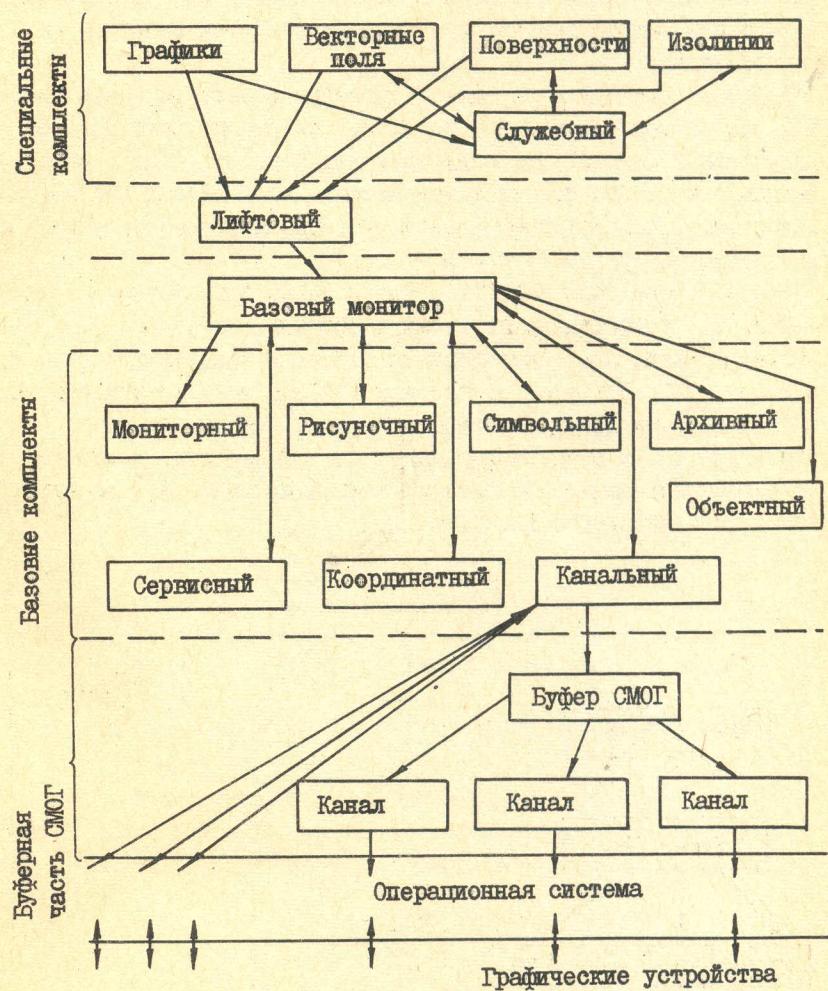


Рис. I. Общая структура системы СИГАМ

Связь программ пользователя и специальных программ системы, написанных на языке программирования типа АЛГОЛ или Фортран (или другом языке), с базовой частью системы осуществляется с помощью "лицевого" комплекта процедур (подпрограмм). Такой комплект может быть выполнен в любой системе программирования, содержащей встроенный автокод (или допускающей работу в машинных кодах).

Лицевые процедуры передают параметры в базовую часть системы, для размещения которой используется специальная область оперативной памяти. Эта область может располагаться в собственном или глобально описанном массиве либо выделяться из общей памяти, используемой данной системой программирования, каким-нибудь другим способом.

Связь базовой части СИГАМ с ОС (и далее с графическими устройствами) осуществляется двумя способами. Во-первых, сохраняется возможность использования буфера и канальных программ системы СМОГ. При этом обеспечиваются все возможности работы с устройствами графического отображения, реализованные в СМОГ [I0, II] и в ее расширениях [6]. Во-вторых, предусматривается реализация собственных канальных модулей в рамках канального комплекта (рис. 2).

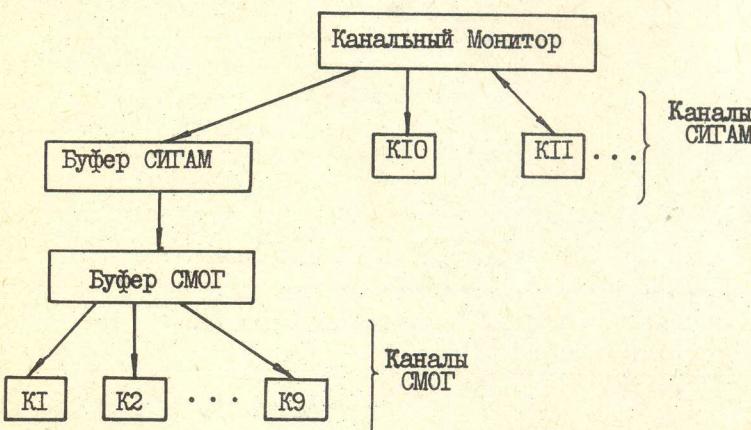


Рис. 2. Структура канального комплекта

4. Графические объекты

Определение: графическим объектом называется объект, построенный по правилам системы СИГАМ и хранящийся в оперативном (ОГА) или внешнем (ВГА) графическом архиве.

Графические объекты (графобы) строятся из простых графобов. Для построения простых графобов используются специальные процедуры. Первая версия системы обеспечивает построение и обработку плоских графобов.

В этой версии определяются четыре типа простых графобов:
точка – совокупность двух чисел (координат),
ребро – совокупность двух точек,
ломаная – набор ребер,

грань – часть плоскости, ограниченная замкнутой ломаной.

Общий тип графического объекта – структура. Ее можно образовать путем объединения, пересечения (или другим образом) из простых графических объектов и других структур.

Графобы связаны между собой в архиве иерархической системой ссылок. Кроме того, некоторые (или все) графобы могут быть снабжены именем.

Например, для построения квадрата нужно сначала задать четыре точки, затем на их базе четыре ребра, затем сам квадрат (грань). При этом, возможно, только точки и сам квадрат будут иметь имена. Для выполнения большинства операций СИГАМ графоб должен быть именован, однако это не обязательно для его составляющих.

Система ссылок обеспечивает доступ к любой части графоба (даже не именованной) после того, как получен доступ к какой-либо другой его части.

Всегда можно выяснить, например, каким графобам принадлежит анализируемое ребро или точка, и наоборот, какие ребра и точки составляют анализируемую грань.

Итак, графоб – это сильно структурированный объект, что обеспечивает легкость его конструирования и корректировки. Следует отметить, что способ хранения, скажем, некоторого рисунка, в виде графоба (особенно полной структуры – со всеми внутренними именами) требует весьма большого объема памяти. Например, хранение полной структуры квадрата требует почти в

пять раз больше памяти, чем занимают координаты его четырех точек.

Принятые в системе способы построения, хранения и обработки графобов не являются жестко фиксированными. На базе имеющихся процедур работы с архивами возможны построение и обработка других типов графобов (например, трехмерных). Для этого в базовую часть системы могут быть добавлены соответствующие модули. Эти же архивные процедуры могут применяться для построения неграфических объектов и связывания их с графобами в общую структуру, что позволяет хранить и обрабатывать любые дополнительные параметры графобов.

5. Рисунки и символы

Графоб – это один из наиболее общих типов объектов, необходимых в машинной графике. Однако хранение графобов требует большого объема памяти, поэтому их построение не всегда оправдано. В тех случаях, когда не требуется столь полная структурированность, можно применять другие, более специальные типы объектов – рисунки и символы.

Объект-рисунок или просто рисунок имеет представление, удобное прежде всего для вывода. Рисунок строится как последовательность приращений. Для построения рисунков в системе применяются как специальные процедуры, так и процедуры, аналогичные имеющимся в системе СМОГ (ТРА, ТРАД и т.д.) [10]. Рисунки могут храниться в тех же оперативных и внешних графических архивах, что и графобы. Кроме того, рисунок может сразу же выводиться (в процессе построения) на графическое устройство без записи в архив. Возможно также совмещение записи с выводом.

