



**ДИАЛОГОВЫЕ СИСТЕМЫ
АВТОМАТИЗАЦИИ
ИССЛЕДОВАНИЙ**

Министерство высшего и среднего специального образования
Латвийской ССР


Латвийский ордена Трудового Красного Знамени
государственный университет имени Петра Стучки

Научно-исследовательский институт физики твердого тела

ДИАЛОГОВЫЕ СИСТЕМЫ
АВТОМАТИЗАЦИИ ИССЛЕДОВАНИЙ

СБОРНИК НАУЧНЫХ ТРУДОВ
(межвузовский)

Латвийский государственный университет им. П.Стучки
Рига 1984



УДК 681.3.1.

ДИАЛОГОВЫЕ СИСТЕМЫ АВТОМАТИЗАЦИИ ИССЛЕДОВАНИЙ: Сборник научных трудов (межвузовский). - Рига: Латв. гос.ун-т им. Петра Стучки, 1984. - 102 с.

В сборнике рассмотрены вопросы организации и применения диалоговых систем в задачах экспериментальной физики твердого тела и программированного обучения пользователей разной категории: специалистов, студентов и др. Сообщается о многопультной диалоговой системе РИГА, созданной в ЛГУ им. Петра Стучки. Приведены технические разработки интерфейса связи дисплеев с мини-ЭВМ для многопультных диалоговых систем.

Сборник предназначен для специалистов и студентов старших курсов, интересующихся автоматизацией научных и дидактических исследований.

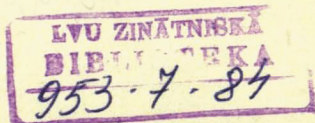
РЕДАКЦИОННАЯ КОЛЛЕГИЯ:

Ю.Кузьмин (отв.редактор), Д.Миллер, В.Полмане,
Я.Страумен, Э.Тарденак, Я.Янсон

Д 20403-048у
МВ12(II)-84

II-доп.84.1704060000

© Латвийский
государственный
университет
им. П.Стучки, 1984



РАЗВИТИЕ ДИСКОВОЙ ОПЕРАЦИОННОЙ СИСТЕМЫ
РЕАЛЬНОГО ВРЕМЕНИ ДЛЯ МИНИ-ЭВМ ТИПА М-6000

М.Востры, А.Н.Кирпа, С.В.Гвоздев, Б.И.Лысов, В.А.Поздеев

Карлов университет, г. Прага, ЧССР

ИВЦ Латвостройтранс

НИИ физики твердого тела ЛГУ им. Петра Стучки

МСУ-97 Моспромтехмонтаж

ИВЦ ЧГУ Минздрава РСФСР

Существующие в настоящее время операционные системы отличаются большим разнообразием сочетаний диалоговых и пакетных режимов, степенью универсальности, различной способностью адаптироваться к изменяющимся условиям эксплуатации, структурой внутренней организации, алгоритмами функционирования и другими параметрами.

Операционные системы реального времени применяются при построении систем, которые непосредственно связаны с событиями внешнего мира, в частности, систем автоматизации научных исследований для управления экспериментальной установкой и накопления измеренной информации, а также для задач АСУТП и т.д. Сюда же относятся задачи, в которых временные ограничения не являются абсолютными, но тем не менее они присутствуют. Например, в системах резервирования мест существует некий естественный предел на время отклика, в системах управления производством требуется иметь к определенному времени сводки о деятельности предприятия.

Естественно, что при разработке программного обеспечения для мини-ЭВМ, которые чаще всего используются в задачах АСНИ и АСУТП, операционные системы реального времени получили большое развитие. С развитием элементной базы возможности мини-ЭВМ значительно возросли, что отразилось на операционных системах. Для повышения эффективности

использования мини-ЭВМ в операционные системы реального времени были включены различные мониторы: пакетной обработки, разделения времени.

Среди мини-ЭВМ класса М-6000 (М-7000, СМ-1) наибольшей популярностью пользуется дисковая операционная система реального времени (DOS RV) [1]. Выбор определяется как необходимостью использовать ЭВМ для решения задач в реальном времени, так и наличием развитого системного прикладного обеспечения, позволяющего не только производить подготовку программ пользователя в фоновом режиме, но и решать широкий круг прикладных задач.

Необходимо отметить, что использование на мини-ЭВМ СМ-1 альтернативной операционной системы DOS АСПО [2,3] не является столь эффективным. Причиной этому служит тот факт, что в DOS АСПО отсутствует свопинг раздела. Динамическая смена программ в разделе ОЗУ возможна только под управлением диспетчера диск-резидентных задач реального времени, являющегося системной задачей и использующего раздел монополюсно. Так как из-за ограниченного размера ОЗУ мини-ЭВМ СМ-1 (64 Кбайта) практически невозможно создать более одного программного раздела для пользователя, то отсюда следует, что в DOS АСПО нельзя работать несколькими пользователями одновременно или же проводить подготовку программ в фоновом режиме. Таким образом, использование DOS АСПО становится эффективным с увеличением объема ОЗУ, что возможно для мини-ЭВМ СМ-2, СМ-2М.

Возвращаясь к DOS RV, следует отметить, что наличие многозадачного диспетчера и свопинга раздела диск-резидентных задач реального времени в DOS RV выгодно отличает эту операционную систему и от дисковой операционной системы (DOS), которую также можно эксплуатировать совместно с мини-ЭВМ данного класса. Явным достоинством системы DOS является ее компактность, что позволяет оставлять под задачи пользователя большой раздел памяти (около 48 Кбайт). Однако неэффективное использование (однопрограммный, однопользовательский режим) мини-ЭВМ при работе под управлением DOS оправдывается в узких классах задач.

Разработка систем мультитерминального доступа (СМД) [4] и мультитерминальной отладки (СМТО) [5,6] явилась логическим развитием заложенной в ДОС РВ возможности организации многопользовательского режима работы. СМД предоставляет механизм для разработки и эксплуатации программ, доступных с нескольких терминалов одновременно. При этом возможно как создавать отдельную копию программы для данного терминала, так и выстраивать очередь к задаче, существующей в единственном экземпляре. СМТО, базируясь на СМД и системных прикладных задачах, является практически завершенной системой для подготовки и отладки программ пользователей.

Особенностью систем СМД и СМТО является то, что они создавались как открытые диалоговые системы. Расширение функций СМТО, включение новых директив не представляет значительных трудностей.

Использование ДОС РВ совместно с системами СМД и СМТО делают практически решенной задачу многопользовательского доступа к мини-ЭВМ рассматриваемого типа. В ряде организаций функционируют многотерминальные системы, решающие задачи АСУТП, АСНИ, АОО (автоматизированного обучения) и т.д.

Вместе с тем, интенсивная эксплуатация мини-ЭВМ под управлением ДОС РВ выявила наличие "узких" мест в операционной системе. Прежде всего это касается использования памяти. Отсутствие свопинга фонового раздела, в котором могут размещаться сегментированные программы, включая и все системные прикладные задачи (трансляторы, система управления файлами), вынуждает размещать в разделе диск-резидентных задач реального времени не только программы управления процессами в реальном времени и основные компоненты СМТО, но и прикладные программы, разработанные пользователями. Это приводит как к снижению скорости реакции задач реального времени, так и к необходимости делить ОЗУ примерно поровну между двумя программными разделами.

Выходом из существующей ситуации является разрешение свопинга фонового раздела. При этом размер раздела можно увеличить, повысив степень его загрузки. Раздел задач реаль-

ного времени соответственно уменьшается. Реактивность задач реального времени при этом возрастает.

Далее, механизм управления системным буфером в ДОС РВ не обеспечивает защиты от программ, у которых скорость поступления запросов не ресурс системного буфера превышает темпы его освобождения. Это приводит к неоправданным задержкам других задач в системе, нуждающихся в этом виде ресурса, а нередко к программным тупикам. Кроме того, при принудительном удалении задачи или отказе буферизованного выводного устройства операционная система не освобождает занятый данной задачей ресурс системного буфера.

К недостаткам ДОС РВ стоит также отнести невозможность заблокировать общее внешнее устройство от доступа другой задачи и отсутствие поддержки многотерминального режима. СМТО в значительной степени дополняет возможности ДОС РВ, но предоставляет пользователю решать самостоятельно проблему копирования задачи при запросе к ней одновременно с нескольких пультов, за исключением редактора символьной информации. Кроме того, при выполнении директивы СМТО пользователь не имеет возможности вмешиваться в работу системы путем ввода команды операционной системы со своего терминала. Это особенно неудобно для пользователей, работающих за терминалами в удалении от ЭВМ.

Для преодоления указанных слабых в ДОС РВ мест была разработана мультитерминальная операционная система реального времени (МОС РВ), являющаяся функционально эквивалентной системе ВТЕ-II для мини-ЭВМ HP2100 [7]. Первоначально система МОС РВ функционировала на мини-ЭВМ М-6000 (процессор А131-7) и СМ-2 (М-7000). В дальнейшем она была адаптирована и для процессора А131-3 М-6000.

В системе МОС РВ, наряду с устранением вышеперечисленных недостатков, введены и дополнительные возможности. В частности, сохраняя практически полную совместимость по формату вызова супервизора операционной системы по отношению к ДОС РВ, набор обращений был расширен, что позволило более эффективно управлять исполнением и синхронизацией задач.

Введение механизма классированного ввода/вывода позволило стандартным образом производить обмен данными между задачами посылками практически неопределенной длины. При этом задача, запросившая ввод данных с указанного ресурса, переводится в состояние ожидания, если данных для нее нет. При появлении для нее информации задача автоматически продолжается. Этот же механизм можно использовать и для управления физическими устройствами ввода/вывода. Система может обеспечивать временное назначение устройств ввода/вывода только определенным программам пользователя. Это особенно важно для задач АСУТП при выводе на печать различных бланков-отчетов о выполнении часовых, сменных, суточных и других заданий.

В МОС РВ используется понятие ресурса. Ресурсом является устройство ввода/вывода, файл, программа или подпрограмма, которые могут быть доступны программе пользователя. Иногда у пользователя возникает необходимость работать с ресурсом, который используется совместно определенным набором программ так, что никакие две из этих программ не могут использовать этот ресурс одновременно. Чтобы осуществлять такой режим работы с ресурсом, упомянутые программы должны иметь возможность устанавливать взаимные блокировки. Метод обмена сигналами, установленными между программами, основан на использовании произвольных номеров идентификации ресурса, доступных программам.

Естественно, что улучшение характеристик операционной системы повлекло за собой модернизацию основных системных прикладных программ, а также разработку программ, дающих новые возможности пользователям. В частности, улучшены трансляторы с языков ФОРТРАН-IV, МНЕМОКОД, АЛГОЛ, позволяющие производить трансляцию из файла в файл, а не с дорожек L8 на L6. Для обеспечения мультитерминального режима важно было отказаться от использования дорожек L8 и L6, назначаемых системой только одному пользователю, что препятствовало использованию трансляторов с нескольких пультов одновременно. Соответственно загрузчик перемещаемых модулей также использует для работы программы из файла. Большим

преимуществом данного загрузчика является то, что он позволяет пользователю создавать персональные библиотеки перемещаемых модулей. Дополнительно в систему МОС РВ включен транслятор с языка ЯСП (язык системного программирования), разработанного в Институте автоматики и электрометрии СО АН СССР и созданного для разработки системных программ. В Центральном институте повышения квалификации (г.Обнинск) подготовлен к эксплуатации на ЭВМ М-6000 препроцессор со структурного варианта языка ФОРТРАН-IV - RATFOR/6000. Препроцессор осуществляет перевод программ с языка, близкого по своим возможностям к ФОРТРАНУ-77 на язык ФОРТРАН-IV. Под управлением МОС РВ работает интерпретирующая система БЭИСИК реального времени.

Улучшены возможности системы управления файлами (СУФ). При сохранении полной совместимости по файловой организации и по формату обращений повышена скорость доступа к файлам. Наличие свопинга в фоновом разделе и использование новой специальной команды СУФ \$У позволяет выполнять все команды операционной системы не выходя из СУФ. В СУФ введен механизм глобальных переменных, позволяющий эффективно использовать аппарат командных файлов. При этом возможны несколько уровней вложения командных файлов, организация циклических исполнений и т.д. Информация при этом передается через глобальные переменные. Использование командных файлов позволяет значительно упростить работу пользователя в системе МОС РВ.

Из новых систем, дополняющих МОС РВ, следует отметить пакетный монитор и сеансовый монитор. Пакетный монитор позволяет исполнять работы, организованные в пакеты. При этом организуется входной поток работ и выходной поток результатов работы. Все результаты работы записываются во временные файлы на диске. После завершения работы пакета результаты можно получить на любом устройстве.

Сеансовый монитор обеспечивает диалоговый режим работы за терминалом операционной системы. При этом допускается автоматическое снабжение пользователя копией программы,

которую он запросил. После завершения сеанса работы сеансовый монитор обеспечивает уничтожение всех временных ресурсов, организованных для данного сеанса. Сеансовый монитор организует авторизованный вход в систему. Пользователю разрешается работа за терминалом, если он правильно ввел свой индивидуальный пароль. При этом ему назначается определенный уровень доступа и диалоговая программа, с которой пользователь будет работать.

Поскольку для работы описанных систем требуется дополнительная память ЭВМ, была разработана драйверная система, позволяющая снизить затраты как на разработку драйверов, так и на объем оперативной памяти, используемой драйверами. Драйверная система состоит из библиотеки драйверов и собственно драйверов различных устройств. Имеются варианты драйверной системы, в которых библиотека и драйверы объединены в виде комбинированного драйвера, а в другом варианте библиотека драйверов включена в системную библиотеку МОС РВ.

Библиотека драйверов включает в себя все общие для драйверов функции: анализ запросов, упаковку/распаковку данных, конфигурацию команд ввода/вывода, организацию возврата в точку продолжения, формирования кодов завершения. Драйверы, лишенные функции связи с операционной системой, становятся короткими и прозрачными, что значительно снижает время их разработки и отладки.

В настоящее время драйверная система поддерживает практически все внешние устройства, имеющие выход на сопряжение 2К: УПК, VT-340, УВВП (FS-1501), УВП (ПЛ-150), СИД-1000, ДМ-2000, ВТА-2000, УБП (VT-343), ДМ-180, УП (DARO -1156), УПЗ (МП-4), МВПК. При этом разработка драйверов для нестандартных устройств не представляет сложностей. Драйверная система не обслуживает устройства, работающие с каналом прямого доступа в память.

Операционная система МОС РВ оставляет доступными большинство системных входов, что позволяет создавать новые и улучшать существующие системные и прикладные программы. Используя основу мультитерминального монитора можно

создавать свои пакеты парольного входа в систему, регистрацию пользователей, защиту данных и введение новых директив оператора.

В Карловом университете (Прага, ЧССР) был разработан улучшенный вариант диалогового редактора символьной информации. Улучшенная версия позволяет производить такие операции, как: копирование части строки в этой же строке, копирование группы строк в другом месте файла, уничтожение группы строк, задание нескольких образцов при поиске строки и т.д. Данный редактор создан на базе библиотеки обработки битов, байтов и символов, разработанной там же. Библиотека состоит из 35 подпрограмм.

