

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Дубна

УКЗ. ЧИТ. ЗАЛА

11 - 6176



ЛАБОРАТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ
И АВТОМАТИЗАЦИИ

А.В.Кавченко, А.А.Карлов, Т.Ф.Смолякова

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДИСПЛЕЯ
СО СВЕТОМЫМ КАРАНДАШОМ
НА ЭВМ CDC -1604A

1971

11 - 6176

А.В.Кавченко, А.А.Карлов, Т.Ф.Смолякова

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДИСПЛЕЯ
СО СВЕТОВЫМ КАРАНДАШОМ
НА ЭВМ CDC -1604A

ОИ И
БИБЛИОТ КА

Кавченко А.В., Карлов А.А., Смолякова Т.Ф.

11 - 6176

Программное обеспечение дисплея со световым карандашом
на ЭВМ CDC -1604A

Описана структура, принципы построения и возможности комплекса дисплейных подпрограмм, разработанных в ОИЯИ для дисплея со световым карандашом на ЭВМ CDC -1604A. Формирование, модификация и взаимодействие с изображением на экране дисплея может осуществляться на языке ФОРТРАН или на автокоде. Принятая идеология построения программного обеспечения может быть использована на других ЭВМ.

Сообщения Объединенного института ядерных исследований
Дубна, 1971

I. Назначение и общая характеристика

Для разработанного в ОИЯИ и подключенного к ЭВМ CDC -1604A дисплея со световым карандашом создан комплекс дисплейных подпрограмм, позволяющий пользователю-программисту осуществлять формирование, модификацию и взаимодействие с изображением на экране дисплея на уровне языка ФОРТРАН или на автокоде без необходимости знать подробные сведения о работе аппаратуры дисплея, канала ЭВМ и схем связи. Несмотря на то, что аппаратура дисплея позволяет в настоящее время выводить на экран только точки, благодаря наличию соответствующих подпрограмм возможно формирование на экране векторов, дуг окружностей и текста с логической точки зрения так, как если бы эти функции были реализованы схемным путем.

Существенной особенностью созданного комплекса дисплейных подпрограмм является возможность присвоения отдельным частям изображения (объектам) буквенно-цифровых названий. Наличие названия позволяет производить с отдельным объектом различные действия, такие как перемещение, удаление, замещение, изменение масштаба и др., указывая при этом лишь название требуемого объекта и параметры преобразования и не заботясь о физическом расположении объекта в памяти.

При работе со световым карандашом (СК) программе пользователя доступны не только координаты указанной СК точки, но и название объекта, которому эта точка принадлежит. Это в значительной мере упрощает программирование процедур взаимодействия человека с изображением и особенно полезно при идентификации объектов, положение которых на экране заранее неизвестно (например, если объект движется).

Дисплей не имеет собственной клавиатуры. Однако пользователь может взаимодействовать со своей программой, а следовательно, и с изображением, через пультовую пишущую машинку ЭВМ (ПМ). Помимо средств, предоставляемых ему в этом смысле языком ФОРТРАН, в его распоряжение предоставлены дополнительные средства в виде набора

служебных подпрограмм ввода информации с ПМ. Эти подпрограммы позволяют вводить с ПМ буквенную и числовую информацию одновременно с высвечиванием изображения на экране.

Дисплейная библиотека, насчитывающая в настоящее время около 50 подпрограмм, записана на мониторную магнитную ленту ЭВМ CDC-1604A и является расширением существующей библиотеки стандартных подпрограмм. Этим обеспечивается автоматическая загрузка дисплейных подпрограмм, к которым обращается пользователь в своей программе. Часть дисплейных подпрограмм, которая используется сравнительно редко, а также ряд подпрограмм, созданных в последнее время, могут быть представлены пользователю на перфокартах.

2. Классификация и выбор возможностей

С функциональной точки зрения все дисплейные подпрограммы можно разделить на три класса:

1. Административные подпрограммы, обеспечивающие выделение памяти и ее использование, а также выполняющие всякого рода действия над объектами. К этому классу следует также отнести и вспомогательные подпрограммы, полезные при отладке дисплейных программ пользователя, особенно когда он еще не имеет достаточно опыта при работе с дисплеем.