Рисунок можно получить также путем выполнения специальной операции преобразования графоб \rightarrow рисунок.

В архиве рисунок хранится как набор блоков приращений. Из рисунков, так же как и из графобов, можно строить структуры, однако минимальным элементом такой структуры будет уже не точка, а целый рисунок. Рисунку можно вернуть полную структуру, выполнив обратное преобразование рисунок \rightarrow графоб.

Плотность кодирования информации в рисунке весьма высока (в одной ячейке можно разместить несколько приращений). Это дает существенный выигрыш в памяти не только по сравнению с графобами, но и по сравнению с тривиальным способом хранения рисунков в виде координат точек (по одной координате в ячейке). В первом случае выигрыш в памяти может достигать нескольких десятков и даже сотен крат (для больших сложных графобов), а во втором случае от двух до двенадцати крат.

Таким образом, большие массивы графической информации или массивы, не предназначенные для частой корректировки, удобнее хранить в виде рисунков. Тем не менее, если составляющие рисунки хранящихся таким образом рисуночных структур не очень велики, то их можно довольно просто корректировать по частям, используя преобразования подрисунков в графоб и обратно.

Символы – это стандартные (наиболее часто используемые) рисунки. Они отличаются от последних способом кодирования и употребления, а также тем, что хранятся в специальных библиотеках символов. Символ можно создать только путем выполнения преобразования рисунок → символ. Обрабатывать символ нельзя, его можно только целиком заменить. Идентификация символа осуществляется по его номеру в каталоге библиотеки.

Символьные библиотеки используются при отрисовке текстов, часто встречающихся знаков и т.д. В СИГАМ имеются специальные процедуры отрисовки символов, текстов и чисел, некоторые из которых являются дальнейшим развитием соответствующих процедур СМОГ.

6. Архивы

В СИГАМ имеются следующие типы архивов:

- оперативный графический архив (ОГА),
- внешний графический архив (НГА),
- библиотека символов (БС).

Архив первого типа (ОГА) создается по запросу программы пользователя и существует в единственном экземпляре и лишь до конца работы с системой или до указания о закрытии ОГА.

Весь ОГА размещается в оперативной памяти. ВГА предназначен для длительного (на МП или МД) либо временного (МБ, МД) хранения графиков и рисунков. В одной программе можно работать с несколькими ВГА.

В библиотеках символов в плотно упакованном виде хранятся стандартные рисунки (символы). Библиотеки символов постоянно находятся на МП или МД, и перед использованием могут вызываться на МБ. В одной программе разрешается использовать несколько БС, однако в каждый момент времени в активном состоянии может находиться только одна БС. Каталог активной БС хранится в оперативной памяти.

Архивы СИГАМ отличаются также по структуре. Архив ОГА не имеет каталога. Поиск по имени объекта осуществляется с помощью специальных именных ссылок, образующих двоичное дерево имен. В ОГА есть также список свободной памяти. Каждый объект в ОГА имеет паспорт, который связан другой системой ссылок с остальными элементами объекта. Таковыми могут являться имя с именными ссылками и список связей с другими объектами или информационными блоками. Например, если объект имеет тип "точка" или это рисунок самого нижнего уровня (не структура), то ссылка вниз в списке связей этого объекта указывает соответственно на блок из двух координат либо на блок кодов приращений. В большинстве остальных случаев ссылки списка связей указывают на поспорт внешних (ссылка вверх) или внутренних (ссылка вниз) объектов.

ВГА имеет каталог, в котором содержатся имена хранимых объектов. Структура объекта в ВГА почти такая же, как в ОГА. Отличаются только способы представления ссылок и паспортов объектов. Возможен обмен объектами между ВГА и ОГА. Перед записью объекта из ОГА в ВГА может производится специальная операция "освобождения" объекта. При выполнении этой операции уничтожаются все "внутренние" имена (имена подобъектов), именные ссылки и все ссылки вверх. При выполнении считывания из ВГА в ОГА восстанавливаются только ссылки вверх; заголовки внутренних объектов (имена и именные ссылки) не восстанавливаются. Пользователь может создать их заново (причем не обязательно для всех подобъектов) при помощи специальных операций именования.

Библиотека символов состоит из каталога и набора блоков приращений, кодирующих символы. Кодировки рисунков и символов совпадают.

7. Лифтовые и специальные графические комплекты

Базовая часть системы для каждого типа ЭВМ должна быть доступна из большинства (или всех) систем программирования данной ЭВМ.

Для обращения к программам базового уровня в каждой системе программирования создается специальный лифтовый комплекс подпрограмм (процедур). Он должен содержать оформленные по правилам данной системы программирования подпрограммы (процедуры), совпадающие по названиям и списку параметров с соответствующими подпрограммами базовой части СИГАМ. Функция этих подпрограмм (процедур) состоит в передаче своих параметров в базовую часть СИГАМ для отработки и в приеме параметров - результатов из базовой части.

В многоязыковых системах программирования (например, в мониторной системе "Дубна") лифтовый комплекс достаточно реализовать только на одном из языков системы. Для версии СИГАМ на ЭВМ БЭСМ-6 такие комплексы созданы в системах программирования АЛГОЛ - БЭСМ, АЛЬФА-6 и в мониторной системе "Дубна".

Наличие лифтового комплекса в некоторой системе программирования позволяет создавать в ней специальные графические комплексы. Функции таких комплексов могут быть весьма разнообразны. Обычно они реализуют расчет и построение сложных рисунков или объектов какого-либо типа (например графиков, изолиний и т.д.).

8. Графические переменные

Для удобства обращения к графикам, рисункам, библиотекам символов и другим объектам системы им могут быть присвоены

имена. Возможны различные способы использования графических объектов в языках программирования. Иногда для этого реализуются специальные языки [12], часто применяются различные расширения стандартных языков [13]. При этом обычно вводятся новые типы переменных, новые описатели типов и т.д. Такой подход хотя и удобен для пользователя, но требует построения специальных фортрансляторов (или трансляторов) для каждой реализации системы.

Один из основных принципов СИГАМ – отказ от расширения синтаксиса того языка, которым пользуется программист, работающий с СИГАМ. При этом остаются возможности особой трактовки переменных стандартных типов, если эти переменные используются как параметры некоторых операторов СИГАМ.

Наиболее удобным для таких целей типом переменных является тип "строка". Однако не во всех языках программирования (и не во всех реализациях этих языков) имеются удобные средства работы со строками. В таких языках (реализациях) для хранения имён могут использоваться, например, переменные типа "целый", строковые значения которым присваиваются специальными кодовыми процедурами. Это возможно в любой системе программирования, содержащей встроенный автокод либо допускающей использование машинных команд.

9. Базовый уровень СИГАМ

Базовая часть системы является центральной и наиболее важной. Она должна обеспечить эффективную реализацию наиболее общих и часто используемых графических операций. Тем самым создается надежная база для построения специальных комплектов и предоставляются удобные средства работы с буферной частью системы. Базовая часть реализует большинство сервисных функций, выполняет контроль и обработку ошибок, а также осуществляет общее управление работой системы.

10. Работа с архивами

Для работы с оперативным и внешними архивами в СИГАМ имеются специальные процедуры. Перед началом работы с оперативным архивом необходимо выделить ему некоторую область оперативной памяти. Для этого используется процедура ОГА, параметры которой — адрес начала блока оперативной памяти и длина этого блока. Процедура может употребляться неоднократно, добавляя к ОГА новые блоки памяти. Аналогично, перед началом работы с некоторым ВГА, его необходимо "описать", употребив процедуру ВГА. Ее параметры — имя ВГА и его адрес во внешней памяти.