Для улучшения сервисных возможностей предназначается программа, осуществляющая чтение файла, состоящего из программ, написанных на различных языках (МНЕМОКОД, ФОРТРАН-IV, АЛГОЛ), вызов соответствующего транслятора, а затем сборку с помощью перемещающего загрузчика из отдельных объектных модулей загрузочного модуля. Программа значительно облегчает сборку систем, написанных на различных языках. Следует также отметить программы копирования дисков на магнитную ленту, получение справочной информации о работающей системе, ее модификация, реконфигурация системы по вводу/выводу и создание архивов.

Из системного прикладного обеспечения необходимо отметить набор различных библиотек подпрограмм, включающих как подпрограммы для научно-технических расчетов, так и подпрограммы для работы с символьной информацией, работы с числовой информацией повышенной точности и т.д., систему управления базой данных, систему сбора данных с технологических рабочих мест, систему, обеспечивающую функционирование сети мини-ЭВМ под управлением МОС РВ. При этом сеть обеспечивает следующие возможности:

- хранение и доступ к данным удаленной ЭВМ;
- возможность взаимодействия программ между собой в различных ЭВМ;
- доступ к периферийным устройствам удаленной ЭВМ как программно, так и по командам оператора;

- возможность планирования команды для удаленной ЭВМ;
- доступ оператора к любой ЭВМ сети;
- возможность подготовки программ для одной ЭВМ, а использования для другой.

При этом все описанные возможности представляются не только для соседних в сети ЭВМ, но и для разделенных несколькими промежуточными. На структуры сети не накладываются никаких ограничений.

Операционная система МОС РВ в существующем варианте значительно превосходит по эксплуатационным характеристикам систему ДОС РВ, а среди операционных систем для мини-ЭВМ с малой памятью (до 64 Кбайт) является одной из лучших.

ЛИТЕРАТУРА

1. Программное обеспечение М-6000/М-7000 АСВТ-М и СМ ЭВМ. Дисковая операционная система реального времени. Краткое описание и руководство по пользованию. З.100.004-01 ОП. 2-я ред. Северодонецк, 1979. 169 с.
2. Айзенберг А.Б., Винокуров В.Г., Костелянский В.М. и др. Агрегатная система программного обеспечения М-7000 АСВТ-М.-УСИМ, 1976, № 6, с.93-98.
3. Айзенберг А.Б., Смолкин В.М., Филиппов Н.В., Щербаков Е.В. Построение ядра операционных систем агрегатной системы программного обеспечения. - УСИМ, 1978, № 2, с.47-50.
4. Программное обеспечение М-6000/М-7000 АСВТ-М и СМ ЭВМ. Система мультитерминального доступа СМД для ДОС РВ-2. Краткое описание и руководство по пользованию. З.100.014Т. Северодонецк, 1979, 39 с.
5. Программное обеспечение М-6000/М-7000 АСВТ-М и СМ ЭВМ. Система мультитерминальной отладки СМО для ДОС РВ-М. Описание программы. З.100.015Т. Северодонецк, 1978, 48 с.
6. Батнер А.А., Круглова Н.А. Программная структура и возможности многопользовательской системы. - В кн.: Автоматизация научных исследований. Рига, 1982, с.26-32.
7. Real-Time Executive II. Programming and Operating Manual (HP-PART. N92001-93001). Hewlett Packard Company, Cupertino, California, 1975.

Статья поступила 9 февраля 1984 года.

ОРГАНИЗАЦИЯ ДИАЛОГА В СИСТЕМЕ АВТОМАТИЗАЦИИ
ВРЕМЯПРОЛЕТНОГО МАСС-СПЕКТРОМЕТРА

О.В.Павлюк

Институт аналитического приборостроения АН СССР
Научно-техническое объединение

Одной из основных характерных черт современного приборостроения для научных исследований является интенсивное применение в них средств вычислительной техники и создания на ее основе автоматизированных систем с диалоговым способом взаимодействия. Причины использования диалога как основного режима проведения эксперимента на автоматизированных аналитических приборах являются достаточно общими и состоят в следующем:

сложность задач управления, обработки и интерпретации результатов делает невозможным применение автоматических систем и требует оптимального сочетания опыта исследователя с мощностью выполнения рутинных операций, предоставляемой вычислительной техникой;

выраженный эволюционный, поисковый характер и "размытость" плана эксперимента, уточняющегося в процессе проведения, делает необходимым обеспечение оперативной реконфигурируемости и гибкости средств его поддержки под управлением пользователя;

оперативный контакт экспериментатора с исследуемым процессом позволяет наиболее адекватно формировать концептуальную модель явления и оптимальным образом осуществлять целенаправленную деятельность по изучению этого явления;

обеспечение комфортного состояния исследователя за счет осуществления психологической естественности взаимодействия, удовлетворения его гностических потребностей и

выдерживания оптимального временного режима работы позволяет улучшить взаимную адаптацию пользователя и системы и в конечном счете повысить производительность труда.

Негативный опыт создания полностью автоматизированных систем, в которых был недооценен "человеческий фактор" [1,2], и экономическая эффективность диалоговых систем [3] подтверждают необходимость и перспективность интерактивного способа взаимодействия в самых различных приложениях.

Перечисленные мотивы обусловили разработку диалоговой системы (ДС) для системы автоматизации времяпролетного масс-спектрометра с лазерной ионизацией (СА МС). Одной из ключевых проблем проектирования ДС является выбор способа взаимодействия пользователя с прибором, т.е. типа диалога. В нашем случае выбор директивного способа общения с СА МС был основан на следующих соображениях.

Во-первых, в большинстве случаев эксперимент на приборах такого рода, как уже отмечалось, носит явно выраженный поисковый характер. Последовательность действий, планируемых исследователем, в сильной степени зависит от получаемых промежуточных результатов и их интерпретации, которая, в свою очередь, разными исследователями представляется по-своему.

Во-вторых, при использовании прибора для широкого круга задач модель получаемого спектра известна лишь приблизительно или может быть неизвестна совсем. Целевая функция, которую реализует пользователь при обработке сигнала, также носит нечеткий характер. Эти обстоятельства приводят к необходимости иметь в составе СА МС и, соответственно, ДС, помимо функций обработки спектра по известным моделям, набор элементарных операций над ним и возможность произвольной конкатенации операций.

В-третьих, в соответствии с основным принципом проектирования ДС-ориентации, рассчитанной на конечного пользователя [4], способ взаимодействия должен соответствовать категории конечных пользователей ДС. Основной категорией пользователей в нашем случае являются пользователи-специалисты с хорошим знанием предметной области и подготовкой,

постоянно работающие с прибором ("компетентные исследователи" в классификации [5]).

Первые два обстоятельства требуют обеспечения максимальной гибкости конструирования планов эксперимента, множество которых в общем случае может являться свободной подгруппой с элементарными операциями-образующими. Третье обстоятельство требует детерминированного, лаконичного, мощного, концептуально единого и психологически естественного способа общения. Диалог на базе языка директив наиболее полно удовлетворяет обоим этим требованиям, предъявляемым к ДС СА МС.

Основными задачами, решаемыми посредством СА МС, являются управление масс-спектрометром, обработка масс-спектрометрической информации и обеспечение взаимодействия экспериментатора с прибором. Далее рассматривается последняя из этих функций, решаемая собственно ДС. Описание директив управления и обработки, непосредственное выполнение которых реализуется прикладным программным обеспечением (ППО), структуры ДС, взаимодействия задач и интерфейса с ППО выходят за рамки данного обсуждения. Перечисляемые ниже средства и системные директивы обеспечивают основные требования к ДС [4] и находятся под непосредственным управлением пользователя.

Каждая директива состоит из двухбуквенного кода, задающего действие, и списка ключевых параметров. Код от списка отделяется пробелом, параметры в списке разделяются запятой. Концом директивы является символ возврата каретки, по которому директива передается ДС на исполнение. Имя параметра (не более четырех символов) отделяется от числового значения знаком равенства.

Каждый параметр может иметь стандартное значение и каждая директива - стандартный список параметров, задаваемых при генерации языка. Таким образом, при вводе директивы пользователь может применять соответствующие умолчания, работая на уровне имен параметров или только кодов директив. Для контроля используемых в данное время умолчаний

ДС после окончания ввода директивы выводит ее в развернутом виде. Любое из умолчаний может быть изменено или введено в процессе диалога: значение параметра - набором ":-" вместо "=", а список параметров - набором "//" в конце директивы. Локальное редактирование строки обеспечивается стандартной клавишей дисплея "забой" (символа), а одиночный символ "/" в конце набранной строки вызывает ее сброс.

Дальнейшее развитие принципа умолчания - от значений параметров и списка параметров к умолчанию последовательности директив - получило свое развитие в концепции макроса. Работа с ним обеспечивается директивами МН, МК, МВ, МУ, МЛ и МР.

Последовательность директив (в том числе и системных), которая будет вводиться и выполняться начиная с директивы, следующей за МН (макрос начать), и кончая директивой, предшествующей МК (макрос кончить), запоминается, образуя макрос, и может быть вызвана для автоматического выполнения по директиве МВ (макрос выполнить). Поскольку макрос создается в ходе "показательного" сеанса с выполнением вводимых директив, то для ликвидации неудачных с точки зрения пользователя действий введена директива МУ (макрос укоротить), удаляющая из него последнюю директиву. По МЛ (макрос ликвидировать) макрос удаляется целиком, а по МР (макрос распечатать) можно в любой момент просмотреть на экране текущее состояние макроса. Директивы, в которых была диагностирована ошибка, в макрос не заносятся.

Приведенная иерархия умолчаний и средства их оперативной смены реализуют адаптацию системы к потребностям пользователей в процессе ее эксплуатации.

Для последующей оценки процесса введения диалога и получения его твердой копии введем директивы протоколирования: ПН, ПВ, ПК, ПР. Протоколирование начинается либо директивой ПН (протокол начать), либо ПВ (протокол выдать). В первом случае протокол фиксируется в файле на диске, во втором - выдается на печать без запоминания в файле.

Директива ПК (протокол кончить) прерывает протоколирование, а ПР (протокол распечатать) распечатывает содержимое файла протокола. В протокол заносятся все реплики, которыми обменивается ДС и пользователь. Сообщения пользователя, начинающиеся с символа "/", ДС рассматривает как комментарий, что дает дополнительные возможности для пояснения выполняемых в диалоге действий.

Обеспечению преемственности между различными сеансами диалога служат работы с состоянием ДС: СЗ, СВ, СС. По СЗ (состояние запомнить) в файле запоминается текущее состояние умолчаний и макроса, а по СВ (состояние вызвать) в ДС устанавливаются умолчания и макрос, запомненные по СЗ. По директиве СС (состояние сбросить) устанавливается состояние умолчаний, заданное при генерации языка. Макрос ликвидируется.

Для того, чтобы дать возможность пользователю проконтролировать себя еще раз в особо ответственных действиях, введена директива ПТ (подтверждение). После нее по завершении ввода любой директивы последняя выводится пользователю еще раз со знаком "?", и ДС ждет повторного подтверждения ее на выполнение. Вновь набранная директива ПТ отменяет этот режим. Следует отметить, что выполнение макроса (по МВ) в режиме подтверждения дает "полуавтоматический" процесс ведения диалога. Если после "?" ввести символы вставки "/В", то ДС выдает приглашение ко вводу, выполняет введенную директиву и снова предлагает пользователю директиву, которую он не подтвердил. Возможность такой вставки совместно с возможностью отмены предлагаемой директивы обеспечивает гибкость использования макроса в "нештатных" ситуациях.

Удовлетворению гностических потребностей пользователя служит директива СП (справка). В любой момент он может получить информацию обо всех возможностях и директивах ДС. Справка построена по принципу иерархического "меню", с возможностями листания назад и "горизонтального" просмотра информации предыдущего уровня иерархии. Завершить сеанс диалога пользователь может набором директивы ФИ (финиш).

Для облегчения восприятия информации на экране дисплея приняты следующие соглашения:

вся информация, относящаяся к одной директиве (директива, сообщение об ошибке, повторный ввод/вывод, информативы) отделяется от другой такой информации пустыми строками, образуя "абзацы" на экране;

приглашением ко вводу директивы и, таким образом, начальными символами директивы являются пара символов "//" в начальных позициях строки в отличие от информативов, начинающихся с ">>" и сообщений об ошибках, сдвинутых вправо на три символа и начинающихся с "жж";

одноразовый вывод информации на экран дисплея (промежуточных или результирующих данных работы директивы) ограничен 20 из 24 строк экрана, тем самым обеспечивается визуальная непрерывность хода диалога (дисплей работает в режиме "роллинга");

взаимодействие с графическим дисплеем (поддиалог), реализующее действия по передвижению маркера, листанию спектра и изменению его масштаба, выделению участков спектра и т.п., выделяется на экране сдвигом начальной позиции набора директив поддиалога на семь символов вправо и другим символом готовности - "х".

Место возникновения ошибки локализуется указанием в сообщении элемента директивы, в котором или окрестностях которого диагностирована ошибка. С целью обеспечения единого ритма работы введена задержка выдачи сообщений об ошибках, а для снижения ошибок набора все необходимые символы располагаются на одном регистре клавиатуры. Число разделителей минимизировано за счет их полифункциональности (например, "/" в конце директивы означает ее сброс, "//" - смену умолчания списка параметров, а "/В/" - вставку директивы в полуавтоматическом режиме).

ДС СА МС реализована на базе ЭВМ СМ-4 со следующим набором внешних устройств: АЦ-дисплей ВТА-2000, НМД ИЗОТ 1370 и плазменного (512x512) графического дисплея 15ИГ-005,



используемого как устройство отображения спектров. Связь прибора с СА и СА с графическим дисплеем осуществляется через аппаратуру КАМАК. Программное обеспечение ДС разработано в среде базовой ОС, основанной на RSX-IIM 3.1, с использованием языков ПАСКАЛЬ и МАКР-ОАССЕМБЛЕРА.

ЛИТЕРАТУРА

1. Говорун Н.Н., Иванченко И.М., Нефедьева Л.С. Диалог в системах автоматизированной обработки данных. - Управляющие системы и машины, 1974, № 1, с.8-13.
2. Чачко С.А. Психологические особенности диалога "человек - ЭВМ" при автоматизированном проектировании. - В кн.: Проблемно-ориентированные диалоговые комплексы. Республиканский науч.-техн.семинар. Тез.докл.Кишинев, 1983, с.31-33.
3. Doherty W.I. The commercial significance of man-computer interaction. Man/Comput.Commun. Infotech State Art.Rept. Maidenhead, 1979, vol.2, p.81-93.
4. Павлюк О.В., Россиков В.В. Диалоговые системы управления и обработки информации: основные требования и их реализация. - Измерения, контроль, автоматизация, 1983, № 3, с.15-26.
5. Крылов А.А., Нафтульев А.И., Черейский М.М. Психология и проектирование диалога человека с ЭВМ. - В кн.: Психология производству и воспитанию. Л., 1977, с.107-115.

Статья поступила 20 октября 1983 года.