2. Подпрограммы-генераторы графической информации, с помощью которых формируется изображение таких элементов как точки, векторы, текст и т.п., а также более сложные изображения, получаемые, например, при построении координатных осей, масштабировании и т.п.

3. Управляющие подпрограммы, обеспечивающие независимо от выполнения программы пользователя регенерацию изображения на экране дисплея, работу светового карандаша и пишущей машинки.

Большинство дисплейных подпрограмм написано на языке ФОРТРАН с целью сократить сроки создания программного обеспечения дисплея. Только в тех случаях, когда использование языка ФОРТРАН было невозможно или нецелесообразно, использовался автокод ЭВМ CDC-1604A.

Почти двухлетний опыт использования дисплея при решении конкретных физических задач на ЭВМ CDC-1604A показал, что в разных случаях к программному обеспечению дисплея предъявляются самые раз-

нообразные требования. Эти требования, как правило, трудно одновременно удовлетворить, если создавать комплекс дисплейных подпрограмм как неделимое целое. Стремление иметь универсальное программное обеспечение дисплея с широким набором возможностей в случае больших физических задач часто наталкивается на ограничение со стороны оперативной памяти ЭВМ. С другой стороны, например, в задачах просмотра и редактирования информации, когда нет жестких требований к памяти со стороны самой задачи, расширение возможностей дисплейного обеспечения приводит к существенному повышению эффективности дисплейных методик.

Наиболее гибким подходом, способным в значительной степени удовлетворить самые противоречивые требования и принятым в данном случае, является, по возможности, минимальная зависимость дисплейных подпрограмм друг от друга и наличие функциональной избыточности. Подобный подход позволяет в зависимости от конкретных требований выбрать такой набор возможностей из числа имеющихся, который бы наиболее полно удовлетворял этим требованиям при заданных ограничениях. В частности, функциональная избыточность комплекса дисплейных подпрограмм позволяет в ряде случаев реализовать одни и те же функции различными путями: либо при минимальных усилиях со стороны пользователя простым обращением к соответствующим дисплейным подпрограммам, что увеличивает требуемый объем памяти и, как правило, приводит в какой-то степени к снижению скорости работы, либо когда пользователь предпочитает сам запрограммировать ту или иную процедуру с помощью дисплейных подпрограмм более низкого уровня, которые уже используются в его программе и поэтому не требуют дополнительной памяти.

Дисплейные подпрограммы, доступные пользователю, перечислены в таблице I. Их можно условно разбить на следующие восемь групп:

1. Подпрограммы, обеспечивающие минимальный набор возможностей. Эта группа позволяет осуществить начальную установку внутренних дисплейных переменных и таблиц, выделение памяти для построения изображения, формирование точек и векторов, заданных в абсолютной или относительной форме (допускается построение сплошных и пунктирных векторов) и простейшие действия над объектами, такие как добавление, уничтожение, прекращение высвечивания, но сохранение в памяти и т.п.

2. Подпрограммы, предназначенные для формирования изображения

произвольной буквенно-цифровой информации, а также чисел в заданном формате. В последнем случае пользователь может указать, какой идентификатор должен предшествовать изображению самого числа. Специальный параметр позволяет выбрать один из трех размеров символов.

3. Подпрограммы построения сплошных и пунктирных изображений окружностей и дуг.

4. Подпрограммы, обеспечивающие работу светового карандаша в режимах указания и слежения. После вызова светового карандаша обработка сигналов прерывания от СК осуществляется независимо от выполнения программы пользователя. В любой момент в своей программе пользователь может узнать, имело ли место прерывание от СК, проверив состояние флага светового карандаша, который задается в качестве одного из параметров при вызове СК. Нулевое состояние этого флага говорит об отсутствии прерывания от СК, а единичное - о том, что прерывание имело место и что название указанного СК объекта, координаты помеченной СК точки и код ее яркости доступны через соответствующие переменные, выделенные пользователем при вызове СК.