Далее необходимо "активировать" тот архив, с которым будет производиться работа. В любой момент времени в активном состоянии может находиться не более одного архива (ОГА либо один из ВГА). Архив активируется при выполнении процедур ГА с параметром "имя архива". Имя ОГА фиксировано и определяется конкретной реализацией системы. Вся дальнейшая работа до следующего обращения к процедуре ГА ведется с активным архивом.

В СИГАМ имеются также процедура "начализации" архива (НГА) и процедура закрытия архива (КГА). Параметр процедуры КГА указывает имя архива. Процедура НГА не имеет параметров. Она используется для освобождения памяти, занятой каталогами и служебной информацией. После закрытия архива допускается его повторное описание и использование.

Для случая ОГА процедура НГА освобождает все занятые блоки архива и возвращает ОГА в начальное состояние. Последующая работа с графическим архивом (ГА) производится с помощью процедуры управления памятью, именных процедур и процедур управления ссылками. Эти процедуры используются, в основном, процедурами работы с рисунками и графиками, однако их можно использовать и для других целей, например, для размещения в ГА нестандартной информации и ее обработки.

Процедура управления памятью (ПАМА) имеет три параметра. Первый задает количество ячеек (КЯ) блока выделяемой, освобождаемой или добавляемой памяти, второй — адрес этого блока памяти (АБП), третий — режим (выделение, освобождение или добавка памяти к архиву).

Для ОГА смысл параметров очевиден. Для ВГА параметр "адрес" состоит из адреса зоны ВГА и адреса в зоне. Если задан режим добавления памяти к ВГА, то АБП означает адрес начальной зоны добавленного блока зон, а КЯ - количество добавляемых зон. Обычно параметр АБП используется как транзитный, т.е. является входным для именных процедур и процедур управления ссылками.

При выполнении процедуры ПАМА происходит перестройка списка свободной памяти активного архива. Если задан режим добавления, к началу списка свободной памяти добавляется новый блок. В режиме освобождения блок памяти из занятой части архива также добавляется к началу списка свободной памяти. В режиме выделения из первого блока списка свободной памяти, имеющего нужный (или больший) размер, выделяется необходимая часть. Остаток блока сохраняет свое место в списке свободной памяти. Заметим, что список свободной памяти в ВГА свой для каждой зоны архива.

Построение всех структур в архиве производится только через процедуру ПАМА. Вообще говоря, с ее помощью можно строить достаточно разнообразные типы структур, однако в архивах СИГАМ для построения графиков и рисунков используются стандартные структуры. На рис. 3 показана такая структура для ОГА.

Как видно из рисунка, стандартная структура объекта состоит из элементов трех типов: паспорта, заголовка и списка связей. Разбивка элементов по ячейкам зависит от реализации. На рис. 3 разбивка соответствует первой версии СИГАМ на БЭСМ-6. Ссылки влево и вправо в заголовке объекта указывают на соседние заголовки в дереве имен. Подробнее об этих ссылках будет сказано ниже.

Уточним, что поле "признаки" в паспорте объекта хранит информацию о типе объекта, его размерности, уровне в структуре и т.д.

Стандартная структура объекта в ВГА несколько сложнее. Объект может храниться в виде нескольких частей, которые размещаются в разных зонах ВГА. Объект может быть снабжен заголовком, который хранится в каталоге ВГА и состоит из имени и ссылки на первую часть объекта. Такие объекты будем называть каталогизированными, а имя в заголовке объекта "общим".

Объект может иметь также "личное имя", которое хранится в первой части объекта и может означать, например, имя, которое этот объект имел в ОГА до переписи в БГА. Личным именем могут обладать как каталогизированные, так и не каталогизированные объекты.

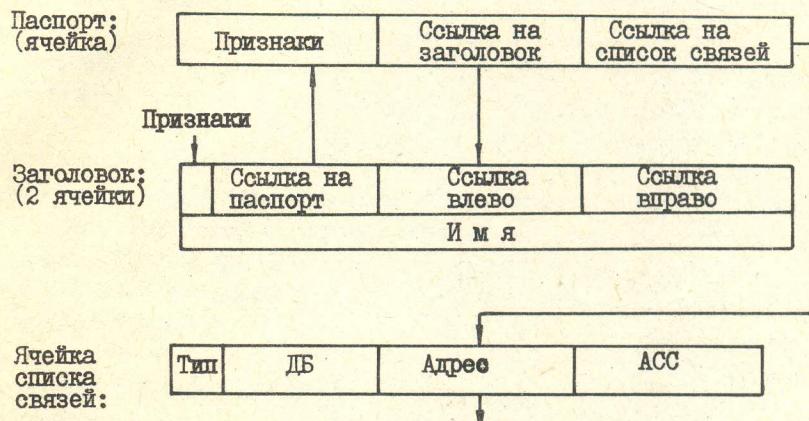


Рис. 3. Стандартная структура объекта в ОГА:
 Адрес – адрес паспорта объекта или блока информации, на который указывает ссылка;
 ACC – адрес следующей ячейки списка связей.
 Если нуль, то последняя ячейка списка связей;
 ДБ – длина информационного блока, на который указывает ссылка. Если ссылка указывает на паспорт объекта, то ДБ = 0;
 Тип=1 – ссылка вверх, Тип = 0 – ссылка вниз.

Каждая часть объекта в БГА состоит из паспорта, списка связей и, если необходимо, блоков информации (например, координат точки или кодов рисунка). Первая часть может также содержать личное имя объекта.

Ссылка объекта во внешнем архиве состоит из типа (вниз или вверх), адреса зоны и адреса в зоне (аналогично параметру АБП процедуры ПАМА), а также длины блока в случае, если она

указывает на блок. Если список связей или дополнительная (обычно графическая) информация не помещаются в первой части объекта, то они продолжаются во второй части и т.д.

Паспорта всех частей содержат:

- признаки объекта (те же, что и в ОГА);
- признак первой части;
- адрес следующей части (равен нулю для последней части);
- адрес первой ячейки списка связей части.

Паспорт первой части, кроме того, содержит адрес личного имени и адрес заголовка. Паспорт не первой части содержит вместо этого адрес первой части.

Заметим, что одним из основных преимуществ стандартной структуры в оперативном архиве является одинаковая доступность элементов, тогда как во внешнем архиве элементы могут быть расположены в разных частях. Это обстоятельство может несколько понизить скорость обработки. Весьма важно, что в ВГА сохраняется гибкая структура связей, что облегчает корректировку сложных объектов.

Для построения стандартных процедур в СИГАМ имеются именные процедуры и процедуры управления ссылками. Эти процедуры, как и процедура ПАМА, используется, в основном, как "внутренние" теми процедурами, которые работают со стандартными типами объектов - рисунками и граffобами. К именным процедурам относятся процедуры ИМЕН - параметры "Адрес паспорта" (АПАС) и "Имя", ПОИМ - параметры те же, и ИСИМ - параметр "Имя".

Процедура ИМЕН используется для именования объекта, паспорт которого расположен по адресу АПАС (трактовка понятия "адрес" здесь и далее такая же, как и для параметра АБП процедуры ПАМА).

Процедура ПОИМ осуществляет поиск имени в архиве и выдает, как результат, адрес паспорта искомого объекта. Процедура ИСИМ позволяет исключить из архива имя любого объекта. Именные элементы стандартных структур в ОГА и НГА отличаются весьма сильно. В ВГА имеется каталог, который является линейным массивом заголовков. При выполнении процедуры ИМЕН в каталоге создается новый заголовок, в который помещается имя и ссылка на паспорт объекта, а в паспорт первой части объекта заносится ссылка на заголовок и признак каталогизации. Проце-

дуре ИСИМ выполняет обратную операцию, а процедура ПОИМ, про-
сматривая каталог, находит и выдает в качестве результата
ссылку на первую часть объекта (адрес).

В ОГА имена расположены в архиве хаотически и связаны ме-
жду собой специальной системой именных ссылок таким образом,
что они образуют двоичное дерево. Способы построения и кор-
ректировки дерева основаны на просмотре кода имени слева на-
право. В процессе просмотра происходит движение по дереву.
При этом число пройденных узлов дерева будет не больше, чем
количество двоичных разрядов кода имени.