ОБРАБОТКА ДАННЫХ ВО ВРЕМЯ ЭКСПЕРИМЕНТА

С.В.Гвоздев, Н.А.Круглова
НИИ физики твердого тела ЛГУ им. Петра Стучки

Обработка экспериментальной информации в области физики твердого тела становится систематическим явлением. Как показано в [1], измерительная информация около 60 % экспериментов (по материалам литературы 1975 года) подвергалась обработке. В настоящее время в связи с развитием мини- и микро- ЭВМ и увеличением числа автоматизированных экспериментальных установок объем обрабатываемой информации значительно вырос. Под обработкой мы понимаем такое преобразование экспериментальных данных, которое позволяет принять обоснованное решение о количественном и/или качественном поведении параметров исследования.

Разнообразие обработки экспериментальной информации зависит от качества и разнообразия самой информации, а также от разнообразия в интерпретации результатов, что, в свою очередь, зависит от уровня теоретических результатов, описывающих процессы в исследуемых объектах. Следовательно, необходимость обработки обуславливается следующими причинами. Во-первых, экспериментальная информация в процессе измерения подвергается различным неконтролируемым изменениям в канале регистрации. Сюда входят как систематические ошибки, вносимые регистрирующими приборами, так и различные шумы и искажения сигнала, появляющиеся из-за нестабильности аппаратуры, поддерживающей заданные уровни воздействующих параметров, например, температуры. Понятно, что даже частичный учет этих факторов является очень сложной задачей.

Во-вторых, во время эксперимента часто регистрируются не основные параметры, интересующие исследователя, а неко-

торые вторичные. В этом случае требуется преобразование информации в соответствии с известной моделью или преобразование координат кривых.

В-третьих, как правило, модель, описывающая исследуемый процесс, неизвестна, либо недостаточно обоснована. При этом задачей экспериментатора является выяснение модели или уточнение ее неизвестных параметров.

Применение ЭВМ в физических исследованиях в известной мере упростило некоторые процедуры первичной обработки и анализа экспериментальных данных. В то же время включение ЭВМ в экспериментальные установки привело к появлению новых методов эксперимента, к увеличению объема информации. Это, в свою очередь, поставило новые задачи при интерпретации экспериментальной информации.

Процедуру обработки можно условно разделить на две части: предварительная (экспресс-анализ) и детальная. Если при детальной обработке экспериментатор оперирует в основном моделью процесса, что позволяет решать наиболее часто встречающуюся в физике твердого тела задачу вторичной обработки - разложение спектров на отдельные полосы и определение их параметров, то экспресс-анализ служит в большей мере для модификации, уточнения или контроля за параметрами экспериментальной установки и ходом эксперимента.

Здесь требуется некоторое замечание относительно терминов первичная обработка и предварительная. Под первичной обработкой понимается такое преобразование данных, которое позволяет использовать более тонкие, чувствительные к ошибкам входных данных алгоритмы. При этом, как правило, не используются априорные данные о характере экспериментальной информации. В то же время предварительная обработка оперирует в основном качественной стороной информации, так как она используется, как отмечалось выше, для настройки экспериментальной установки и уточнения методики проведения эксперимента. Поэтому такой вид обработки, как сглаживание, входит в понятие первичной обработки, но не используется в предварительной.

Разделение обработки на предварительную и детальную позволяет выбирать необходимый объем программного обеспечения при разработке автоматизированной экспериментальной системы. Такую предварительную обработку, как например, усреднение, нормировка, вычитание фона, можно проводить непосредственно в ходе эксперимента (*en line*). Другую часть экспресс-анализа - сравнение кривых, преобразование координат, различные операции над измеренными кривыми можно проводить сразу после окончания очередного цикла измерений (*off line*). Для детальной обработки создаются специальные системы, не связанные непосредственно с экспериментальными системами, а ориентированные на определенные для данных областей методы интерпретации и представления экспериментальных данных. В качестве примера приведем систему БАРИАТОР [2], разработанную для обработки экспериментальных данных, представляемых суперпозицией элементарных процессов известного вида, например, гауссовыми кривыми.

В качестве иллюстрации описанного подхода рассмотрим организацию обработки в автоматизированной системе СПЕКТР [3], разработанной в НИИЭТТ.

Система СПЕКТР создавалась как автоматизированная система для исследования оптических свойств (поглощение, люминесценция, отражение) твердых тел. При разработке системы авторы исходили из того, что автоматизированная система должна удовлетворять следующим требованиям. Система должна допускать расширение набора методик экспериментов, должна обладать простой настройкой на конкретный набор приборов, участвующих в эксперименте. Система должна быть диалоговой и допускать вмешательство оператора в ход эксперимента в любой момент времени, представлять оперативную информацию в удобной для исследователя форме (на экране графического дисплея, в цифровой форме и т.д.).

Что касается обработки измерительной информации в системе СПЕКТР, то здесь не предполагалось включение в нее широких возможностей детальной обработки, как например, сглаживание, разложение спектров на полосы и другие. В сис-

теме СПЕКТР предусмотрена возможность выхода в системы, занимающиеся указанной обработкой.

В системе СПЕКТР решается задача обработки, которую мы называем предварительной. Так, в ходе эксперимента в режиме *on line* может проводиться усреднение измеряемой информации, вычитание фона, нормировка, другие операции, как например, вычисление степени поляризации. Такая обработка зависит от характера эксперимента и поэтому включается непосредственно в алгоритм эксперимента. Измеренная информация может записываться в архив для хранения и последующей более полной обработки.

В системе СПЕКТР архив измеренной и обработанной информации хранится на магнитных дисках. Каждая экспериментальная кривая представляет собой символьный файл, в котором измерительная информация сопровождается комментарием экспериментатора. В комментарий экспериментатор может записать любую информацию, характеризующую данные. Имена файлов состоят из символов, идентифицирующих того или иного пользователя, и порядкового номера файла. Файлы могут быть защищены пользователем. При работе с архивом пользователь может: просмотреть каталог своих файлов на экране дисплея или вывести его на печать; просмотреть или вывести на печать содержимое файла; вывести кривую на экран графического дисплея или на графопостроитель.

В режиме обработки могут использоваться данные из архива и только что измеренная или обработанная информация, еще не записанная в архив. Работа в этом режиме организована по принципу калькулятора, имеющего один универсальный регистр (сумматор) и один вспомогательный регистр для хранения второго операнда при выполнении двухместных операций. Действия пользователя являются естественными с точки зрения работы с обыкновенным калькулятором:

I. Информация (первый операнд) вызывается из архива. Если необходимо, вводится вторая кривая (второй операнд). Только что измеренная кривая всегда является первым операндом.

2. Вводится операция.

3. Если запрошенная операция двухместная и значение второго операнда не определено, то запрашивается и вводится вторая кривая.

4. Вычисленный результат становится первым операндом.

Все операнды, участвующие в данной операции, а затем и результат, высвечиваются на экране графического дисплея.

В системе СПАКТР реализован набор директив, позволяющий осуществлять следующие возможности обработки.

1. Показ кривой из архива на экране графического дисплея. Эта кривая будет использоваться в последующих операциях обработки как первый операнд.

2. Добавление второй кривой к кривой на экране графического дисплея (кривые нормируются перед показом на общий максимум). Добавленная кривая становится вторым операндом.

3. Показ двух кривых на экране графического дисплея (каждая кривая нормируется перед показом на свой максимум).

4. Сложение двух кривых. Результат сложения (как и результаты других операций) остается на экране графического дисплея и становится первым операндом.

5. Вычисление разности двух кривых.

6. Перемножение двух кривых.

7. Деление первого операнда на второй.

8. Вычисление степени поляризации для двух кривых.

9. Отказ от операндов (очистить регистры).

10. Корректировка экспериментальной кривой. Сюда входит:

10.1. удаление точек,

10.2. замена точек.

11. Умножение кривой на константу.

12. Вычисление логарифма X.

13. Вычисление логарифма Y.

14. Вывод информации из архива на печать или дисплей.

При этом выводится имя файла, комментарий к информации в файле и информация.

15. Вывод каталога файлов пользователя на печать или дисплей. Выводятся имена файлов и комментарий к информации.

16. Замена комментариев в файле архива.

17. Запись кривой в архив на диск. Пользователь при записи в архив вводит комментарий. Имя файла формируется автоматически системой. В архив записывается текущий первый операнд.

Набор директив включает как универсальные операции над кривыми, так и специфические виды обработки (вычислительные степени поляризации).

В настоящей версии системы СПЕКТР отсутствует возможность определять на базе существующих директив макродирективы (программируемый калькулятор), что приводит к необходимости вводить дополнительные директивы обработки, являющиеся несложными комбинациями существующих.

В то же время архитектура программного обеспечения системы СПЕКТР позволяет без значительных затрат изменять набор директив обработки в зависимости от потребностей эксперимента.

Л И Т Е Р А Т У Р А

1. Кузьмин Ю.Я. Базовые эксперименты и обработка данных в области физики твердого тела. - Учен. зап. Латв. ун-та, 1976, т.255, с.3-35.

2. Аболиньш Я.Я., Кузьмина Л.М. Анализ спектров комбинационного рассеяния в режиме диалога. - В кн.: Автоматизация научных исследований. Рига, 1982, с.48-61.

3. Гвоздев С.В., Гужа И.А., Круглова Н.А., Савельев В.Л. Автоматизированная система СПЕКТР для исследования оптических свойств твердых тел. - В кн.: Автоматизация научных исследований. Рига, 1982, с.3-12.

Статья поступила 26 января 1984 года.

РАЗВИТИЕ СИСТЕМЫ ВАРИАТОР ДЛЯ ОБРАБОТКИ
СПЕКТРОМЕТРИЧЕСКОЙ ИНФОРМАЦИИЛ.М.Кузьмина
НИИ физики твердого тела ЛГУ им. Петра Стучки

Успешно эксплуатируемая в НИИФТТ диалоговая система ВАРИАТОР ориентирована на обработку спектров, требующих разделения на элементарные полосы, описываемые гауссовыми или лоренцовыми функциями [1]. Создание новых автоматизированных систем для проведения спектрально-кинетических исследований породило поток спектрометрической информации, записываемой в символьные файлы на диски и требующий разнообразной математической обработки.

В рассматриваемой версии система ВАРИАТОР - это пакет прикладных программ, реализованных в рамках многопультной обучающей системы РИГА [2]. Система эксплуатируется на мини-ЭВМ М-6000 с подключенными к ней дисплеями. Все программы пакета написаны на ФОРТРАНЕ-4 для операционной системы ДОС РВ-2.

Важная проблема, возникающая при реализации диалоговых систем - это обучение неподготовленных пользователей работе в системе. Задача легко решается в рамках обучающей системы РИГА. В ней впервые предложена и использована новая информационная единица - диалоговая документированная программа (ДДП), представляющая собой симбиоз программных и текстовых файлов.

Система обработки экспериментальных данных представляет собой пакет диалоговых документированных программ. Весь текст программ объединяется в курс, имеющий обложку, содержание и разделы.

В первой строке обложки курса находится счетчик числа использования курса, далее - название курса, его назначение, авторы и организация.

За обложкой курса обычно следует содержание, в нем перечисляется весь набор программ, обеспечивающих обработку для всех спектрометрических экспериментов. Каждый программный модуль ориентирован на решение отдельной задачи. Для вызова нужной ДДП пользователь вводит соответствующий ей номер в содержании.

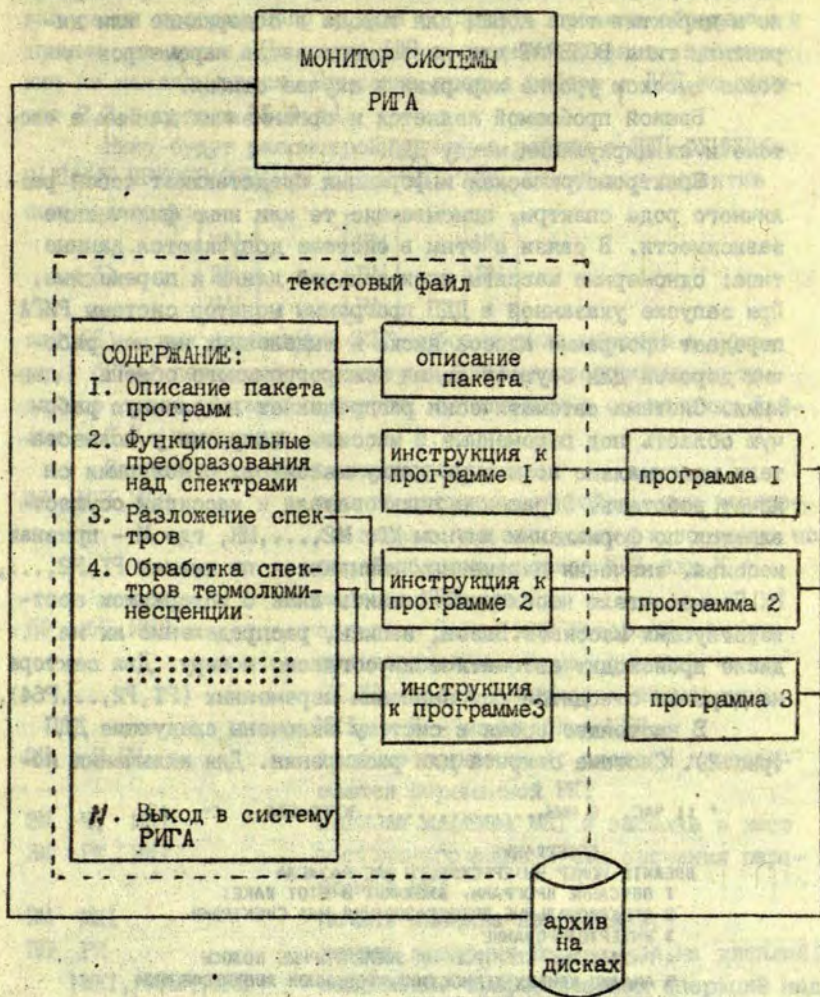
Сама ДДП состоит из текстового и программного модуля (рис. I). В текстовом файле содержатся описание алгоритма решения задачи, сообщения об изменениях в работе программного модуля, описание директив работы с программой, предупреждения о включении нужных устройств. Здесь может быть предложен и небольшой экзамен, после сдачи которого пользователь допускается к работе с программой. Наличие такого текстового файла позволяет пользователям работать с системой без предварительного ее изучения и специальной подготовки.

Программная часть научных ДДП состоит из набора прикладных программ, непосредственно решающих научную задачу. Монитор системы РИГА осуществляет поиск и запуск указанной в ДДП программы.

Работа прикладных программ также организована в режиме диалога. При этом большое внимание уделялось:

- простоте языка,
- строгому синтаксическому контролю вводимых сообщений,
- сохранению работоспособности системы при ошибках пользователя.

При общении пользователя с программами была выбрана самая простая форма диалога, при которой машина после каждого очередного действия предлагает пользователю список возможных директив так называемый 'меню'. Под директивой понимается набор слов, вводимых через запятую. Первое слово - имя директивы, определяет тип действия, остальные - параметры. Широко используется принцип умолчания. В более сложных программах обработки ввод параметров осуществляется по вопросам ЭВМ, организованным по иерархическому принципу. При



Р и с. 1. Структура пакета программ.

этом допускается ввод не только запрашиваемых параметров, но и директив типа КОНЕЦ для выхода в содержание или директивы типа ВОЗВРАТ для повторного ввода параметров на более высоком уровне иерархии в случае ошибки.