Режим слежения за СК позволяет пользователю вводить графическую информацию в ЭВМ и на экран непосредственно с помощью СК, а также перемещать световым карандашом объекты, уже существующие на экране.

5. Подпрограммы для ввода буквенно-цифровой и числовой информации с пультовой ПМ. Вывод информации на ПМ в существующей мониторной системе CDC-1604A выполняется достаточно просто через оператор WRITE, тогда как ввод информации с ПМ довольно сложен. Поэтому и было разработано несколько специальных подпрограмм ввода с ПМ. При обращении к подпрограмме ввода произвольной символьной информации ввод символов в заданную пользователем область памяти, а при необходимости их перекодировка и упаковка, осуществляются независимо от выполнения программы пользователя. В любой момент пользователь может узнать, закончен ли прием информации с ПМ, проверив состояние флага клавиатуры, который задается им в качестве одного из параметров при вызове ПМ. Ввод считается законченным, если введено заданное при вызове ПМ число символов или если на клавиатуре была нажата клавиша "возврат каретки". Число действительно введенных символов доступно пользователю через соответствующую переменную, выделенную им для этой цели при вызове ПМ.

Для частных случаев ввода целых чисел и чисел фиксированной запятой предусмотрены подпрограммы, которые при обращении к ним выдают на ПМ указанный пользователем идентификатор (например, в виде XMAX =), обеспечивают проверку того, что введенное число является числом заданного типа и в случае ошибок повторяют запрос на ввод. При обращении к этим подпрограммам выполнение программы пользователя продолжается только после нормального завершения ввода. Поэтому необходимость в задании флага клавиатуры и в проверке его состояния отсутствует.

6. Дополнительные административные подпрограммы для расширения возможностей пользователя с точки зрения действий над объектами и памятью, выделяемой под графическую информацию. К этой группе, в частности, относятся подпрограммы, позволяющие объявить объектом массив графической информации, поступивший в ЭВМ из внешней памяти или по линиям связи, выполнить такие действия над объектом как перемещение его по экрану, замещение старого объекта вновь сформированным и др., определить расположение объекта в памяти и его длину, производить различные действия над отдельными точками выбранного объекта и т.п.

7. Дополнительные подпрограммы-генераторы, предназначенные для формирования графической информации в операциях масштабирования, построения координатных осей, трехмерных перспективных изображений (в том числе с анализом на "невидимые" линии), в операциях поворота трехмерных объектов и т.д.

8. Вспомогательные дисплейные подпрограммы. К ним относятся подпрограммы, позволяющие задержать выполнение программы пользователя на заданный промежуток времени, что удобно, например, при программировании движущихся по экрану объектов, определить степень заполнения графической информацией области памяти, выделенной для построения изображения, осуществить режим работы с координатами в произвольном диапазоне величин и др. К этой же группе относится набор подпрограмм, позволяющий определить состояние и выдать на печать содержимое внутренних дисплейных переменных и таблиц, что весьма полезно при освоении работы с дисплеем, а также при отладке новых подпрограмм для дисплейной библиотеки.

Из перечисленных групп первая группа используется всегда. Остальные группы могут быть выбраны в любой комбинации независимо

друг от друга, исходя из требований конкретной задачи и квалификации программиста. Деление на группы является условным в том смысле, что в большинстве своем подпрограммы внутри групп не зависят друг от друга и могут использоваться выборочно. Особенно это справедливо для групп 6, 7 и 8, где подпрограммы объединены в группы скорее с методической точки зрения.

3. Организация графической информации в ЭВМ

А. Объекты

Изображение на экране строится в виде последовательности точек. Позиция луча определяется двумя 10-разрядными координатами, так что допустимая дискретность составляет 1024×1024 точки при размере полезной области экрана 150×150 мм².