Длина кода имени определяется конкретной реализацией систе-
мы. Для первой версии СИГАМ на БЭСМ-6 она равна 48. Таким
образом, при поиске имени в дереве будет пройдено не больше
48 узлов. При вставке или удалении любого узла требуется кор-
ректировка только одного или двух узлов.

Отметим также одну особенность дерева имен ОГА. Если в де-
рево включать коды имен, представляющих из себя подряд иду-
щие числа, например, коды имен А1, А2, ..., АМ, то полученное
дерево будет выровненным. Такой случай использования имен до-
вольно часто встречается при именовании элементов графических
структур.

Обосновывая принятый способ построения дерева имен в ОГА,
отметим, что весьма важным фактором в графической, особенно
диалоговой, системе является большая частота перестройки.
Действительно, оперативный архив служит в основном для по-
строения и корректировки объектов. В данном случае увеличение
длины пути поиска (по сравнению с выровненными или подравнен-
ными деревьями) компенсируется уменьшением трудоемкости пере-
стройки.

Рассмотрим теперь процедуры управления иерархической систе-
мой ссылок архивов СИГАМ. К ним относятся процедуры УСТСС,
СНЯСС и ССП. Процедура УСТСС имеет параметры АПАС, АС и ТИП.
АПАС задает адрес паспорта объекта, из которого исходит ссыл-
ка, АС - адрес паспорта объекта или адрес блока информации,
на который указывает ссылка, а ТИП задает тип ссылки (вниз
или вверх по иерархии). Процедура СНЯСС с параметрами АПАС и
АС позволяет убрать из списка ссылок объекта, определяемого
АПАС, ссылку с адресной частью, равной АС. Процедура ССП с

с параметрами ПАПАС, АС, ТИШ, НС, ПРЗП позволяет считать или записать (в зависимости от значения ПРЗП) ссылку с номером НС, и, возможно, заданного ТИШа из списка ссылок объекта, определяемого АПАС. АС является входным параметром при считывании и выходным при записи ссылки.

Память для хранения ссылок заказывается этими процедурами через процедуру ПАМА. Принятая структура ссылок обеспечивает доступ к любой части объекта, если известен адрес паспорта любой другой части этого объекта. Действительно, с помощью процедуры ССЛ можно просмотреть список ссылок заданного подобъекта, найти в нем ссылку вверх на объект, затем в списке ссылок этого объекта найти ссылку вниз на нужный подобъект.

II. Построение графических объектов

Построение графиков осуществляется процедурами ТОЧКА, РЕБРО, ЛОМАН, ГРАНЬ и СТРУК. Опишем параметры этих процедур и поясним на примере ОГА способы хранения и корректировки графиков.

Простейший график "точка" генерируется при обращении к процедуре ТОЧКА, которая имеет три параметра - имя точки и координаты X и Y. Процедура ТОЧКА строит в архиве структуру, показанную на рис. 4.

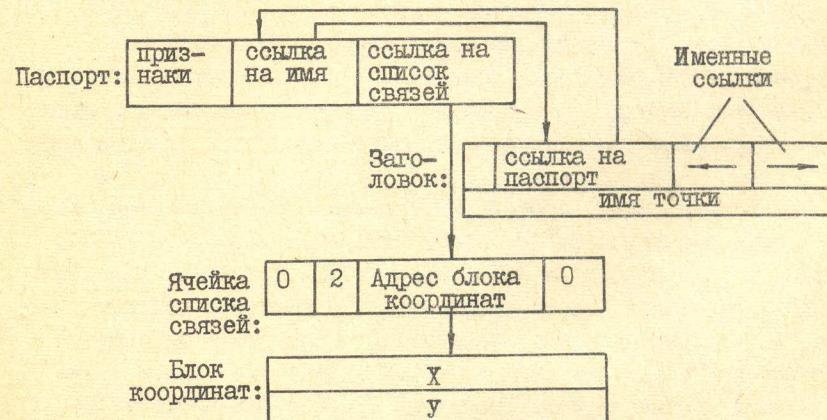


Рис. 4. Точка в ОГА

При повторном обращении к процедуре ТОЧКА с тем же именем происходит только установка новых значений координат X и Y.

Процедура РЕБРО имеет три параметра - имя ребра, имя первой точки ребра и имя второй точки ребра. После выполнения процедуры структура в архиве будет иметь вид, показанный на рис. 5.

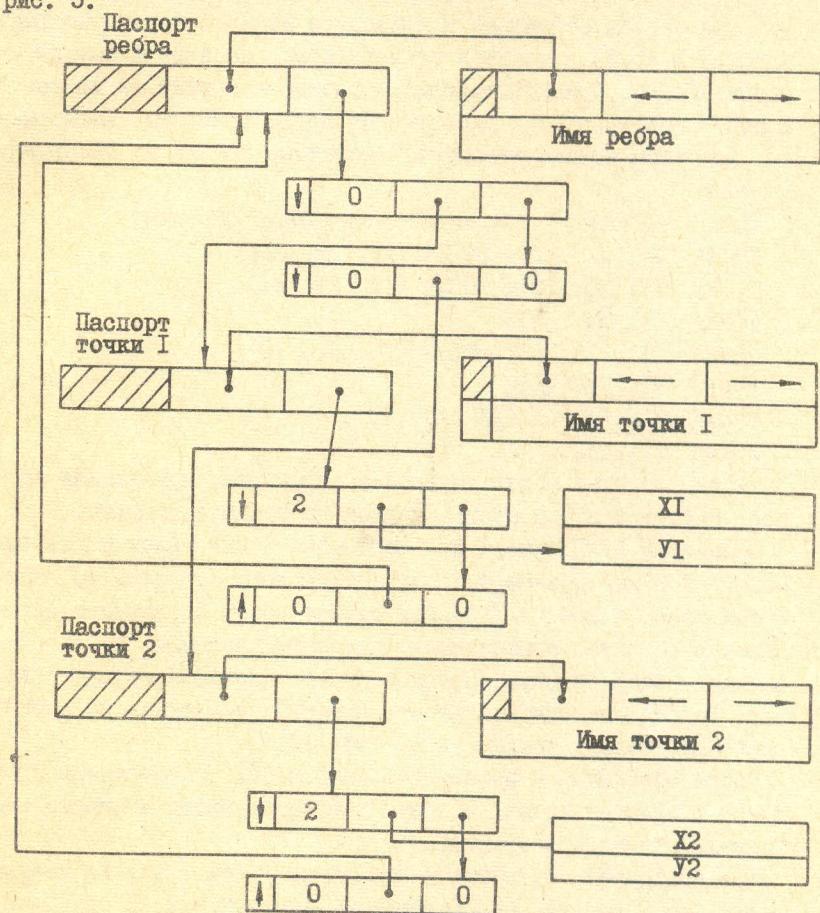


Рис. 5. Ребро в ОГА

При повторном обращении к процедуре РЕБРО с тем же именем ребра, но с другими именами точек происходит перестройка системы ссылок. Между паспортом ребра и паспортами этих точек устанавливаются взаимные ссылки, а первоначальные связи уничтожаются.

Процедура ЛОМАН имеет два параметра -- имя ломаной и имя ребра. При первом обращении к процедуре ЛОМАН с некоторым именем ломаной в архив заносится ее заголовок, паспорт и ссылка на первое ребро. При последующих обращениях с этим же именем ломаной к списку ссылок добавляются новые элементы. Естественно, при этом устанавливаются и обратные ссылки из ребер на ломаную.

Пусть, например, выполнялись следующие операторы:

ТОЧКА (T1, X1, Y1); ТОЧКА (T2, X2, Y2);
ТОЧКА (T3, X3, Y3); ТОЧКА (T4, X4, Y4);
ТОЧКА (T5, X5, Y5);
РЕБРО (P1, T1, T2); РЕБРО (P2, T2, T3);
РЕБРО (P3, T4, T5);
ЛОМАН (Л1, P1); ЛОМАН (Л1; P2);
ЛОМАН (Л1, P3).