Важной проблемой является и организация данных в системе и их циркуляция между ДДП.

Спектрометрическая информация представляет собой различного рода спектры, описывающие те или иные физические зависимости. В связи с этим в системе допускаются данные типа: одномерные массивы произвольной длины и переменные. При запуске указанной в ДДП программы монитор системы РИГА передает программе адреса диска и выделенных на нем рабочих дорожек для осуществления межпрограммного обмена данными. Система автоматически распределяет выделенную рабочую область под переменные и массивы, запросив у пользователя максимально возможную длину массивов, с которыми он хочет работать. Обращение пользователя к массивам осуществляется по формальным именам M_1, M_2, \dots, M_K , где M - признак массива, значения переменных вызываются по именам P_1, P_2, \dots, P_K . Пользователю необходимо помнить лишь о содержимом соответствующих массивов. Вызов, запись, распределение их на диске происходит автоматически согласно номеру. Два сектора на дорожке отводятся под значения переменных (P_1, P_2, \dots, P_{64}).

В настоящее время в систему включены следующие ДДП (рис.2). Система открыта для расширения. Для включения но-

11 ЧАС. В МИН.....КУРС=С36 С= 17

СОДЕРЖАНИЕ :

ВВЕДИТЕ НОМЕР ИНТЕРЕСУЮЩЕГО ВАС РАЗДЕЛА

- 1 ОПИСАНИЕ ПРОГРАММ, ВХОДЯЩИХ В ЭТОТ ПАКЕТ
- 2 ФУНКЦИОНАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ НАД СПЕКТРАМИ
- 3 ИНТЕРПОЛИРОВАНИЕ
- 4 РАЗЛОЖЕНИЕ СПЕКТРОВ НА ЭЛЕМЕНТАРНЫЕ ПОЛОСЫ
- 5 АНАЛИЗ КРИВЫХ ТЕРМОСТИМУЛИРОВАННОЙ ЛУМИНЕСЦЕНЦИИ
- 6 ДЕЙСТВИЯ НАД ПОЛИНОМАМИ
- 7 РАБОТА С МАТРИЦАМИ
- 8 ЛИНЕЙНЫЕ УРАВНЕНИЯ И ИХ СИСТЕМЫ
- 9 ПОСТРОЕНИЕ ГРАФИКОВ НА УЛЗ
- 10 РАЗДЕЛ ЗАМЕТОК

Р и с. 2. Кадр содержания пакета программ.

вой ДДП необходимо создать текстовый файл-инструкцию и при написании программного модуля соблюдать правила обращения к данным на диске. На число ДДП в системе никакие ограничения не накладываются. Описание правил работы с ДДП из пунктов 2,3,4 дано в [1,3,4].

Ниже будут рассмотрены правила работы с ДДП ФУНКЦИОНАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ НАД СПЕКТРАМИ. Общий вид директив пользователя:

$$\langle D \rangle \quad \begin{bmatrix} MN \\ PN \\ CN \end{bmatrix}, \quad \begin{bmatrix} MN \\ PN \\ CN \end{bmatrix}, \quad \begin{bmatrix} MN \\ PN \end{bmatrix},$$

где $\langle D \rangle$ - имя директивы, $\langle MN \rangle$ - обозначение массива с номером, $\langle PN \rangle$ - переменная с номером, $\langle CN \rangle$ - непосредственно заданное число. Формат спектров, записанных в символьных файлах - X, Y построчно.

Список директив пользователя:

- | | | |
|------|--------------------|--|
| ФД | МК1, МК2 | чтение спектра из файла, имя которого задано пользователем, и создание на общей дорожке массива МК1 для X и массива МК2 для Y; |
| ДФ | МК1, МК2 | запись в файл, имя которого задано пользователем, массива пар чисел (X, Y), где X выбирается из массива МК1, а Y - из массива МК2; |
| ЗП | Ч, РК | задание переменной: число Ч присваивается переменной РК; |
| ЗМ | Ч, МК1 | задание массива МК1 и засылка в него |
| ЗМ | РК, МК1 | постоянного числа Ч или значения переменной РК; |
| ПМ | МК1 | печать массива на дисплей; |
| ПП | РК | печать значения переменной на дисплей; |
| <ОП> | МК1, МК2, МК3 | выполнение арифметических операций над массивами МК1, МК2 и засылка результатов в массив МК3. <ОП>- имя операций: |
| | РК1, РК2
Ч1, Ч2 | |
| <ОП> | РК, РК2, РК3 | выполнение арифметических операций с |

+, -, *, /, ↑

Ч1, Ч2	переменными или числами. ОП - имя операции:
ВМ МК1, N1, N2, МК2	выделение из массива МК1 подмассива с точки N1 по N2 и создание нового массива МК2.
<Ф> МК1, МК2	выполнение функционального преобразования над массивом МК1 и засылка результата в МК2, <Ф> - имя функции: SIN, COS, TAN, LN, LOG, EXP, ATAN.
ВГ МК1, МК2	вывод графика на дисплей, МК1 - массив X, МК2 - массив Y.

После выполнения вышеперечисленных преобразований над спектрами для перехода на другие ДДП осуществляется выход в содержание и вызов следующей ДДП, например, программы вывода спектральных зависимостей на графопостроитель и т.д.

Если результаты обработки необходимо сохранить для последующего анализа, их записывают в архив на диски в символическом виде при помощи директивы ДФ. Перечисленные выше директивы расширяются по мере запросов пользователей.

ЛИТЕРАТУРА

1. Кузьмина Л.М., Круглова Н.А., Растопчина В.А. Диалоговая программа для анализа ИК-спектров поглощения. - В кн.: Кибернетизация научного эксперимента, 8. Рига, 1978, с.13-28.
2. Кузьмин Ю.Я. Алгоритм универсальной обучающей программы. - В кн.: Автоматизация научных исследований. Рига, 1982, с.20-25.
3. Глзфеке Х., Кузьмин Ю.Я., Назарова А.Н. Диалоговая система РИГА-РОСТОК для анализа кривых термостимулированной люминесценции. - В кн.: Кибернетизация научного эксперимента, 8. Рига, 1978, с.3-12.
4. Аболиньш Я.Я., Банга А.Я., Бауманис Э.А., Кузьмина Л.М. Организация эксперимента КИНЕТИКА в ДОС РВ М-6000. - В кн.: Автоматизация научных исследований. Рига, 1982, с.13-19.

Статья поступила 17 ноября 1983 года.

ДИАЛОГОВАЯ ПОДСИСТЕМА РАДУГА

Я.Я.Боканс, Л.М.Кузьмина
НИИ физики твердого тела ЛГУ им. Петра Стучки

Один из основных методов изучения свойств твердых оптических сред - измерение оптического пропускания материалов в широком диапазоне спектра. Метод дает информацию о ширине запрещенной зоны материала (нахождение края собственного поглощения), о примесях, дающих полосы поглощения в области прозрачности самого материала, а также о механизмах создания и разрушения центров окраски при воздействии на образец внешних возмущений (температура, ионизирующее излучение, подсветка в полосах поглощения, аддитивное окрашивание и т.д.).

Спектры пропускания исследуемых образцов в большинстве случаев мало информативны, поскольку математическое описание кривых поглощения содержит не величину пропускания T , а оптическую плотность D (или пропорциональную величину полярной экстинкции E). Величины оптического пропускания и оптической плотности при данной длине волны связаны соотношением

$$D = -\text{LOG}(T), \quad (1)$$

где T измеряется в процентах, а D - безразмерная величина. Так как край собственного поглощения описывается законом Урбаха

$$D = D_0 \exp \left[\frac{\delta}{kT_0} \text{Th} \left(\frac{T_0}{T} \right) (E - E_0) \right], \quad (2)$$

где k - постоянная Больцмана, T - температура в градусах Кельвина, E - энергия фотона, D_0, δ, T_0, E_0 - параметры, а полоса поглощения примесного центра или центра окраски описывается гауссовой функцией

$$D(E) = D_M \exp\left(-\frac{(E - E_M)^2}{\delta^2}\right). \quad (3)$$

где D_M - оптическая плотность в максимуме полосы, E_M - положение максимума полосы на шкале энергий, δ - полуширина в единицах энергии, в общем виде формулы (2,3) выражают функциональную зависимость вида

$$D = F(E). \quad (4)$$

Для исследования спектров поглощения используется регистрирующий спектрофотометр СФ-8. Данный прибор имеет широкий спектральный диапазон (200-2500 нм), сравнительно малую дисперсию, сервисные устройства (автоматическая развертка спектра с разными скоростями, автоматическое регулирование раскрытия щелей монохроматора и т.д.), которые делают прибор удобным при эксплуатации. Однако в силу того, что прибор регистрирует только оптическое пропускание, при этом отсчет развертки спектра производится по длине волны, регистрируемый спектр поглощения имеет вид:

$$T = F(\lambda). \quad (5)$$

Для получения спектра поглощения в виде (4) необходимо произвести пересчет по формуле (1) и по формуле

$$E_{(эВ)} = 1239.5 / \lambda_{(нм)}. \quad (6)$$

При ручной обработке пересчет по этим формулам занимает до 70 % времени эксперимента. Для ускорения получения результата была создана автоматизированная подсистема РАДУГА. Она обеспечивает снятие кривых пропускания до и после воздействия на образец ионизирующей радиации, расчет спектральной зависимости оптической плотности наведенного поглощения, вывод спектра на графический дисплей СИГД и его распечатка на ДМ-2000, запись спектра в архив на диск для последующего вывода на графопостроитель.

Состав технических средств. В состав технических средств подсистемы входят спектрофотометр СФ-8, аналоговый блок деления напряжений, усилитель аналогового сигнала, дисплей ДМ-2000 и телевизионный приемник, расположенные непосредственно около экспериментальной установки, мини-ЭВМ М-6000 удаленная от установки на 40 м, графический дисплей СИГД, на который выводится спектр оптической плотности наведенного поглощения после измерения и расчета. Посредством телевизионной камеры и кабельной линии изображение с экрана СИГД передается на телевизионный приемник (ТВ), находящийся около установки. Функциональная схема подсистемы РАДУГА дана на рисунке. Ввод аналоговой информации в ЭВМ осуществляется через АЦП. Считывание показаний АЦП происходит при подаче импульса инициативного сигнала (через каждые 10 нм при постоянной скорости развертки монохроматора спектрофотометра) на вход модуля ввода инициативных сигналов (МВвис). Предусмотрена возможность подачи инициативных сигналов вручную.

Программное обеспечение подсистемы. Программное обеспечение подсистемы РАДУГА написано на языках ФОРТРАН-4 и АССЕМБЛЕР для дисковой операционной системы реального времени (DOS РВ-II). В него входят:

- программа управления экспериментом, осуществляющая диалог с пользователем, измерение спектров и запись их в архив на диск,
- программа анализа прерываний, в случае готовности спектрофотометра запускающая программу измерения спектров,
- программа регенерации изображения на графический дисплей, запускаемая по прерыванию от графического дисплея,
- программа расчета спектров наведенного поглощения,
- программа построения графиков на графопостроителе.

Программа управления экспериментом использует три подпрограммы:

- подпрограмму опроса аналогового ЦП,

- подпрограмму открытия или закрытия прерывания по заданному каналу,
- подпрограмму подготовки буфера для графического дисплея.

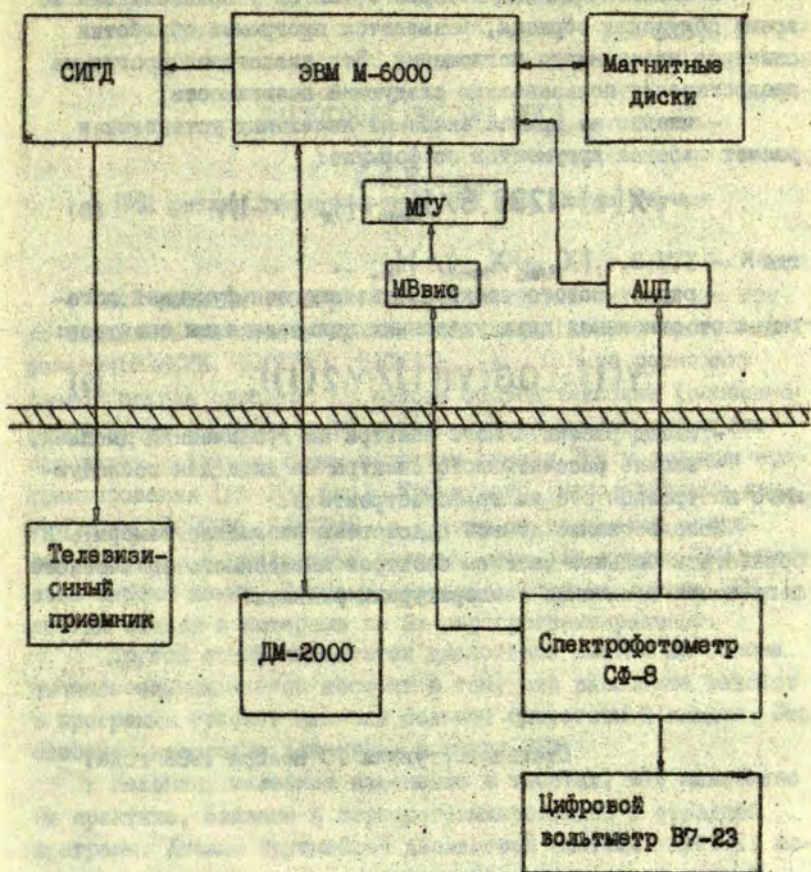
Работа подсистемы. Программа управления экспериментом работает в режиме диалога. После ее запуска с дисплея ДМ-2000, находящегося около установки, на его экран выдается список директив пользователя: ВВОД НАЧАЛЬНЫХ УСТАНОВОК, ИЗМЕРИТЬ СПЕКТР, ПЕЧАТЬ СПЕКТРА НА ЭКРАН ДМ-2000, ЗАПИСЬ СПЕКТРА В АРХИВ НА ДИСК, СПРАВКА, ВЫВОД ОПИСАНИЯ ДИРЕКТИВ НА ЭКРАН ДМ-2000.

При первоначальном запуске программы пользователь в обязательном порядке должен ввести директиву НАЧАЛЬНЫЕ УСТАНОВКИ. При этом вводятся минимальное и максимальное значения аргумента, шаг измерений, значение фона. По директиве ИЗМЕРИТЬ СПЕКТР устанавливается на нуль маска прерывания канала, по которому подключен спектрофотометр и программа управления приостанавливается. После готовности спектрофотометра программа анализа прерываний запустит программу измерения спектра с точки приостанова. Величина пропускания образца T в приборе СФ-8 регистрируется как отношение падений напряжений U_1 и U_2 на сопротивлениях электрических фильтров канала сравнения R_2 и канала образца R_1 .

$$T = U_2 / U_1. \quad (7)$$

Аналоговый блок деления двух напряжений выдает напряжение U , пропорциональное T . Усилитель аналогового сигнала усиливает напряжение U до U_1 для обеспечения нормального функционирования линии связи и АЦП и согласует выход делителя со входом АЦП.