В одном 48-разрядном машинном слове располагается информация об абсолютных координатах двух точек. Каждая координатная пара сопровождается разрядами "метка" и "запрет". Разряд "метка" при нулевом состоянии разряда "запрет" позволяет высвечивать точку с нормальной (0) или повышенной (1) яркостью. Единичное состояние разряда "запрет" обеспечивает установку луча в заданную позицию на экране без подсвета. При существующей скорости обмена между ЭВМ и дисплеем и достаточно большом времени послесвечения экрана возможен вывод на экран до 5000 точек при хорошем качестве.

Совокупность непрерывно расположенных машинных слов с графической информацией, которой присвоено персональное буквенно-цифровое название или номер, рассматривается как объект. Объекты могут располагаться в памяти произвольным образом в зависимости от требований рабочей программы. Поэтому для учета объектов предусмотрены две управляющие таблицы: таблица объектов и таблица названий объектов (рис.1). Каждому объекту, существующему в данный момент в системе, в таблице названий соответствует строка, которая содержит название во внутреннем VCD-коде, присвоенное этому объекту программистом. Одноименная строка в таблице объектов содержит начальный адрес объекта и его длину, а также разряд маски, который определяет, должен ли высвечиваться данный объект на экране.

При добавлении нового объекта административные подпрограммы отыскивают свободную строку в таблице названий (по ее нулевому содержанию), записывают в нее название добавляемого объекта, а в со-

ответствующую строку таблицы объектов заносят параметры объекта. Если производится добавление объекта без высвечивания, то разряд маски устанавливается в единицу, чем запрещается выдача объекта на экран. Если же добавляемый объект должен высвечиваться, то разряд маски устанавливается равным нулю. Имеются подпрограммы, позволяющие изменить состояние разряда маски существующих объектов динамически в процессе работы с изображением. Удаление объекта из числа существующих выполняется за счет очистки соответствующих строк в каждой из таблиц. Никаких действий над самим объектом не производится.

В дальнейшем будем называть объекты с разрядом маски, равным 0, активными, а с разрядом маски, равным 1, — замаскированными (неактивными). Размерность каждой из таблиц равна 64, чем и ограничивается общее число одновременно существующих объектов. Служебная подпрограмма регенерации изображения включается в работу, когда в таблице объектов появляется хотя бы один активный объект. Ее задача состоит в циклическом просмотре таблицы объектов и организации для каждого активного объекта обмена через буферный канал ЭВМ между областью памяти, где расположен объект, и дисплеем. Так как обмен через буферный канал происходит независимо от работы центрального процессора, подпрограмма регенерации, запустив обмен, возвращает управление прерванной программе. После высвечивания очередного объекта по сигналу прерывания вновь вызывается подпрограмма регенерации, которая, располагая информацией о том, какой объект был выдан последним, ищет в таблице объектов следующий активный объект и организует его выдачу на экран. Перед выдачей очередного объекта его название фиксируется в специальной ячейке (внутренней дисплейной переменной) и в случае прерывания от светового карандаша это название становится доступным пользователю.

Однократное изображение на экране всех активных объектов соответствует одному кадру. Частота выдачи кадров синхронизована с частотой сети, что обеспечивает постоянство яркости изображения при изменении объема выводимой на экран информации и помехоустойчивость.

Б. Формирование объектов

Местоположение объекта в памяти определяется тем, какая область памяти была заказана пользователем в качестве текущего буфера перед процедурой формирования объекта. При объявлении текущего буфера

программист указывает в качестве параметров массив, который он выделяет для этой цели, и его длину. Последующие обращения к подпрограммам-генераторам приведут к тому, что в текущий буфер, начиная с его первого слова, будут помещаться точечные (т.е. представляемые совокупностью точек) изображения формируемых геометрических элементов и символов - по две точки в каждом слове. Специальная переменная - маркер первой свободной позиции - следит за тем, в какое место текущего буфера относительно его начала должна быть помещена очередная точка. Непосредственно после объявления текущего буфера маркер указывает на верхнюю половину первого слова объявленного массива.