Тогда в архиве будет построена структура, показанная на рис. 6 (заголовки и ячейки ссылок на рисунке опущены).

Процедура ГРАНЬ имеет два параметра -- имя грани и имя ломаной. Ломаная должна быть замкнутой либо состоять из замкнутых наборов ребер. В первом случае грань определяет односвязную область, ограниченную замкнутой ломаной, во втором случае -- многосвязную. Порядок ребер в ломаной не существует. Ребра компонент связности ломаной не должны пересекаться во внутренних точках ребер (рис. 7).

После обращения к процедуре ГРАНЬ в ОГА записывается заголовок и паспорт грани, а также устанавливаются взаимные ссылки между ломаной и гранью.

При повторном обращении к процедуре ГРАНЬ с тем же именем грани, но с другим именем ломаной взаимные ссылки связывают грань с новой ломаной, а старые ссылки уничтожаются. Если ломаная, задающая грань, не удовлетворяет приведенным выше условиям, возникает ошибочная ситуация.

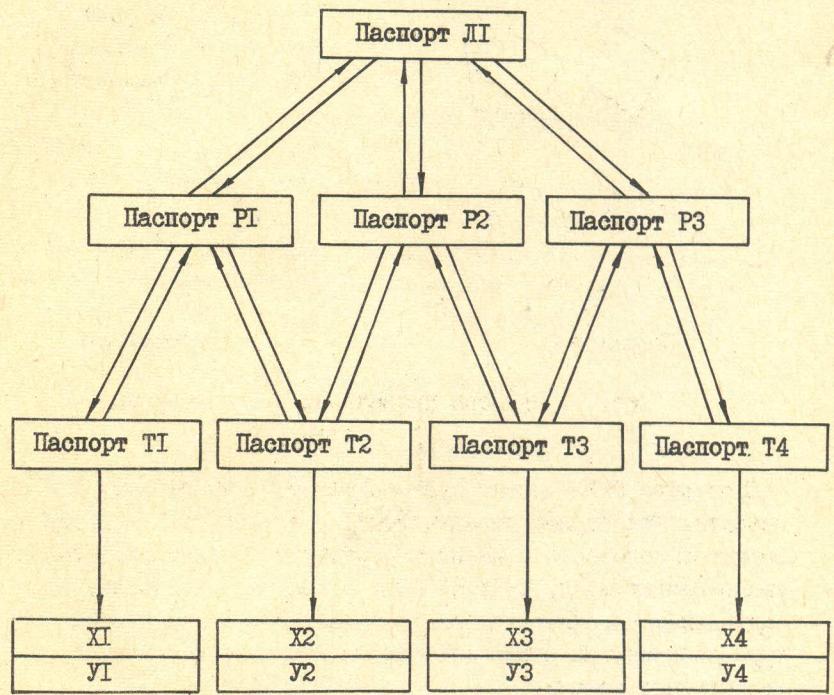


Рис. 6. Ломаная в ОГА (заголовки и ячейки ссылок опущены)

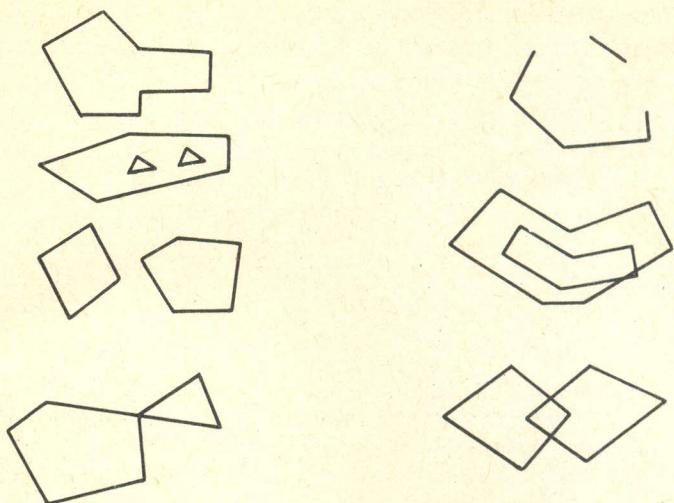


Рис. 7. Примеры правильных и неправильных граней

Процедура СТРУК имеет два параметра – имя структуры и имя объекта. При первом обращении к процедуре СТРУК в архив записываются заголовок и паспорт структуры. Устанавливаются взаимные ссылки между структурой и объектом. При последующих обращениях к процедуре с тем же именем структуры к этой структуре добавляются новые объекты, а в архиве появляются соответствующие ссылки.

Процедуры ТОЧКА, РЕБРО, ЛОМАН, ГРАНЬ и СТРУК обеспечивают построение плоских графобов. Однако можно отметить, что достаточно изменить только процедуру ТОЧКА, добавив к ней один параметр (координату z), чтобы те же процедуры строили графы в трехмерном пространстве. Нужно еще ввести ограничение на ломаную, задающую грань: все ее точки должны лежать в одной плоскости. Конечно, для работы в трехмерном пространстве нужно также добавить процедуру построения поверхностей из граней (аналогично построению ломанной из ребер) и процедуру

построения тела, как части плоскости, ограниченной замкнутой поверхностью (аналогично построению грани как части плоскости, ограниченной замкнутой ломаной).

12. Операции анализа и преобразований графических объектов

Для выполнения операций над графобами в СИГАМ имеются процедуры анализа и преобразований.

В системе шесть процедур анализа графобов. Все они являются процедурами-функциями целого типа. Процедура ТИП с параметром ИМЯ выдает тип объекта с заданным именем. Тип объекта-рисунка равен нулю, точки – единице и т.д.

Процедуры КС и КВ с параметром ИМЯ выдают, соответственно, количество составляющих (подобъектов) объекта с заданным именем, т.е. количество ссылок вниз, или количество включающих (надобъектов), т.е. количество ссылок вверх.

Аналогично, процедуры ИМЯС и ИМЯВ с параметрами ИМЯ и Н (номер) выдают имя составляющего или включающего объекта с номером Н для объекта, заданного именем ИМЯ.

Процедура КТ с параметрами ИМЯТ, Х и У выдает значения координат точки с именем ИМЯТ в параметры результаты Х и У.

Процедур преобразования также шесть.

Для обмена графобами между разными архивами в системе имеется процедура ПТ (преобразование графоба в графоб). Ее параметры – имя архива ИМЯРХ, имя преобразуемого графоба ИМЯПГ, хранящегося в ИМЯРХ и ИМЯГП – имя графоба-результата, который будет построен в активном архиве. Все составляющие вновь построенного графоба будут без имен.

При выполнении преобразования в однотипных архивах (ОГА → ОГА, ВГА → ВГА) происходит простое дублирование (с уничтожением имен составляющих). При записи из ОГА в ВГА сначала записывается паспорт, затем графоб каталогизируется, и начинают записываться его ссылки и составляющие сверху вниз по уровням иерархии вплоть до информационных блоков (координат точек и блоков кодов приращений).

При считывании из ВГА в ОГА структура строится также свер-

ху вниз. Как только в архив записывается паспорт некоторой составляющей, сразу же устанавливаются взаимные ссылки между ним и паспортом включающего графоба.

Процедуры преобразования используют специальный массив для хранения пути в иерархии преобразуемого графоба. В системе СИГАМ принято соглашение о том, что глубина вложенности (длина пути) не превосходит 64, что ограничивает размер этого массива 64 ячейками.

Процедуры ПРГ и ПРР с параметрами ИМЯРХ, ИМЯБ и ИМЯР (в обращении к процедуре ПРР задается также точность кодирования рисунка), выполняют соответственно преобразования графоб \rightarrow рисунок и рисунок \rightarrow графоб.