После снятия заданного числа точек спектр можно распечатать на экране ДМ-2000 для визуального просмотра, записать в архив на диск. Для этого пользователь должен ввести имя файла (до 6 символов), и спектр в формате $X (|)$, $Y (|)$ записывается на диск для последующего использования.



Р и с. Функциональная схема подсистемы РАДУГА.

В моменты времени, которые остаются у пользователя во время облучения образца, вызывается программа обработки спектров наведенного поглощения. Эта диалоговая программа предоставляет пользователю следующие возможности:

-- чтение из архива значений начальных установок и расчет массива аргументов по формуле:

$$X(k) = 1239.5 / X_{\min} + H_x \cdot (k-1), \quad (8)$$

где $k = 1, 2, 3, \dots, (X_{\max} - X_{\min}) / H_x; \dots$

-- расчет нового спектра, являющегося функцией логарифма от отношения двух указанных пользователем спектров:

$$Y(i) = \text{LOG}(Y1(i) / Y2(i)); \quad (9)$$

-- вывод рассчитанного спектра на графический дисплей;
-- запись рассчитанного спектра на диск для последующего построения его на графопостроителе.

Использование данной подсистемы позволило измерить и обработать большие массивы спектров наведенного оптического поглощения в сложных температурных режимах.

Статья поступила 10 ноября 1983 года.

ДИАЛоговая СИСТЕМА РИГА

Ю.Я. Кузьмин

НИИ физики твердого тела ЛГУ им. Петра Стучки

Как показал опыт развития диалоговых систем, их можно создавать на основе практически любых языков программирования (БЕЙСИК, ФОРТРАН, ПАСКАЛЬ...). Однако поскольку диалог всегда строится на основе обмена текстами (особенно в обучающих системах), то использование традиционных языков ведет к значительным затратам памяти ЭВМ и времени программирования (до 100 раз). Кроме того, использование языков типа БЕЙСИК, ФОРТРАН и других не позволяет создавать мобильные диалоговые системы, так как каждая ЭВМ имеет свою версию языка. Поэтому перенос систем на другие ЭВМ всегда связан с потерями на их перепрограммирование.

Другой явный недостаток диалоговых систем на основе вычислительных языков состоит в том, что включение текстов в программы требует наличия большой оперативной памяти. Это особенно невыгодно для мини- и микро-ЭВМ.

Наконец, малейшее изменение в текстах, что неизбежно на практике, связано с перепрограммированием и отладкой программы. Анализ крупнейшей диалоговой системы РИГО (II) показал, что в текстовом диалоге всегда есть некий полный набор директив, доступных пользователям: "листание" кадров, запрос справок и содержания текстов, вызов разделов файла и режима калькулятора, выход на режим заметок и т.д. Все это, если не предусмотрено самой системой, отнимает много усилий у авторов, организующих диалог.

Из вышесказанного следует, что применение традиционных языков программирования для создания текстовых диалоговых систем не является оптимальным. Нужен язык текстовой

ориентации. Такой язык должен обеспечивать создание достаточно широкого набора текстовых материалов: программированные курсы, тексты, задачки, каталоги, инструкции, заметки, текстовые имитаторы, игры и т.п. Указанные выше соображения и легли в основу создания диалоговой системы РИГА.

Эта система реализована в виде комплекса технических и программных средств. Технические средства зависят от конкретного исполнения системы и рассчитаны на использование ЭВМ с подключенными к ней дисплеями. Дисплеи позволяют вести работу в режиме диалога пользователя с ЭВМ. Один из действующих вариантов системы РИГА представляет собой мини-ЭВМ М-6000 с 16 одновременно работающими дисплеями. 12 дисплеев вынесены в дисплейный зал, что позволяет проводить групповые занятия. Один дисплей находится в машинном зале для запуска и контроля работы системы. Остальные дисплеи находятся на рабочих местах пользователей. Кроме этой версии, существуют и другие на основе СМ-1 и СМ-4.

Программированные средства состоят из трех компонентов. Первый - это операционная система мини-ЭВМ, которая организует режим деления времени ЭВМ между одновременно работающими пользователями.

Второй компонент - монитор системы РИГА, осуществляющий диалог с пользователем и обеспечивающий поиск и выдачу информации на экран дисплея.

Третий компонент - база данных системы РИГА. Сюда входят программные и текстовые файлы, которые могут иметь специальную структуру.

Система РИГА мобильна относительно базы данных, т.е. вся текстовая информация независима от комплектации и типа ЭВМ. Перенос базы данных на другие ЭВМ требует лишь переноса монитора системы и не затрагивает текстовых файлов. Это достигается унификацией способа структуризации текстовых файлов и унификацией режима диалога с ЭВМ. Структура текстовых файлов вводится исходя из следующих правил:

1. Если файл образует сплошной текст, то система сама разбивает его на последовательные кадры в 20 строк.

Пользователь может рассматривать такой текст в обоих направлениях (с начала - до конца и наоборот).

2. Если файл должен иметь более сложную структуру, то автор файла обязан использовать строки-инструкции, начинающиеся символом α . По этим инструкциям монитор системы реализует представление файла специальным образом, зависящим от назначения инструкции.

Защита информации в системе РИГА. Проблема защиты информации в системе РИГА возникает по многим причинам:

- 1) ограничение доступа пользователей к незаконченной информации;
- 2) защита от некомпетентных пользователей, которые из-за незнания могут впустую тратить ресурсы ЭВМ;
- 3) защита программ от случайных пользователей;
- 4) защита личной информации (личные записки);
- 5) защита конфиденциальной информации (планы, отчеты, характеристики и т.п.);
- 6) защита от пользователей, не умеющих работать с оборудованием;
- 7) Защита информации от случайного разрушения (изменения и уничтожения файлов).

Во всех эти случаях можно применять различные методы защиты. Простейшей защитой является код защиты файла. Код защиты вводится на стадии создания файла. Если он положительный, то информация сохраняется от случайной замены строк файла. Если код отрицательный, то те, кто его не знают, прочитать информацию не сумеют.

Система РИГА может работать как с файлами, имеющими код = -13, так и с незащищенными файлами. В остальных случаях выдается сообщение "Курс недоступен". Для чтения таких курсов необходимо сообщить в режиме выбора курса его код защиты.

Следующая защита файлов основана на том, что файлы, которые рекомендуются пользователям, регистрируются в каталогах. Выход на такой файл происходит автоматически из каталога, если последний оказался доступным пользователю.

Поскольку обычно файлов очень много, то каталоги - практически единственная форма ориентации пользователя. Ограничить на время доступ к файлам, охваченным таким каталогом, проще всего изменением имени каталога. Этот же прием можно использовать, меняя на время имя отдельных файлов. Так можно, например, закрывать доступ к игровым файлам.

Защита файла от некомпетентных пользователей возможна путем использования на его входе теста, который кончается инструкцией $\times B$, при неудачной проверке, либо выходит на защищаемый файл (или его продолжение) для пользователей прошедших тест.

Простейший вариант защиты оборудования, которое связано с работой системы по логике в файле, состоит в выдаче прямых вопросов типа:

```
x?
ВЫ ВКЛЮЧИЛИ ГРАФИЧЕСКИЙ ДИСПЛЕЙ?
xY 2.0.-2
x^x?
```

При ответе "нет" пользователю снова будет задан этот вопрос. Можно вместо $\times \times$? указывать на процедуру включения прибора или обучения работы с ним.

Проверить знание пользователем той или иной программы можно постановкой какого-нибудь вопроса по ее работе, например:

```
x?
КАКУЮ ДИРЕКТИВУ НУЖНО ВВЕСТИ, ЧТОБЫ ЗАВЕРШИТЬ РАБОТУ С ПРОГ-
РАММОЙ ОБРАБОТКИ СПЕКТРОВ?
x#
:BYE
xY 2.0.-1
xB
x#P35AC
```

Если пользователь не знает, что выход из запрашиваемой программы - BУE, то он не попадает на программу. Конечно, это минимальные средства защиты, по сравнению с защитой на основе контроля знаний, но и они эффективнее незащищен-

ных программ. В системе РИГА можно защитить файлы, используя пароль, например:

```
x?
ВВЕДИТЕ ПАРОЛЬ:
x#
  ,КГ ЯСЕНЬ ЛУНА
xУ 2,0,-1
xВ
(ПРОДОЛЖЕНИЕ)
```

Пройти дальше может лишь пользователь, который введет "кг, ясень, луна", остальные - покинут файл. Для большей надежности пароль следует менять, либо комбинировать разные способы защиты. Защиту файлов по месту работы с ними можно производить на основе номера дисплея. Например, если известно, что с приборами эксперимента можно работать только у дисплея I7, то следует включить такую проверку:

```
x+52,46,0 [ЗАДАНИЕ НОМЕРА ДИСПЛЕЯ НА ПРОВЕРКУ]
x)16,18 [ПРОВЕРКА НОМЕРА ДИСПЛЕЯ]
xУ 2,0,-1
xВ
(ПРОДОЛЖЕНИЕ)
```

Здесь проверка основана на том, что система РИГА заносит в системный параметр P(46) номер дисплея пользователя. Номер передается инструкцией x+52,46 в P(6) - параметр, который можно сравнить на максимум и минимум инструкцией x)I6, I8. Если выполнится условие I8>P(6)>I6, то по инструкции xУ 2,0,-1 произойдет переход на продолжение файла, иначе - на xВ, т.е. на выход из файла.

Этот же прием годится для защиты по номерам группы дисплеев, например, для доступа к программе P35AK с дисплеев I0, I1, I2, I3 нужно заменить строку x)I6, I8 на x)9, I4. Файлы, защищенные на основе кода пользователя, строятся похожим образом. Например, для участка файла:

*+ 52,47 [ЗАНЕСЕНИЕ КОДА НА ПРОВЕРКУ]

*)34,36 [ПРОВЕРКА КОДА]

*У 2,0,-1

*В

(ПРОДОЛЖЕНИЕ)

Далее продолжение доступно лишь пользователю, имеющему код регистрации 35 (этот код хранится в P(47)). Так же можно защищаться от группы пользователей. Например; если заменить *) 34,36 на *) 30, то все, имеющие код менее 31, не смогут работать с защищенным файлом. Возможны комбинированные варианты подобной защиты, когда проверяются код защиты и номер дисплея, например:

*+ 52,46

*)16,18

*У 2,0,-1

*В

*+ 52,47

*)34,36

*У 2,0,-1

*В

(ПРОДОЛЖЕНИЕ)

На продолжение файла попадает лишь пользователь, имеющий код 35 и находящийся за дисплеем I7.

Рассматриваемый механизм можно использовать для защиты участков файла от пользователей, которые пытаются войти на них либо слишком быстро, либо слишком медленно. Такая защита необходима при организации процессов, зависящих от времени. В подобном случае следует пользоваться параметрами P(49), P(50) и P(51), в которые система заносит время (в минутах) работы пользователя в системе, курсе и кадре курса, соответственно.

Текстовые имитаторы. Рассмотрим особенности программирования в системе РИГА на примере создания файлов в виде текстовых имитаторов. Текстовым имитатором в системе РИГА называется текстовый файл, основным назначением которого является моделирование диалоговых сообщений систем-

ных или прикладных программ. Идея применения текстовой имитации основана на применении современной методики проектирования программ, получившей название "нисходящего проектирования" (TOPDOWN), а также технологии программирования HIPO, предложенной фирмой IBM.

В первом случае предлагается начинать разработку с верхнего уровня, заменяя элементы низших уровней "затычками", т.е. имитаторами программ.

Во втором случае утверждается, что при проектировании по иерархическому принципу (TOPDOWN) особое внимание надо уделять разработке структуры входной информации и выходной информации для каждого программного модуля.

Имитаторы прикладных программ. Имеются в виду диалоговые программы, поскольку именно они основаны на обмене текстами между ЭВМ и человеком. Система РИГА позволяет производить имитацию работы диалоговых прикладных программ, чему может быть две причины.

Во-первых, такая имитация полезна как инструмент для проектирования программ. В этом случае средствами языка системы РИГА воспроизводится структура будущей прикладной программы. В основе такой структуры может лежать диалог типа "меню", когда пользователю предлагается набор директив, из которых он может выбрать любую, либо диалог со скрытым списком директив, зная которые пользователь вводит их по имени, а специальный интерпретатор вызывает исполнительные программы. Пример простого диалога типа "меню" показан ниже.

```
XC
(КАДР ВЫБОРА ДИРЕКТИВЫ ИНСТРУКЦИЕЙ X1)
X^XC
XN1
(ИМИТАЦИЯ ДИРЕКТИВЫ 1)
X^XC
XN2
(ИМИТАЦИЯ ДИРЕКТИВЫ 2)
X^XC
...
XNN
(ИМИТАЦИЯ ДИРЕКТИВЫ N)
X^XC
```

Здесь в кадре выбора директив можно опробовать разные варианты представления директив пользователю, а в кадре имитации директив можно опробовать разные варианты запросов ЭЕМ и ее ответов на сообщение человека. Цель такой имитации - достижение максимальной однозначности, понятности и простоты диалога. Разработав подобный имитатор, можно опробовать его на потенциальных пользователях и после учета всех замечаний приступить к реализации самой прикладной программы. Вариант со скрытым списком директив можно проектировать, используя инструкцию № III, например:

```
x?
ВВЕДИТЕ ДИРЕКТИВУ:
x#
<ШАБЛОН ДИРЕКТИВЫ A1>
xY 0,2,-1
<ШАБЛОН ДИРЕКТИВЫ A2>
xY 0,2,-1
x:xx2
<РЕАКЦИЯ НА ОШИБКУ В ЗАПРОСЕ>
x^x?
x#1
<ИМИТАЦИЯ ДИРЕКТИВЫ A2>
x^x?
x#2
<ИМИТАЦИЯ ДИРЕКТИВЫ A2>
x^x?
...

```

Данный вариант отличается тем, что пользователь может вводить имя директивы с параметрами. После ввода такой директивы происходит ее сравнение с включенными в список (используя инструкции № III и № У). При наличии директивы в списке происходит ее поиск (x:) и имитация исполнения.

Другая цель имитации прикладной программы состоит в обучении пользователей работе с такой программой. В этом случае средствами языка системы РИГА можно имитировать диалог, лежащий в основе программы. Преимущество состоит в том, что не нужно загружать систему работой с реальной программой, которая может требовать больших ресурсов машинного времени и памяти. Особенно выгодна имитация прикладных программ, связанных с обсервацией, например, с экспериментальными установками. Пользователь, обучаясь на реальной программе, может повредить оборудование из-за незнания особенностей работы программы, что исключено для имитатора. Другим достоинством имитаторов является возможность подробных комментариев на ошибочные действия пользователей.

Имитаторы системных программ. Текстовые имитаторы можно применять для обучения работе пользователей с операционными системами ЭВМ, трансляторами и другими системными программами, а также для проектирования этих средств. Конечно, такая имитация проводится по входу и выходу имитируемой системы. Поэтому она возможна лишь в пределах отдельных задач, что является слабой стороной подхода. Однако здесь не требуется наличия самих имитируемых систем и возможно моделирование фатальных ситуаций с целью демонстрации последствий неверных действий пользователя в системе. Особенно убедителен предлагаемый подход для многопультных диалоговых систем, поскольку он не приводит к останову системы. Чтобы пояснить идею текстовой имитации операционной системы, приведен следующий пример.