Завершив формирование нового объекта, программист добавляет его к числу существующих или как активный объект, или как замаскированный, присваивая ему при этом некоторое название или номер через параметр соответствующей административной подпрограммы.

Административные подпрограммы в качестве добавляемого объекта рассматривают информацию, расположенную между началом текущего буфера и первой свободной позицией, на которую указывает маркер. При добавлении объекта помимо описанных ранее действий над управляющими таблицами автоматически выполняется перенос начала текущего буфера на место первой свободной позиции с точностью до границы слова. Поэтому следующий объект может располагаться в памяти вслед за только что сформированным, если, конечно, программист не захочет заказать для нового объекта текущий буфер в другом месте памяти.

На рис.2 показано, как происходит перемещение маркера первой свободной позиции в текущем буфере при формировании объекта, а также перенос начала буфера при добавлении объекта.

Обычно общий буфер для нескольких объектов заказывают, когда предполагается, что эти объекты в дальнейшем изменяться не будут, или что они будут формироваться заново все одновременно. В случае же, когда некоторый объект требуется изменять в процессе работы с изображением независимо от других объектов, удобно, хотя и не обязательно, выделять под такой объект отдельный буфер с тем, чтобы упростить вызов буфера перед каждым новым формированием объекта.

Обычно пользователь работает с максимальной рабочей областью экрана, оперируя с координатами в диапазоне $0 + 1023$. Если встречаются координаты вне этого диапазона, то дисплейные подпрограммы рассматривают только младшие 10 разрядов кодов координат. Этот режим

устанавливается автоматически в начале работы с дисплеем, а также при начальной установке внутренних дисплейных переменных. Однако пользователь может изменять в процессе работы как размеры рабочей области, так и режим обработки точек вне ее. Такая необходимость может возникнуть, например, при операциях масштабирования, когда увеличенная часть изображения не должна выходить за пределы некоторой области на экране, так как вне этой области располагается сопутствующая графическая информация (например, световые клавиши, текст и т.п.)

Предусмотрены дисплейные подпрограммы, позволяющие осуществлять преобразование произвольных величин в координаты экрана с минимальными усилиями со стороны программиста.

ж ж
ж

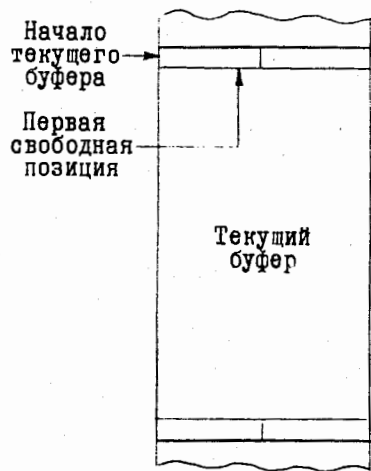
Программное обеспечение дисплея на ЭВМ CDC-1604A прошло длительную проверку при решении разного рода физических задач и доказало свою работоспособность и эффективность. Предоставляя пользователям широкий набор возможностей для формирования, модификации и взаимодействия с изображением на экране, созданное обеспечение предъявляет минимальные требования к аппаратуре дисплея и сопряжению дисплея с ЭВМ. Большинство дисплейных подпрограмм написано на языке ФОРТРАН и особенности аппаратной реализации отражаются лишь на некоторых управляющих подпрограммах. Благодаря этому принятая структура построения программного обеспечения и большинство дисплейных подпрограмм сравнительно просто могут быть адаптированы на другие ЭВМ.

Авторы выражают благодарность А.А. Корнейчуку и А.Ф. Лукьянцеву за полезные обсуждения данной работы, и также Ю.И. Шелонцеву за ценные советы по вопросам программирования на ЭВМ CDC-1604A.