При преобразовании рисунка в графоб все приращения превращаются в ребра, появляются также составляющие их точки. Если рисунок не структурирован, то результатом преобразования будет графоб типа ЛОМАН, иначе – структура. Если преобразованию подвергается графоб типа структура, у которого некоторые составляющие имеют тип рисунок, они подвергаются преобразованию, а остальные составляющие не преобразуются. Преобразование приводит к простому дублированию графоба, если ни одна из его составляющих всех уровней вложенности не является рисунком.

Преобразование графоба в рисунок выполняется следующим образом. Сначала все содержащиеся в нем ломаные преобразуются в рисунки. При этом ребра переходят в приращения. Затем все оставшиеся ребра (не входящие в ломаные) преобразуются в рисунки (состоящие из одного приращения). После этого уничтожаются все оставшиеся одиночные точки и все грани (ссылки на грани будут указывать на рисунок, в который превратилась составляющая ломаная).

При выполнении преобразований ПРГ и ПРР исходный графоб (рисунок) остается в архиве ИМЯРХ неизменным, а преобразованный помещается в активный архив и снабжается заданным именем. У этого нового графоба (рисунка) составляющие всех уровней будут безымянными.

Процедура ИСК позволяет исключить графоб из активного архива. При этом в архиве уничтожаются все ссылки на этот графоб. Однако все составляющие (всех уровней) исключаемого графоба,

на которые есть ссылки из других графобов, остаются в архиве.

Процедура ИМЕНС с параметрами ИМЯ, Н (номер составляющей) и ИМЯС (имя составляющей) позволяет снабдить составляющую Н графоба ИМЯ именем ИМЯС. Наличие процедуры ИМЕНС избавляет от необходимости хранения имен всех составляющих.

Переименование уже именованных графобов выполняет процедура НИМЯ.

Операции анализа и преобразований предоставляют пользователю необходимые средства редактирования графических объектов (как графобов, так и рисунков). Эти средства применимы как в диалоговой графике, так и при работе в пакетном режиме. Используя процедуры построения, анализа и преобразований можно за несколько сеансов (этапов) сконструировать практически любой требуемый графический объект.

I3. Работа с рисунками

Операции над рисунками делятся на четыре категории. Это операции кодирования, рисования, преобразования рисунка в графоб (рассмотренная в п. 3.2) и преобразования рисунка в символ.

Операции кодирования реализуются процедурами НРИС, КРИС и СОРИС. Кодирование любого рисунка производится между обращениями к НРИС и КРИС. Параметры НРИС - ИМЯ (рисунка) и ЧШКВЕТ (число шагов кодирования в единице ТРА). КРИС имеет только первый из этих параметров.

При кодировании поступающая в рисующую процедуру ТРА информация в виде координат и приращений перехватывается и с помощью обращения к служебной процедуре СОРИС передается в тот модуль системы, который осуществляет кодирование. Закодировать таким образом можно только неструктурированный (простой) рисунок. Построить структуру можно, применив процедуру СТРУК к простым рисункам или структурам рисунков.

Построение нескольких рисунков может вестись параллельно. В этом случае при обработке очередного приращения оно записывается во все кодируемые рисунки (но с разной точностью, определяемой для каждого рисунка своим параметром ЧШКВЕТ в обращении к НРИС с именем этого рисунка). Порядок обращений к

НРИС и КРИС для разных рисунков не ограничивается. Число одновременно кодируемых рисунков не должно превышать 64.

При построении рисунок может выводиться и (или) записываться в текущий архив (как простой рисунок). В ОГА все ссылки вниз (кроме первой) простого рисунка указывают непосредственно на блоки приращений. Приращения записываются в специальной кодировке, длина кода которой изменяется от 8 до 56 бит (и всегда кратна 4).

Кодировка, принятая в СИГАМ, обладает на 20 - 40% лучшими показателями, чем кодировка [5] при кодировании рисунков (символов) с небольшими значениями приращений (меньше 32 шагов). Большие приращения (больше 2000 шагов) кодируются с равными затратами памяти. На 16 - 40% больше памяти чем в [5] тратится на кодирование средних движений (от 32 до 2000 шагов) по направлениям, близким к оси X или Y, но не совпадающим с ними. По остальным направлениям для средних движений показатели одинаковы.

В закодированном рисунке хранится также информация о его размерах, т.е. величины максимальных отклонений от начальной точки рисунка и отклонений конечной точки от начальной (в шагах кодирования), а также величина ЧИКВЕТ. Эта информация оформлена в виде первого блока рисунка и не нее указывает первая ссылка вниз этого рисунка.

В системе СИГАМ имеются специальные модули, реализующие кодирование и декодирование рисунков в принятой кодировке. Для рисования закодированных рисунков используются процедуры РИС, РИСР, РИСД и РИСП.

Процедуры РИС, РИСД и РИСП имеют один параметр - имя рисунка и позволяют нарисовать его соответственно в текущих листовой, декартовой и полярной системах координат. Процедура РИСР имеет еще два параметра - размеры RX и PY и рисует заданный рисунок, вписывая его в заданные размеры, используя размерную информацию. При отрисовке рисуночных структур процедура РИСР сначала определяет максимальные отклонения, проанализировав размерную информацию составляющих простых рисунков, а затем, выбрав соответствующий масштаб, выполняет рисование.

Во время кодирования рисунков, возможно одновременное де-

кодирование других рисунков, что позволяет включать одни рисунки в другие. Например, для преобразования рисуночной структуры РС в простой рисунок Р достаточно выполнить три оператора:

НРИС (Р, Т);
РИС (РС);
КРИС (Р).

Все составляющие рисунка РС будут декодированы (соответственно хранящимся в них точностям кодирования) и закодированы в виде простого рисунка Р с точностью, заданной параметром Т. Кодировка рисунка может также содержать дополнительную информацию о подъеме и опускании инструмента (пера), смене инструмента и т.д.

14. Операции над символами и библиотеками символов

Кодировка символов такая же, как и рисунков. Отличия касаются способов хранения и рисования символов. Символ хранится не в ОГА или ВГА, а в специальном архиве, который называется "Библиотека Символов" (БС). Программа может использовать несколько БС, которые располагаются на внешних носителях. В активном состоянии (т.е. доступном для использования) может находиться только одна БС. БС состоит из сектора каталога и секторов кодов символов.

Длина кодировки каждого символа ограничена длиной сектора. Количество символов в БС определяется длиной элемента каталога. В каталоге на каждый символ хранится номер сектора БС, где расположена его кодировка, адрес начала массива кодировки в этом секторе и длина массива кодировки. Каталог вмещает такую информацию на 510 символов. Кроме того, в каталоге хранится имя БС и размеры "символов площадки". Эти размеры используются при рисовании символов для масштабирования.

Кодировка каждого символа представляет из себя непрерывный массив кодов, заканчивающийся специальным кодом "конец рисунка". В кодировке символа может содержаться такая же дополнительная информация, как в кодировке рисунка. Идентифи-

кация символа в БС осуществляется по его номеру в каталоге этой БС.

Перед использованием БС ее необходимо описать. Это выполняется обращением к процедуре ОБС, параметры которой - имя БС и адрес БС во внешней памяти.

После описания нужных БС одну из них можно активировать обращением к процедуре БС с параметром ИМЯБС. Все остальные процедуры работают с активной БС. Процедура НБС "начализует" активную БС. При этом очищается каталог и устанавливаются размеры символьной площадки. После начализации БС готова к записи в нее символов.

Запись символа в БС осуществляется процедурой ПРС с параметрами ИМЯР (имя рисунка) и НС (номер символа). Она преобразует рисунок ИМЯР в символ НС. При этом игнорируется размерная информация, а коды приращений считаются из блоков рисунка и упаковываются в единый массив кодов символа.