Пусть в системе есть только 3 директивы: TI - запрос времени, ON - пуск программы, OF - останов программы. Пусть также в системе есть программа PMGR. Простейший имитатор для этого случая имеет вид (нумерация введена для описания):

```
1  *H  0
2  *?
3  *
4  *3
5  TI
6  *Y  0,4,-1
7  *?
8  TH 1984 15 13 40 55
9  *K
10 *3
11 ON,FMGR
12 *Y  0,4,-1
13 *?
14  *
15 *K
16 *3
17 OF,FMGR
18 *Y  0,4,-1
19 *?
20 FMGR ABORTED
21 *K
22 *?
23 ERROR
24 *K
```

Здесь *Ц 0 позволяет возвращаться ко второй строке по инструкции *K. Строки 2,3 имитируют разрешение операционной системы на ввод директивы (*.- для запрета перевода курсора на следующую строку). Строки 4,5,6,7,8,9 проверяют и обрабатывают запрос времени (TI). Аналогично строки 10-15 и 16-21 проверяют и обрабатывают пуск и останов программы FMGR. Если ни одна из директив (TI, ON, OF) не имела места либо была допущена ошибка при вводе, то имитатор реагирует сообщением ERROR (строки 22-23) и возвращается к строке 2. Конечно, набор директив можно расширить, а некоторые из них (например, ON детализировать путем имитации соответствующих программ (в нашем примере - FMGR).

На основе имитаторов можно строить задачки, обучающие работе с прикладными программами. В этом случае, получив задание, пользователь вводит решение (например, директивы работы с операционной системой, выполняющие предложенное задание). ЭМ согласно имитатору анализирует решение

пользователя, отвечая так, как это произошло бы в реальной ситуации. Дополнительно возможно включение в имитатор вспомогательных инструкций, например, комментариев ошибок человека. Интересной разновидностью являются объясняющие имитаторы. Их сущность состоит в том, что, наряду с имитацией действия реальных программ, выдаются объяснительные тексты по каждому действию. Для вышеприведенного примера можно вставить объяснения в строки 3,8,14,20,23 следующим образом:

3: *	-ВВОДИТЕ ОДНУ ИЗ ДИРЕКТИВ: TI, ON, OF
8: TN 1984 15 13 40 55	-ЭТО ТЕКУЩЕЕ ВРЕМЯ: ГОД, МЕНЬ, ЧАС, МИН, СЕК
14: *	-ВЫ ЗАПУСТИЛИ ПРОГРАММУ FMGR.
20: FMGR ABORTED	-ВЫ ОСТАНОВИЛИ FMGR
23: ERROR	-ВЫ ВВЕЛИ ДИРЕКТИВУ С ОШИБКОЙ.

Завершая рассмотрение имитаторов, укажем на еще одну полезную их особенность. Независимость от операционных систем позволяет реализовать текстовые имитаторы программ одних ЭВМ на других. Таким образом, можно, например, обучать пользователей работе в ОС ЕС на мини-ЭВМ, располагающими соответствующими имитаторами. Аналогично можно обучать работе с микро-ЭВМ либо с операционными системами различных версий для одной и той же мини-ЭВМ.

Имитаторы оборудования. Имитация оборудования возможна в системе РИГА, по крайней мере, в двух режимах: на экране дисплея и в виде макетов, состояние которых опрашивается инструкцией \uparrow ВВОД, а управление осуществляется инструкцией \uparrow ВЫВОД.

В первом случае на экране дисплея ЭВМ выдает, например, вид панели управления имитируемого прибора, указывая специальными знаками или словами кнопки управления и лампочки индикации. В этом же кадре формируется вопрос-задание, например: "Включите прибор". Пользователь, отвечая на вопрос, должен ввести номер кнопки включения прибора, либо ее название. ЭВМ далее задает другие вопросы, выясняющие знания пользователя по данному прибору.

Во втором случае к ЭВМ подключен макет, внешне похожий на имитируемый прибор. Каждая кнопка панели или набор могут быть опрошены в системе РИГА инструкцией χ ВВОД # где # - номер канала ЭВМ, к которому подключен макет.

Результат зависит от места кнопки в канале и ее включения. Всего предусмотрено 16 мест в каждом канале. Каждому месту соответствует буква А, В, С, ... так что старший разряд канала ЭВМ связан с буквой А. Если кнопка макета нажата, то ЭВМ в результате опроса формирует соответствующую букву и знак "плюс", например: +С. Если кнопка не нажата, то знак у буквы будет "минус", например: -Р. Кроме отображения состояния макета буквами, оно же записывается как целое число в системном параметре Р().

Рассмотрим следующий пример работы с подобным имитатором в системе РИГА. Пусть имитатор подключен к 20-му каналу и содержит 4 кнопки: А - включение прибора, В - диапазон I вольт, С - диапазон 10 вольт, D - замер.

```
x?1
ВКЛЮЧИТЕ ВСЕ КНОПКИ
xВВОД 20
x#
, -А -В -С -D
xУ 3,0,-1
ВМ НЕ ВКЛЮЧИЛИ ВСЕ КНОПКИ!
x^x?
x?2
ВКЛЮЧИТЕ ПРИБОР
xВВОД
x#
, +А -В -С -D
xУ 3,0,-1
НАЖМИТЕ КНОПКУ 'ВКЛ'
x^x?
x?3
ИЗМЕРЬТЕ СИГНАЛ НА ДИАПАЗОНЕ 10 ВОЛЬТ
xВВОД
В
, +А -В +С -D
xУ 3,0,-1
НАЖМИТЕ КНОПКИ '10 ВОЛЬТ' И 'ПУСК'
x^x?
ПРАВИЛЬНО!
```

В этом примере происходит пошаговый контроль работы пользователя с прибором. Вопрос 3 предполагает выполнение двух действий: включение диапазона и проведение замера.

Применение инструкции Вывод расширяет возможности имитации за счет введения управляемой от ЭВМ индикации в приборе-макете. Например, в верхнем варианте можно было использовать инструкцию: А = 'сеть', В = 'готов', С = 'диапазон' перегружен. Здесь после инструкции *Вывод* должна следовать строка с выводимой в канал *#* информацией, допустим:

ЛИТЕРАТУРА

I. Smith S.G., Sherwood B.A. Educational Uses of the PLATO Computer System. - Science, 1976, vol.192, p.344-352.

Статья поступила 14 февраля 1984 года.

СТРУКТУРА ДИАЛОГОВЫХ ТЕСТОВ В СИСТЕМЕ РИГА

Ю.Я.Кузьмин

НИИ физики твердого тела ЛГУ им. Петра Стучки

Программированными тестами (или просто тестами) в системе РИГА называются текстовые файлы, основным назначением которых является контроль знаний пользователя либо проведение опросов.

Использование ЭВМ для этих целей является исторически наиболее распространенным в системах автоматизации обучения. Преимущества проверки знаний при помощи программированных тестов перед традиционной проверкой заключаются в более однородной оценке знаний, хорошей документированности процесса контроля и не зависят от уровня преподавательского персонала на месте. Сложности заключаются в больших потерях времени на подготовку тестов и трудности охвата множества возможных правильных ответов проверяемых. Рассмотрим некоторые виды тестов.

Тест-анкета. Этот вид тестов можно использовать для автоматизации опросов разного рода. Структура теста следующая (для ответов "да/нет"):

```

X?
(ВОПРОС ЗНИ)
XN00000   [ДЛЯ ПОДСЧЕТА ВСЕХ ОТВЕТОВ НА ВОПРОС]
XU 0.2.-2 [ДЛЯ ОТВЕТОВ "ДА"]
XN00000   [ДЛЯ ПОДСЧЕТА ОТВЕТОВ "ДА"]

```

Этот элемент повторяется по числу вопросов теста. После ответа на вопросы теста подсчитывается отношение счетчика XN (после XU) ответов "да" на каждый вопрос к счетчику всех ответов (до XU). В случае более сложных ответов используется следующая структура:

к?
(ВОПРОС ЭВИ)
к#
(ШАБЛОН ОТВЕТА)
к#00000 [СЧЕТЧИК ОТВЕТОВ]
к# 0,2 -1 [ЕСЛИ ОТВЕТ НЕ ПРЕДУСМОТРЕН, ТО НА 2-В СТРКУ]
к#00000 [СЧЕТЧИК ПРЕДУСМОТРЕННЫХ ОТВЕТОВ]

Шаблон задается в виде групп ключевых слов. Каждая группа может состоять из набора синонимов.

Тест-зачет. Может быть реализован различным образом, в зависимости от конкретных требований, но в любом случае целью является контроль знаний пользователя ЭВМ. Результатом этих тестов является определение пробелов знаний, либо непонятных мест в предъявленной пользователю информации.

Программированный тест-зачет целесообразно применять после программированных курсов для оценки приобретенных знаний. После установления тестом ошибок в ответах необходимо вновь вернуться к курсу для повторения соответствующих мест. Тест-зачет можно использовать также при допуске к работе с операционной системой ЭВМ, прикладными программами, автоматизированными системами и т.п. Во всех этих случаях при отрицательном результате проверки знаний ЭВМ может автоматически не допускать пользователя к соответствующей работе.

В тестах этого типа нужно проверять в первую очередь знание принципиальных элементов работы, особенно тех, которые могут привести к аварийной ситуации, либо к большим затратам ресурсов.

Тест-альтернатива. Тесты этого типа основаны на повторении элементов, которые предполагают ответ "да/нет":

к?
(ВОПРОС)
к#00000 [СЧЕТЧИК ВСЕХ ВОПРОСОВ]
к# 3,0,-2 [ЕСЛИ НЕПРАВИЛЬНЫЙ ОТВЕТ "ДА"]
к#00000 [СЧЕТЧИК НЕПРАВИЛЬНЫХ ОТВЕТОВ]
к# 6*1,0 [ФИКСИРУЮЩИЕ ОШИБКИ В Р(6)]

Здесь предполагается, что правильный ответ будет "да". В этом случае КУ укажет на 3-ю строку ниже (следующий элемент теста), иначе увеличится на 1 содержание счетчика неверных ответов и в параметре системы P(6) появится 1. Для следующих вопросов нужно поменять номер параметра на P(7), P(8), ... чтобы фиксировать каждую ошибку в своем параметре. По завершении теста все ошибки можно узнать, введя директиву </>.

Если предполагается ответ "нет", то элемент теста имеет вид:

```
к?
(ВОПРОС)
х000000
ку 3.0,-3
х000000
х*6.1.0
```

Во всех случаях вероятность угадывания ответа равна 0,5, однако, чем больше вопросов, тем труднее случайно угадать весь тест.

Выборочный тест. Кадр выборочного теста строится на основе инструкции КС:

```
(ВОПРОС)
ВВЕДИТЕ НОМЕР ВЕРНОГО ОТВЕТА:
-0001 (ОТВЕТ 1 - НЕВЕРНЫЙ)
х1х+ [ПОИСК РЕАКЦИИ НА (ОТВЕТ 1)]
-0001 (ОТВЕТ 2 - ВЕРНЫЙ)
х1х+ [ПОИСК РЕАКЦИИ НА (ОТВЕТ 2)]
-0001 (ОТВЕТ N - НЕВЕРНЫЙ)
х1х+ [ПОИСК РЕАКЦИИ НА (ОТВЕТ N)]
00000
к*жс
х* 6.1.0 [ФИКСАЦИЯ ПЛИВКИ В P(6)]
х00000 [СЧЕТЧИК ОШИБОК]
хх
х000000 [СЧЕТЧИК ВСЕХ ОТВЕТОВ]
```

Этот кадр выдается пользователю в виде:

```
(ВОПРОС)
ВЫБИТЕ НОМЕР ВЕРНОГО ОТВЕТА:
1 (ОТВЕТ 1)
2 (ОТВЕТ 2)
...
N (ОТВЕТ N)
```

После приема номера ответа от человека, ЭВМ ищет место соответствующей реакции на ответ, например, при выборе 1 произойдет поиск α для фиксации ошибки, а затем счетчик ошибок χ N увеличится на 1 и следующий счетчик всех ответов тоже увеличится на 1. При выборе 2 произойдет увеличение только счетчика всех ответов.

Вероятность угадывания здесь теоретически обратно пропорциональна числу выборочных ответов. Однако на практике, имея перед собой весь набор ответов, пользователь может угадать верный ответ чисто логически, даже плохо зная материал.

Тест по эталону. Состоит из элементов вида:

```
*7
(ВПРОС)
*3
(ПРЕДУСМОТРЕННЫЙ ОТВЕТ)
*NOO000 [СЧЕТЧИК ВСЕХ ОТВЕТОВ]
*Y 3,0,-1
*NOO000 [СЧЕТЧИК ОШИБОЧНЫХ ОТВЕТОВ]
*+ 6,1,0 [ФИКСАЦИЯ ОШИБКИ В P(6)]
```

Тест хорош в случае, если мы требуем точного ответа, совпадающего с предусмотренным ЭВМ. Подобная ситуация возникает, например, при контроле знания директив операционной системы.

Вероятность угадывания ответа здесь очень низкая.

Тест по шаблону. Разновидность тестов предполагает анализ ответа по ключевым словам, предусмотренным в шаблоне:

к?
(ВПРОС)
кв
(КЛЮЧЕВЫЕ СЛОВА)
хНО0000 [СЧЕТЧИК ОТВЕТОВ]
хУ 3,0,-1
хНО0000 [СЧЕТЧИК ОШИБОК]
х+ 6,1;0 [ФИКСАЦИЯ ПЛИЕКИ В Р(6)]

Вероятность угадывания ответа здесь весьма низкая, хотя ответы допускаются в широких пределах.

Тест-соответствие. Является комбинацией выборочного теста и теста по шаблону. Суть его состоит в том, что пользователь, видя перед собой две (или более) колонки помеченных элементов, должен установить правильное соответствие между этими колонками, например:

к?
ВЫЯДИТЕ ПОПАРНО ЧЕРЕЗ ЗАПЯТУЮ ИМЕНИ ГОРОДОВ И РЕСПУБЛИК, КОТОРЫМ ОНИ ПРИНАДЛЕЖАТ:
А. ТАЛЛИН Д. ЛАТВИЯ
Б. ВИЛЬНИС Е. ГРУЗИЯ
В. РИГА Ж. ЛИТВА
Г. ЛИБЕВАЯ З. ЭСТОНИЯ
(НАПРИМЕР: АЕ, БА, ВЖ, ГЗ)
кв
АЗ БЖ ВД ГА
хНО0000 [СЧЕТЧИК ОТВЕТОВ]
хУ 3,0,-1
хНО0000 [СЧЕТЧИК ОШИБОК]
х+ 6,1;0 [ФИКСАЦИЯ ОШИБКИ В Р(6)]

Здесь человек должен ввести для всех городов пары букв (первая - город, вторая - республика).

Тесты этого типа угадывать случайно гораздо труднее, чем выборочные, а создавать проще, чем тесты по шаблону (ибо не надо предусматривать все ключевые слова ответов).