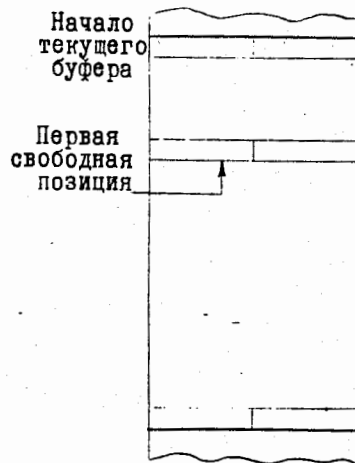
Таблица I

OUTCLEAR	Начальная установка внутренних дисплейных переменных
OUTBUF	Выделение текущего буфера под графическую информацию
OUTDOT	Формирование изображения точки
OUTMOVE	Формирование изображения вектора, заданного в относительной форме
OUTSHIFT	Формирование изображения вектора, заданного в абсолютной форме
OUTADD	Объявление содержимого текущего буфера объектом без выдачи на экран
OUTADDON	Объявление содержимого текущего буфера объектом с выдачей на экран
OUTON	Выдача на экран ранее сформированного объекта
OUTOFF	Прекращение высвечивания объекта на экране с сохранением его в памяти
OUTDEL	Уничтожение объекта
OUTVCD	Формирование изображения символьной информации
OUTNUM	Формирование изображения чисел в заданном формате
OUTARC	Формирование изображения дуги
OUTCIRCL	Формирование изображения окружности
OUTLPEN	Задание параметров для работы СК в режиме указания
OUTCROSS	Выбор режима слежения за СК и формирование маркера
OUTDRAW	Задание способа рисования и шага фиксации промежуточных положений маркера
OUTSTRIP	Выбор режима перемещения объектов по экрану с помощью маркера
OUTKVIN	Ввод с ПМ произвольной символьной информации
OUTENTI	Ввод с ПМ целых чисел с контролем
OUTENTF	Ввод с ПМ действительных чисел с контролем

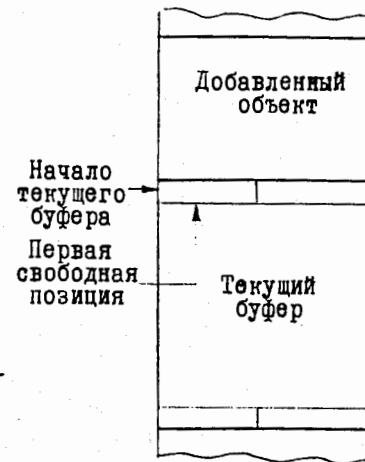
OUTAREA	Объявление текущим буфером с графической информацией произвольной области
OUTCLAIM	Объявление текущим буфером области памяти, занимаемой заданным объектом
OUTEDGE	Установка границ рабочей области экрана и режима обработки координат вне этой области
OUTFIND	Определение физического расположения в памяти и длины заданного объекта
OUTFNDPT	Определение в заданном объекте номера точки с заданными координатами
OUTPICK	Выделение параметров заданной точки
OUTIN	Замещение заданной точки точкой с новыми параметрами
OUTPLACE	Перемещение объекта в заданное место экрана
OUTCUT	Укорачивание объекта на заданную величину
OUTWIDE	Удлинение объекта на заданную величину
OUTREN	Изменение названия объекта
OUTREP	Замещение объекта вновь сформированным
OUTSCR	Задание области на экране при работе с координатами в произвольном диапазоне величин
OUTRANGE	Задание диапазона изменения и типа величин, преобразуемых в координаты экрана
OUTXUC	Преобразование пары величин в координаты точки на экране
OUTPOLY	Формирование изображения последовательности векторов и (или) точек, заданных в относительной форме
OUTGRAPH	Формирование изображения последовательности векторов и (или) точек, заданных в абсолютной форме
OUTCOPY	Копирование объекта в текущий буфер
OUTSCL	Преобразование масштаба объекта
OUTCURVE	Построение изображения функции одной переменной
OUTAXIX	Построение изображения координатных осей
OUTEQI	Построение изображения эквипотенциальной линии



Заказ текущего буфера



Заполнение текущего буфера



Добавление объекта

Рис. 2 Заказ и заполнение текущего буфера