Чтобы масштаб рисунка, кодируемого для последующего перевода в символ, совпал с масштабом символа, рисуемого в символьной площадке стандартного для данной БС размера, необходимо параметр ЧШКВЕТ в обращении к НРИС задавать равным единице.

Для рисования закодированного символа необходимо задать его размеры и параметры шрифта - углы поворота и наклона символьной площадки. Это выполняют процедуры РСИМ с параметрами РХ, РУ и ШРИФТ с параметрами ПОВОРОТ, НАКЛОН.

После этого отрисовать символ можно обратившись к процедуре СИМ, задав номер символа НС в текущей БС. Система помнит заданные размеры и параметры шрифта, что позволяет обращаться далее к СИМ до тех пор, пока не потребуется изменить размеры или шрифт.

Процедуры РСИМ и/или НАКЛОН употреблять не обязательно, если пользователя устраивают стандартные значения размеров и/или шрифта, установленные в системе до начала работы с библиотеками символов.

Одна из БС в СИГАМ является системной, ее имя фиксируется. Конкретное значение имени системной БС определяется при реализации версии системы, здесь будем называть ее СБС. Описывать СБС не нужно, поскольку ее расположение определяется при генерации системы. Эта БС считается активированной до начала

работы с библиотеками символов. СБС включает все символы ГОСТ 2.304-68 и некоторые дополнительные символы.

Для кодирования СБС была написана специальная программа, состоящая из блоков кодирования рисунков для всех символов и цикла обращений к ПРС. Все символы, содержащиеся в ГОСТ, за- кодированы в точном соответствии с ним. Размеры символьной площадки при кодировке СБС взяты 16 x 28. Поскольку размер шрифта по ГОСТу определяется его высотой РУ, символы, рисуе- мые с размером 16 x 28 мм, будут соответствовать одному из разрешенных ГОСТом размеров шрифта - 28 мм. При пропорциональ- ном изменении размеров символов соответствие сохраняется.

После отрисовки любого символа ГОСТ делается подвод пера в точку, удобную для отрисовки следующего символа текста согласно положениям ГОСТа для текстов. Отрисовку текстов вы- полняет процедура ТЕКСТ с одним параметром, определяющим вы- водимую последовательность символов. Способ задания этого па- раметра зависит от используемой системы программирования.

При кодировке системной БС учитывались также положения ГОСТа, касающиеся толщины линии. Символ рисуется не точно по границам площадки, заданной его размерами, а несколько мень- ше (внутри ее), чтобы внешние края линий лежали на границах пло- щадки при толщине линии, установленной ГОСТом для данного разме- ра шрифта. Большинство типов перьев, применяемых на гра- фопостроителях, чертит линию толщиной 0,2 - 0,5 мм. Для таких перьев требования ГОСТа будут соблюдены в случае, если разме- ры символов будут соответствовать шрифтам высотой 2,5, 3,5 и 5 мм.

Рисование чисел выполняется процедурой ЧИСЛО. Форматы вы- дачи разнообразнее, чем в системе СМОГ. Кроме вывода в экспо- ненциальной форме и с фиксированной запятой реализуется вы- вод с позиционированием и рисование восьмеричных констант.

Обеспечивается также рисование с различными способами вы-вода знака числа и заполнения пустых позиций слева.

Символ можно перевести из одной БС в другую. Эту задачу вы- полняет процедура ПСС с параметрами ИМЯБС1, НС1, НС. Символ с номером НС1 из БС с именем ИМЯБС1 записывается в активную БС и получает в ней номер НС. Если символ с номером НС уже был в активной БС, то его массив кодов заменяется на новый

(это же замечание справедливо и для процедуры ПРС).

Имеется также процедура преобразования символа в рисунок (ПСР) с параметрами НС, ИМЯР, которая выполняет те же действия, что и следующая последовательность операторов:

НРИС (ИМЯР, I);

СИМ (НС);

КРИС (ИМЯР);

но без рисования символа.

Уничтожить символ в БС можно оператором УНИС (НС). Сформированную БС можно записать под другим именем во внешнюю память обращением ДБС (ИМЯН). Активная БС записывается с именем ИМЯН во внешнюю память по адресу, определенному предшествующим описанием БС ИМЯН (в обращении к ОБС).

15. Модульная структура

Вся базовая часть СИГАМ состоит из модулей. Модульная структура обеспечивает эффективное использование оперативной памяти, простоту изменений, возможность генерации вариантов, облегчает отладку системы.

Модуль включает одну или несколько процедур. Размер модуля кратен величине кванта обмена между оперативной и внешней памятью. Модули размещаются во внешней памяти и вызываются в оперативную, когда происходит обращение к одной из его процедур. Модуль может находиться в оперативной памяти до тех пор, пока отведенное ему место не потребуется для размещения других модулей.

В СИГАМ БЭСМ-6 модули написаны на автокоде БЕМII с использованием макросов. Длина модуля кратна сектору (256 ячеек).

По реализуемым функциям модули разбиты на базовые комплексы. Это разбиение достаточно условию, модули разных комплексов могут вызывать друг друга.

Мониторный и сервисный комплексы содержат модули управления, генерации системы, контроля и выдачи ошибок, печатей и т.д.

Рисуночный комплекс содержит модули рисования, оформления кадра и заказа листа, управления рисунками, кодирования и декодирования.

Канальный комплект состоит из модуля управления каналами и модулей, реализующих каналы графического вывода и диалога.

Координатный комплект включает модули работы с системами координат.

Символьный комплект содержит модули работы с библиотеками символов и отрисовки чисел и текстов.

Архивный комплект включает модули управления ОГА и НГА.

Объектный комплект состоит из модулей построения, анализа и преобразования графиков.

I6. Базовый монитор и связь модулей

Для управления базовой частью системы и обмена информацией с лифтовым спецкомплектом в СИГАМ имеется особый модуль, который называется "Базовый Монитор" (БМ). Этот модуль представляет собой СП, оформленную по правилам библиотеки СП БЭСМ-6. При первом обращении модуль вызывается как СП и ему указываются начало и длина поля для базовой части системы СИГАМ.

Для этого лифтовый спецкомплект должен вызвать сгенерированный вариант с диска или МЛ и записать его на МБ в выделенное для СИГАМ "внешнее поле СИГАМ" (ВПС). В начале этого поля располагается БМ, затем модули и в конце поля - СБС.

Обращение к любому модулю и возврат из модуля идут через БМ. Вид обращения не зависит от того, кто вызывает модуль - другой модуль (обращение "изнутри" базовой части) или лифтовый спецкомплект ("извне" базовой части). В обращении указываются имя модуля, имя процедуры модуля и параметры этой процедуры (адреса или значения).

При вызове некоторого модуля БМ прежде всего проверяет, находится ли этот модуль в оперативной памяти. Если да, то управление передается на начало модуля. В противном случае ищется "максимально свободное" место в выделенной для базовой части СИГАМ области оперативной памяти, называемой "оперативное поле СИГАМ" (ОПС). Расположенные в этом месте ОПС модули записываются в отведенные им области ВПС, на освободившееся место ОПС считывается вызываемый модуль и на него передается управление.

Длина модуля кратна сектору МБ (абзацу оперативной памяти). Для управления обменом между ОПС и ВПС в БМ имеются таблица используемая абзацев (ТИА) и массив информации о модулях (МИМ). Элемент ТИА равен нулю, если данный абзац свободен, иначе он содержит номер модуля в МИМ, которому принадлежит данный абзац.

МИМ формируется во время генерации варианта СИГАМ и содержит имя модуля, адрес его в ВПС, длину модуля (количество секторов или абзацев), приоритет модуля (число от 0 до 15), количество считываний модуля из ВПС в ОПС, номер начального абзаца модуля в ОПС (если модуля нет в ОПС, то нуль) и некоторую дополнительную информацию.