Тест с разбором ответа. Разбор ответа можно применять тогда, когда ответ пользователя не совпал с ожидаемым. В таких случаях необходимо предусмотреть заранее причины возможных ошибок. Например, если ответ основан на знании по-

нятий А и В, то причиной ошибки может быть незнание как самих понятий А и В, так и их связей. При разборе ответа происходит сравнение по шаблону для каждого важного компонента, который обязан присутствовать в ответе. Отсутствие любого из них фиксируется отдельно. Например, в матрице ошибок Р(6), ... Р(45) вопрос, основанный на понятиях (А,В), реализуется в виде:

⟨ВОПРОС НА (А,В)⟩

×Ш

⟨ОТВЕТ ПОЛНЫЙ⟩

×У 0,2,-1

[ПРИ УСПЕШНОМ ОТВЕТЕ - НА ×××?]]

×××?

[ПОИСК СЛЕДУЮЩЕГО ВОПРОСА]

×+ 6,1,

[ФИКСАЦИЯ ОБЩЕЙ ОШИБКИ В Р(6)]]

×Ш

⟨ОТВЕТ ОБ'ЯСНЯЮЩИЙ (А)⟩

×У 2,0,-1

[ФИКСАЦИЯ ОШИБКИ (А) В Р(?)]]

×+ 7,1,0

×Ш

⟨ПТРЕГ ОБ'ЯСНЯЮЩИЙ (В)⟩

×У 2,0,-1

[ФИКСАЦИЯ ОШИБКИ (В) В Р(В)]]

×+ 8,1,0

При проверке ответов, требующих полного совпадения с предусмотренными заранее, нужно заменить ×Ш полного ответа на ×Э и далее указать эталон. Для накопления общей статистики следует после каждой инструкции (×У 2) и перед каждой (×У 0) поставить счетчики (×Н00000), а (×У 2,0,-1) заменить на (×У 3,0,-1). При требовании точного ответа следует заменить сравнение по шаблону (×Ш) на сравнение с эталоном (×Э).

Тест с уточняющими вопросами. Разбор ответа в предыдущем тесте происходит незаметно для опрашиваемого. Но не всегда легко организовать такой разбор. Значительно проще уточнить причину ошибки в полном ответе набором самостоятельных вопросов. Пусть полный вопрос основан на компонентах знания (А,В), тогда элемент теста с уточняющими вопросами может быть следующим:

x7*
(ВОПРОС (A,B))
x#
(ПОЛНЫЙ ОТВЕТ (A,B))
xy 0,7,-1
x##
x+A,1,0 [ФИКСАЦИЯ (A,B) - ОШИБКИ В P(6)]
x?
(ВОПРОС НА (A))
x#
(ОТВЕТ (A))
xy 2,0,-1
x+7,1,0 [ФИКСАЦИЯ A - ОШИБКИ В P(7)]
x?
(ВОПРОС НА (B))
x#
(ОТВЕТ (B))
xy 2,0,-1
x+8,1,0 [ФИКСАЦИЯ B - ОШИБКИ В P(8)]

Здесь, как и в предыдущем тесте, можно использовать сравнение по эталону \mathbb{X} Э вместо \mathbb{X} Ш и расставить счетчики всех ошибок \mathbb{X} Н, увеличив при этом в \mathbb{X} У соответствующие номера строк.

Обучающий тест. Основной элемент обучающего теста имеет вид:

x?
(ВОПРОС ЭМ)
x#
x#00000 [СЧЕТЧИК ВСЕХ ОТВЕТОВ]
xy 0,3,-1
x#00000 [СЧЕТЧИК ВЕРНЫХ ОТВЕТОВ]
xix? [ПЕРЕХОД К НОВОМУ ВОПРОСУ]
(ОТВЕТ ЭМ НА ВОПРОС)
x^x? [ВОЗВРАТ К ВОПРОСУ]

Здесь после вопроса ЭМ происходит прием ответа и его сравнение с шаблоном. В случае правильного ответа инкремен-

тируется счетчик (НН) и происходит переход (Н:Н?) к следующему вопросу. При неверном ответе - выдается ответ на вопрос и происходит возврат (Н:Н?) к заданному вопросу.

Обучающий тест с подсказкой. Он строится на основе элемента:

к?	
(ВОПРОС)	[ВЫЯТИТЬ ВОПРОС]
кВ	
(НАВЯЗОН)	[СРАВНИТЬ ОТВЕТ С НАВЯЗОНОМ]
кУ 0,3,-1	[ЕСЛИ ОТВЕТ ПРЕДУСМОТРЕН, ТО НА 3-Ю СТРОКУ]
к+6,0,0	[СВЕРС СЧЕТЧИКА ПОДСКАЗКИ P(6)]
к+к?	[ПОИСК НОВОГО ВОПРОСА]
кУ 2,0,-6	[ЕСЛИ ПОДСКАЗКИ НЕ ВМНО, ТО НА 2-Ю СТРОКУ]
к+кТ	[ПОИСК ПОЛНОГО ОТВЕТА]
к+к.1,0	[СЧИТАТЬ ВЫДАчу ПОДСКАЗКИ В P(6)]
(ПОДСКАЗКА)	[ВЫЯТИТЬ ПОДСКАЗКУ]
к+к?	[ПОВТОРИТЬ ВОПРОС]
кТ	
(ПОЛНЫЙ ОТВЕТ)	[ВЫЯТИТЬ ПОЛНЫЙ ОТВЕТ]
к+к?	[ПОВТОРИТЬ ВОПРОС]

После верного ответа на вопрос ЭВМ происходит переход (Н:Н?) к следующему вопросу. При первом ошибочном ответе выдается подсказка и вновь задается вопрос. После второй попытки - выдается полный ответ и тоже повторяется вопрос. С целью подсчета числа заданных вопросов, подсказок и полных ответов перед ними можно расставить счетчики (НН). Предлагаемый вариант теста с одной подсказкой может быть модифицирован на две и более подсказки.

Обучающий тест с многоступенчатым анализом. Разновидность тестов, позволяющая не только установить факт ошибки, но и провести разбор ответа. Пусть, например, на вопрос "Какая электроника широко используется дома?" - ожидается ответ: "Радиоприемники, магнитофоны и проигрыватели". Ниже приведен пример многоступенчатого разбора ответа:

x? КАКАЯ ЭЛЕКТРОНИКА ШИРОКО ИСПОЛЬЗУЕТСЯ ДОМА,
x# [ПОЛННЯ ШАБЛОН]
•РАДИО ТЕЛЕВИЗ МАГНИТОФОН ПРОИГРЫВАТ, ЭЛЕКТРОФ
xу 0,2,-1
ПРАВИЛЬНО!
xix? [ПОИСК НОВОГО ВОПРОСА]
x#
•РАДИО
xу 3,0,-1
ВЫ ЗАБЫЛИ О РАДИОПРИЕМНИКЕ!
x^x? [ВОЗВРАТ К ВОПРОСУ]
x#
•ТЕЛЕВИЗ
xу 3,0,-1
ВЫ ЗАБЫЛИ О ТЕЛЕВИЗОРАХ!
x^x?
x#
•МАГНИТОФ
xу 3,0,-1
ВЫ ЗАБЫЛИ МАГНИТОФОНЫ!
x^x?
ВЫ ЗАБЫЛИ ПРОИГРЫВАТЕЛИ!
x^x?
x? [НОВЫЙ ВОПРОС]

На любое пропущенное наименование прибора ЭВМ реагирует отдельно и снова возвращает на прежний вопрос. Особенно эффективен этот прием для комбинации с инструкцией (№3), например:

*?
КАК ВЫГЛЯДИТ НА ЯЗЫКЕ BASIC СТРОКА 45, В КОТОРОЙ С КЛАВИА-
ТУРЫ ВВОДИТСЯ ПАРАМЕТР R?
КЭТАЛОН
65 PRINT R
КУ 0,3,-1
ПРАВИЛЬНО!
ККХ?
КВ
65
КУ 3,0,-1
ВЫ ЗАБЫЛИ НОМЕР СТРОКИ 65.
К^Х?
КВ
65 INPUT
КУ 3,0,-1
ПОСЛЕ НОМЕРА ДОЛЖЕН БЫТЬ ОПЕРАТОР INPUT.
К^Х?
КВ
65 R
КУ 3,0,-1
ПОСЛЕ INPUT ДОЛЖЕН БЫТЬ R.
К^Х?
ПРОВЕРЬТЕ СВОИ ОТВЕТ, ОН ДОЛЖЕН ИМЕТЬ ВИД:
65 INPUT R
К^Х?

Подобный анализ можно проводить и без текстовой реак-
ции на ошибку, а используя, например, инструкцию фиксации
ошибок КН - по всем обучаемым, либо К+ - индивидуально.

Тест с предъявлением ответа. Иногда бывает удобно
предъявить пользователю его неверный ответ для сравнения с
правильным ответом, чтобы самостоятельно установить ошибку.
Это достигается путем использования механизма буферизации
ответов пользователей, например;

*?
НАЗОВИТЕ СТОЛПИЦУ ЭСТАЦИИ
КВ
СТАЛЛИН
КУ 0,3,-1
ПРАВИЛЬНО!
ККХ?
ВЫ ОТВЕТИЛИ:
00 [ВЫЛАСА ОТВЕТА ПОЛЬЗОВАТЕЛЯ]
НО ПРАВИЛЬНО БУДЕТ:
СТАЛЛИН
К^Х?

Здесь при ответе с ошибкой: "Таллин" ЭВМ выдаст:

ВЫ ОТМЕТИЛИ:
ТАЛИНН
НО ПРАВИЛЬНО БУДЕТ:
ТАЛЛИН

Тесты с переменными вопросами. Такие тесты целесообразно создавать для проведения групповых занятий. Существует несколько способов создания подобных тестов.

Многовариантные тесты. В этом случае делается несколько тестов по одной и той же теме. Каждому обучаемому предлагается свой вариант.

Тесты с генерацией текстов. Здесь каждый вопрос и подсказка создаются с использованием механизма генерации текста, например.

x?

ЯВЛЯЕТСЯ ЛИ ЧЕЛОВЕК&КОРОВА&СУСЛИК ЖИВОТНЫМ?00

ЭВМ может предъявить следующие вопросы:

"ЯВЛЯЕТСЯ ЛИ КОРОВА ЖИВОТНЫМ?"

"ЯВЛЯЕТСЯ ЛИ СУСЛИК ЖИВОТНЫМ?"

"ЯВЛЯЕТСЯ ЛИ ЧЕЛОВЕК ЖИВОТНЫМ?"

Тесты со случайным выбором вопросов. Такие тесты используют, например, счетчик секунд P(55) или миллисекунд P(46) при выборе одного из нескольких предусмотренных вопросов, например:

x+ 63,55,-30	[P(63)=P(55)-30]
xу 0,2,-63	[ЕСЛИ P(63)>0, ТО НА 2 СТРОКИ НИЖЕ]
xix?#	[ПОИСК ВТОРОГО ВАРИАНТА ВОПРОСА]
x?	
<ВОПРОС 1>	[ПЕРВЫЙ ВАРИАНТ ВОПРОСА]
x#	
<ВАРИАНТ 1>	[СРАВНИТЬ ОТВЕТ С ВАРИАНТОМ]
xу 0,2,-1	[ЕСЛИ ОТВЕТ НЕВЕРНЫЙ, ТО НА 2-В СТРОКУ]
x+6,1,0	[ЗАНЕСТИ ПОВЕДУ ОТВЕТА P(6)]
xix?	[ПОИСК СЛЕДУЮЩЕГО ВОПРОСА]

к?#
(ВОПРОС 2) [ВТОРОЙ ВАРИАНТ ВОПРОСА]
к#
(ШАБЛОН 2)
кУ 0,2,-1 [ЕСЛИ ОТВЕТ НЕВЕРНЫЙ, ТО НА 2-В СТРОКУ]
к+ 6,1,0 [ЗАНЕСТИ ОШИБКУ ОТВЕТА P(6)]

Здесь в рабочем параметре P(6) формируется разность P(55)-30. Если эта разность меньше I, то выдается вопрос I, иначе - вопрос 2. Ошибка ответа по вариантам этих вопросов формируется в P(6). Если нужно фиксировать ошибки по каждому вопросу, то в каждом новом элементе следует использовать при к+ свой счетчик: P(7), P(8), P(9),...

Встречные вопросы в тестах. Встречные вопросы в тестах могут иметь разную причину. Прежде всего, для получения справочной информации, например:

к?
БУДЕТ ЛИ ПЛАВАТЬ ЖЕЛЕЗНЫЙ ШАРИК В РТУТИ?
к#
,ДА,БУДЕ!,КОНЕЧНО
кУ 0,3,-1
ВЕРНО!
к#
,ПЛОТН ?
кУ 3,0,-1 [ОШИБКА]
к+ 6,1,0
ккх?
ПЛОТНОСТЬ ЖЕЛЕЗА= 7.9 (Г/СМЗ)
ПЛОТНОСТЬ РТУТИ= 13.6 (Г/СМЗ)
к^к?

Ответ "да" считается верным. Но можно задать и встречный вопрос: "Как ва их п.отность?" ЭВМ даст справку: "Плотность железа = 7.9 Г:СМЗ. Плотность ртути = 13.6 Г:СМЗ. После справки ЭВМ задает тот же вопрос, при правильном ответе она не засчитает ошибки. При неверном ответе будет зафик-

сирована ошибка $P(6) = 1$. Встречные вопросы могут использоваться и для уточнения основного вопроса. Однако нужно помнить, что предусмотреть все встречные вопросы трудно, поэтому лучше всего добиваться однозначности в основном вопросе.

Проблема неоднозначных ответов. Эта проблема возникает при попытке использования технологии шаблона для анализа ответов на вопросы ЭВМ. Суть состоит в том, что ответ проверяется на присутствие в нем ключевых слов шаблона. Если в ответе имеются как верные слова, так и неверные, то механизм срабатывает по первым, например:

```
x?
ВВЕДИТЕ ВСЕ ЧЕТНЫЕ ЧИСЛА МЕЖДУ 1 И 10!
x#
  2 4 6 8
xY 0,2,-1
ПРАВИЛЬНО!
```

В приведенном примере верными будут считаться ответы 2, 4, 6, 8, а также: 1, 2, 3, 4, 7, 8; 1, 2, 3, 4, 5, 6, 7, 8, 9 и т.п. Одним из простых способов решения этой проблемы является предварительный анализ неверных вариантов:

```
x?
ВВЕДИТЕ ВСЕ ЧЕТНЫЕ ЧИСЛА МЕЖДУ 1 И 10!
x#
  1,3,5,7,9
xY 0,3,-1
ВЫ НАЗВАЛИ И НЕЧЕТНЫЕ ЧИСЛА!
x^x?
x#
  2 4 6 8
xY 3,0,-1
ВЫ НАЗВАЛИ НЕ ВСЕ ЧИСЛА!
x^x?
ПРАВИЛЬНО!
```

В этом варианте вначале проверяется наличие в ответе 1,3,5,7,9. Если обнаружено хоть одно такое число, то выдается сообщение "Вы назвали нечетные числа", а затем вопрос повторяется.

Если названы не все четные числа, то после фразы: "Вы назвали не все числа" вопрос повторяется.