При считывании модуля из ВПС в ОПС ищется максимально свободное место ОПС, вмещающее этот модуль. Пусть n — начальный абзац ОПС, k — конечный абзац, c — количество абзацев, необходимое для размещения модуля. Сканируются элементы ТИА от n до k , ищется свободный отрезок ОПС длиной c . Если такой отрезок есть, то модуль считывается в него. В противном случае свободными считаются также абзацы, занимаемые модулями с приоритетом 0. Пусть $BO(N)$ — количество вызовов модуля с приоритетом 0, которому принадлежит абзац с номером n . Тогда x , на котором достигается минимум

$$\min_{n \leq x \leq k-c+1} F(x) = \sum_{y=x}^{x+c-1} BO(y), \quad Y=X$$

дает номер начального сектора ОПС для размещения заданного модуля.

Если и в этом случае нет места (т.е. не нашелся отрезок ОПС длиной c среди свободных абзацев и абзацев, занимаемых модулями с приоритетом 0, то свободными считаются также абзацы, занимаемые модулями с приоритетом 1 и т.д.).

Модули с наивысшим приоритетом 15 никогда не убираются из ОПС, поэтому если места не хватает и в том случае, когда просмотрены абзацы, принадлежащие модулям с приоритетами меньшими 15, то длина ОПС недостаточна для нормальной работы данного варианта СИГАМ.

Если место для модуля в ОПС найдено, то расположенные в

нем модули записываются в ВПС, информация об этом заносится в ТИА и МИМ, заданный модуль считывается в ОПС, что отмечается в ТИА, а его адрес в ОПС записывается в МИМ.

Кроме размещения модулей, БМ управляет размещением модульных массивов (ММ). ММ – это массив, который может быть затребован любым модулем в ОПС. Его размер кратен абзацу. Модульные массивы размещаются всегда с начала ОПС и не могут быть убраны в ВПС, как и модули с приоритетом 15. Освободить поле в ОПС, занятое ММ, может только тот модуль, который его заказал. Массив отличается от модуля прежде всего тем, что его место в ОПС постоянно, а место модуля может меняться от вызова к вызову.

Более того, модуль может быть убран из ОПС в тот момент, когда он выдал в БМ запрос на обращение к другому модулю или заказ на ММ. Поэтому все адреса, переданные как параметры в некоторый модуль, не должны быть адресами ячеек модуля, так как в момент, когда они будут использоваться другим модулем, там может находиться совсем не та информация.

Отсюда следует, что параметр-адрес может указывать либо адрес вне ОПС, либо адрес в ММ, либо адрес в БМ (место БМ также не меняется, по крайней мере, до выхода из базовой части). Заметим, что даже внутри любого модуля (кроме тех, чей приоритет равен 15) использование абсолютных адресов ограничено периодом между двумя выходами из модуля для обращения к другому модулю или для заказа массива.

Для передачи параметров в БМ имеется специальный транзитный массив, адрес которого (относительно начала БМ) фиксирован. Фиксированы также относительные адреса некоторых параметров, расположенных в БМ, значения которых могут потребоваться другим модулем.

База (адрес начала) БМ хранится в одном из индекс-регистров, который не должен изменяться ни одним модулем. При обращении к любому модулю БМ устанавливает также и его базу в другой фиксированный индекс-регистр. Остальные индекс-регистры модуль может использовать без ограничений, причем их состояние сохраняется БМ при обращении к другому модулю. Для этого в БМ имеется магазинный массив, который рассчитан на максимально возможную глубину обращений в данной версии СИГАМ.

17. Генерация системы

Весьма часто возникает потребность в генерации специализированных версий СИГАМ. Например, в некотором конкретном ВЦ может эксплуатироваться лишь часть графических устройств, работа с которыми обеспечивается системой, и решаться лишь часть задач, которые способна решать система. В этом случае неэффективно держать на МБ во время счета прикладной программы те части системы, которые она использовать не будет. В таком ВЦ окажется полезным сгенерировать усеченную версию системы.

С другой стороны, на ВЦ могут решаться такие задачи, которым требуются совсем другие модули системы. Для таких задач также желательна генерация специализированной версии. При генерации могут устанавливаться состав системы, приоритеты модулей, адрес ВПС, адрес СВС, размеры модулей и т.д.

Для выполнения генерации в СИГАМ имеется служебная программа "Генератор системы СИГАМ" (ГСС). Входной информацией для нее служат массивы описания модулей и тестов системы, параметры расположения полей СИГАМ на внешних носителях, список имен, включаемых в данную версию модулей, и список имен запускаемых тестов.

Массив описания модулей содержит имена модулей, адреса модулей на системных МП, длины и приоритеты модулей. Массив описания тестов содержит имена и адреса тестов.

ГСС формирует версию СИГАМ согласно заданному списку модулей, составляет массив МИМ и помещает его в БМ, устанавливает в БМ параметры ВПС, СВС и другие системные параметры, записывает получившийся вариант системы в указанное поле внешней памяти и печатает информацию о своих действиях и их результатах. Затем ГСС последовательно запускает заданные тесты.

Средства генерации СИГАМ, базовый монитор, принципы связи между модулями являются весьма общими, они могут быть использованы в других аналогичных системах, не обязательно в области машинной графики. На их базе можно строить пакеты программ различного назначения, информационные системы и т.д.

Л и т е р а т у р а

- I. Баяковский Ю.М. и др. ГРАФОР: комплекс графических программ на ФОРТРАНе. Вып.5, Структура и основные принципы. - М.: Б.и., 1975. - 58 с. (Препринт/ИДМ АН СССР; 90).
2. Васильева Л.Ф., Горин С.В., Дворжец В.И. и др. СМОГ. Система математического обеспечения графических устройств ВЦ СО АН СССР. - В кн.: Материалы VI Конференции по эксплуатации вычислительной машины БЭСМ-6 (Тбилиси, 19-23 окт. 1976 г.) Программное обеспечение. Тбилиси, 1977, с. 128-139.
3. Горин С.В., Дворжец В.И. СМО дисплея для М-222. - В сб.: Машинная графика и ее применение. Новосибирск, 1974, с. 38-46.
4. Дворжец В.И. Основные принципы графической системы СИГАМ. - Автометрия, 1978, № 5, с. 18-25.
5. Дебелов В.А. Буферная часть СМОГ. - В сб.: Машинная графика и ее применение. Новосибирск, 1974, с. 19-37.
6. Дебелов В.А., Мацокин А.М. Программное обеспечение устройства микрофильмирования КАРАТ. - Новосибирск, Б.и., 1977, 17с. (Препринт/ВЦ СО АН СССР; 60).
7. Дебелов В.А., Мацокин А.М. Структура программного обеспечения графических дисплеев. - Автометрия, 1978, № 5, с. 85-86.
8. Катков В.Л. Программное обеспечение машинной графики для решения научно-технических задач. - В сб.: Вычислительные системы. Новосибирск, 1977, т.71, с. 3-13.
9. Кобелев В.В. Машинная графика для системы БЭСМ-АЛГОЛ. - М.: Наука, 1978. - 175 с.
10. Математическое обеспечение графопостроителей. I уровень: СМОГ. Инструкция по программированию / Под ред. Ю.А.Кузнецова. - Новосибирск, Б.и., 1976. - 118 с. - В надзаг.: Сиб. отд-ние АН СССР, ВЦ.
- II. Математическое обеспечение графопостроителей. II уровень: СМОГ. Инструкция по программированию / Под ред. Ю.А.Кузнецова. - Новосибирск, Б.и., 1976. - 78 с. - В надзаг.: Сиб. отд-ние АН СССР, ВЦ.

12. Christensen C., Pinson E.N. Multi-function graphics for a large computer system. - In: AFIPS Confer. Proc., 1967, FJCC, v.31, p.697-711.
13. METAVISU. A general purpose graphic system. /P.Boullier, J.Gros, P.Lancene a.o. - In: Graphic Languages. Amsterdam-London: North-Holland Publ.Comp., 1972, p.244-267.