Начало теста. В начале теста нужно дать кадр с правилами ответов, например:

ПРАВИЛА ОТВЕТОВ:

1. ОТВЕЧАЙТЕ С НАЧАЛА СТРОКИ.
2. ОТВЕЧАЙТЕ НЕ ДЛИННЕЕ ОДНОЙ СТРОКИ.
3. ЧИСЛОВЫЙ ОТВЕТ ВВОДИТЕ ЦИФРАМИ.
4. В ЧИСЛОВЫМ ОТВЕТЕ РАЗМЕРНОСТЬ БЕЖИЧКИ НЕ ВВОДИТЬ
5. ПЕРЕЖДЕ ЧЕМ ВВЕСТИ ОТВЕТ, ПОДУМАЙТЕ!

Правила должны отличаться четкостью и краткостью, чтобы их можно было легко запомнить. В них можно включать и предложения психологического характера, имеющие своей целью снять стресс у экзаменуемого. Начало теста может также содержать краткую аннотацию или источник информации, который нужно знать, чтобы тест был успешным.

Завершение тестов. После продолжения основных вопросов теста нужно выдать текст типа: "Экзамен завершен. Позовите инструктора". Чтобы обучаемый не вышел из теста случайным образом, этот кадр нужно оформить как вопросительный с возвратом:

x?
ЭКЗАМЕН ЗАВЕРШЕН. ПОЗОВИТЕ ИНСТРУКТОРА.
x^x?

Здесь при попытке ввести пробел или другую информацию, ЭВМ вернет обучаемого на это же предложение. Вызванный обучаемый инструктор должен ввести директиву (</>), т.е. вызов результата экзамена.

Для удобства расшифровки матрицы ошибок, выдаваемой по директиве (</>), можно ввести перечень возможных ошибок, например:

x?
ВОЗМОЖНЫЕ ОШИБКИ:
1. ПУСК ПРОГРАММЫ.
2. ПРОСМОТР ПРОГРАММЫ.
3. СТРУКТУРА СТРОКИ.
...
x^x?

В таком перечне перечислены ошибки, соответствующие каждому вопросу экзамена, в порядке следования этих вопросов в тесте. Тесты могут кончатся выдачей перечня конкретных ошибок обучаемого или списка курсов, книг, пособий, которые нужно изучить, чтобы не возникало таких ошибок. В этом случае нужно использовать параметры P, например:

xY 0,2,-6 (ЕСЛИ ЕСТЬ ОШИБКА P(6), ТО Н. 2-В СТРОКУ)
xixY (ПЕРЕХОД НА СЛЕДУЮЩУЮ СТРОКУ С xY)
(РЕКОМЕНДАЦИЯ ПО ОШИБКЕ P(6))
xixY

Использование тестовых структур. Тестовые структуры носят универсальный характер и могут быть использованы при создании любых материалов. Например, их можно менять при создании программированных курсов с контролем понимания материала, программированных экзаменаторов, программированных игр и в диалоговых документированных программах обработки данных, утилитах и текстовом сопровождении системных программ

Статья поступила 13 февраля 1984 го.

КОНТРОЛЬ ЗНАНИЙ СТУДЕНТОВ В СИСТЕМЕ РИГА

Л.И.Карташева, И.В.Белова
Кафедра общей физики ЛГУ им. Петра Стучки

На кафедре общей физики для контроля знаний студентов первого и второго курсов приступили к использованию обучающей системы РИГА [1]. Контроль осуществляется в диалоговом режиме при посредстве дисплея. Таким образом проверяется готовность студента к выполнению лабораторной работы. Как известно, перед тем, как допустить студента к очередной лабораторной работе, преподаватель проверяет, как студент понял задачу, разобрался ли он с принципом работы экспериментальной установки, усвоил ли последовательность операций, знает ли, какие физические величины надо определять в прямых измерениях и т.п. Если такая беседа займет хотя бы пять минут, то десятый или двенадцатый студент будет ждать допуска к работе около часа. Индивидуальное собеседование с каждым студентом необходимо по той причине, что в физическом практикуме работы не носят фронтальный характер.

Разгрузить преподавателя в начале занятия и проверить готовность некоторых студентов к лабораторной работе поможет ЭВМ, работающая в режиме диалога. Обучающая система РИГА позволяет составить программу так, что студенту не только задаются вопросы и фиксируются верные или неверные ответы, но и даётся комментарий на эти ответы. При этом возможно также задавать ему наводящие вопросы и возвращаться после пояснения к уже предлагавшимся вопросам. В таком режиме контроля и одновременного обучения, близком к реальным отношениям студента и преподавателя при проведении физического практикума, предусмотрен нами диалог студента и ЭВМ, осуществляемый посредством дисплея.

Противники такого метода обучения могут возразить, что при этом разрушается один из принципов педагогики: преподаватель лишается возможности воспитывать студента личным отношением к предмету, воспитывать культуру общения, культуру языка, не может активно влиять на формирование мировоззрения студента. При такой постановке вопроса не учитывается тот факт, что контакт преподавателя со студентом при проведении физического практикума теснее, чем у лектора и вполне допустимо четверть четырехчасового занятия отвести для работы студента с ЭВМ. Зато при использовании ЭВМ в процессе обучения появляются возможности воспитания у студента качеств, необходимых хорошему специалисту, организатору. Студент приобретает навык давать точный ответ, а не расплывчатый, общение с ЭВМ воспитывает собранность, дисциплинированность, умение выделить главное, поскольку имеется возможность контролировать свое время и нельзя воспользоваться той ситуацией, что преподаватель торопится дать задание всем студентам и задает минимальное число вопросов или др. Студент должен ответить ЭВМ на определенное количество вопросов, охватывающих всю лабораторную работу, а не отдельные элементы, как обычно бывает при устном опросе. При этом вырабатываются такие качества, как обязательное выполнение работы и личная ответственность за свои действия. Положительно и то, что объективно выставленная оценка ЭВМ может послужить стимулом для развертывания соревнования студентов за повышение успеваемости в группе. И если учесть, что будущему специалисту придется работать в эпоху повсеместного использования ЭВМ в народном хозяйстве, то ознакомление студента с подобной техникой на первых курсах имеет для него большое практическое и эмоциональное значение.

Конечно, для составления хорошей программы требуется много времени, а так как система РИГА является открытой системой и программу можно постоянно усовершенствовать, то со временем можно будет научиться составлять программы, отвечающие всем требованиям методики преподавания и составлять их довольно быстро.

Известно, что учебные программы могут быть информационными, консультативными, обучающими и контролирующими. Возможно также в одной программе сочетать элементы названных программ. Кроме того, можно комбинировать линейные и разветвленные программы. Контрольный вопрос следует составить так, чтобы студент мог дать активный ответ. Ответ должен быть однозначным и содержать либо конкретное число, либо вполне определенное слово, либо короткую фразу, содержащую ключевые слова. Пусть, например, требуется проверить знание теоремы переноса осей (теоремы Штейнера). Если спросить формулировку этой теоремы, можно получить ответ в различной редакции и даже ключевые слова не помогут выявить правильный ответ. Поэтому целесообразно задать вопрос в виде задачи, например: во сколько раз момент инерции $I(0)$ диска относительно оси симметрии, перпендикулярной основанию, меньше момента инерции I этого же диска относительно оси, проходящей через край диска и перпендикулярной основанию? Ответ в данном случае однозначен (число 3).

При составлении вопроса можно попросить студента выбрать правильный ответ из предложенного набора возможных ответов. При этом не рекомендуется записывать неверные формулы, не следует давать такие варианты возможных ответов, из которых студент выбрал бы правильные не на основе знаний предмета, а руководствуясь интуицией, угадывая психологию составляющего и т.п. Приведем пример такого вопроса.

Человек, равномерно вращающийся на скамье Жуковского, держит в вытянутых руках две одинаковые гири; не меняя положение рук, он выпускает из них гири. Изменится ли скорость вращения скамьи? Возможные ответы:

- 1 - скорость вращения не изменится,
- 2 - скорость вращения уменьшится
- 3 - скорость вращения увеличится.

Укажите номер правильного ответа.

Можно в рассмотренном примере (правильный ответ - 3) не предлагать варианты ответов, а предусмотреть возможные правильные ответы - "увеличится", "возрастет", "станет

больше", "будет вращаться быстрее", "да". Все эти ответы нетрудно ввести, и ЭВМ будет на них реагировать как на верные. Но последний ответ "да" неполный и не дает уверенности в действительном знании студентом соотношения между моментом инерции и ускорением. Следовательно, предложение вариантов правильных ответов предпочтительнее.

Основная задача преподавателя в физическом практикуме — обучение. Поэтому нужно составлять некоторые вопросы так, чтобы можно было предусмотреть возможные ошибки студента в ответе и в случае неверного ответа прокомментировать его или дать дополнительные пояснения. Ниже приводится структурная схема части алгоритма программы контроля знания с пояснением входящих элементов (рис.).

Рассмотрим последовательность вопросов и комментариев на ответы, вытекающую из этой схемы.

ОСН.ВОПР.1 Чему равна кинетическая энергия обруча массой $m = 1$ кг и радиуса $r = 0,5$ м, катящегося со скоростью $v = 4$ м/с? Рассчитайте в единицах СИ, но ответ введите без наименования единиц.
ж) При верном ответе предлагается основной вопрос 2.

КОММЕНТ.
НА ОШ. 1 Вы ошиблись, учтите, что если тело одновременно участвует в поступательном и вращательном движениях, то его кинетическая энергия равна сумме кинетических энергий поступательного движения и вращения.

ДОП.ВОПР.1 Вычислите кинетическую энергию поступательного движения того же обруча $r = 0,5$ м, $m = 1$ кг, катящегося со скоростью $v = 4$ м/с.

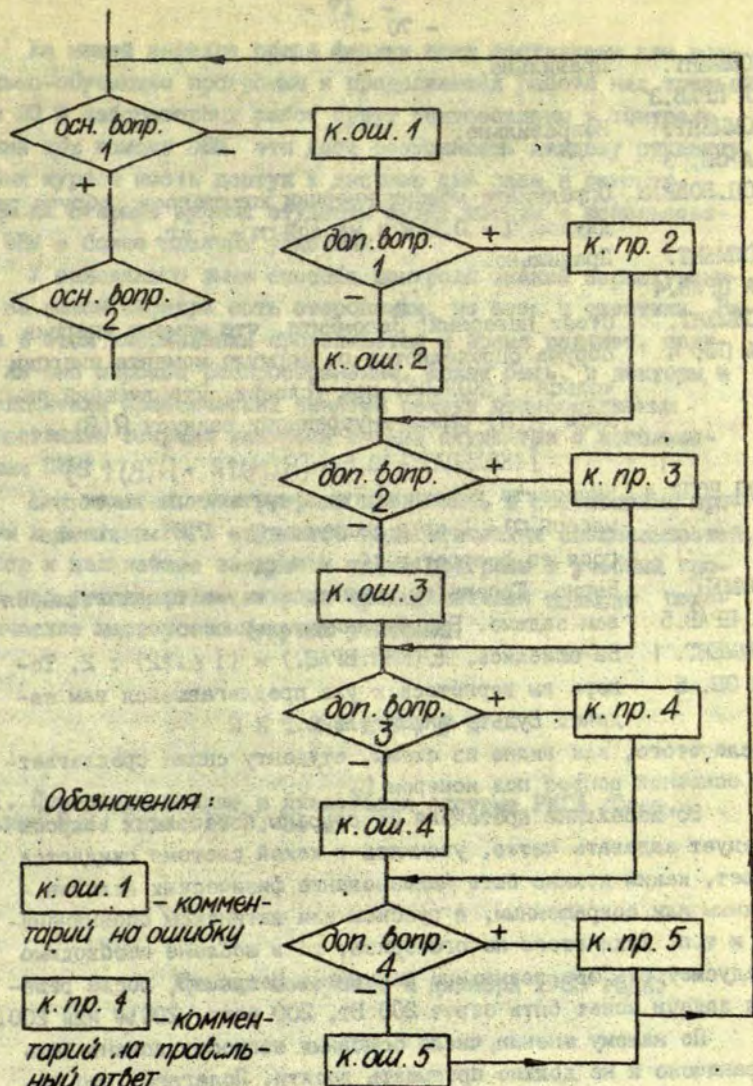
КОММЕНТ. Ответ верный.

НА ПРАВ.2

КОММЕНТ. Вы ошиблись.

НА ОШ. 2

ДОП.ВОПР.2 Вычислите угловую скорость ω обруча радиусом $r = 0,5$ м, катящегося со скоростью $v = 4$ м/с.



Обозначения:

к. ош. 1 — комментарий на ошибку

к. пр. 4 — комментарий на правильный ответ

Р и с. Структурная схема алгоритма программы контроля знаний.

- КОММЕНТ. Правильно.
НА ПРАВ.3
КОММЕНТ. Неправильно.
НА ОШ. 3
ДОП.ВОПР.3 Определите момент инерции катящегося обруча радиусом $r = 0,5$ м и массой $m = 1$ кг.

КОММЕНТ. Правильно.

НА ПРАВ.4

- КОММЕНТ. Ответ неверный! Запомните, что момент инерции обруча определяется по формуле момента инерции кольца I (КОЛЬЦА) при условии, что внешний радиус $R(H)$ равен внутреннему радиусу $R(B)$.

$$I (\text{КОЛЬЦА}) = 0,5m[R(H)^2 + R(B)^2]$$

- ДОП.ВОПР.4 Вычислите кинетическую энергию того же обруча массой $m = 1$ кг и радиусом $r = 0,5$ м, катящегося со скоростью $v = 4$ м/с.

КОММЕНТ. Верно. Теперь вы вернетесь к уже предлагавшейся вам задаче. Будьте внимательны.

- КОММЕНТ. Вы ошиблись, E (КИН.БРАЩ.) = $(I \cdot \omega^2) : 2$.
НА ОШ. 5 Теперь вы вернетесь к уже предлагавшейся вам задаче. Будьте внимательны.

После этого, как видно из схемы, студенту снова предлагается основной вопрос под номером 1.

Во избежание претензий со стороны отвечающих вопросы следует задавать четко, уточнять в какой системе ожидается ответ, каким должно быть наименование физических величин — полным или сокращенным, в русском или латинском обозначениях и т.п. Если этого не оговорить, то в шаблоне необходимо предусмотреть все возможные варианты. Например, после решения задачи может быть ответ 200 Вт, 200 ватт, 200 W или 200.

По нашему мнению, число основных вопросов должно быть ограничено и не должно превышать десяти. Полагаем, что покинув дисплей, студент запомнит эти вопросы и, если они его заинтересовали, продумает ответы на них еще раз.

Наблюдение за студентами, отвечающими на вопросы, предлагаемые на дисплее, позволяет нам сделать вывод, что часть студентов охотно общается с ЭВМ, подходит к делу серьезно и относится положительно к такому опыту.

На нашей кафедре общей физики пока составлены две контрольно-обучающие программы и продолжается работа над третьей. Если 20 % лабораторных работ будут подготовлены к контролю знаний при помощи ЭВМ, это даст возможность каждому студенту первых курсов иметь доступ к дисплею два раза в семестр. Тогда на старших курсах студенты будут готовы к использованию ЭВМ в более сложных условиях.

У описанного выше способа контроля знаний первокурсников на нашей кафедре есть сторонники, но есть и скептики. Работа в этом направлении продолжается и время покажет, получит ли оно широкое распространение. Может быть, и лекторы, и руководители практических занятий сочтут целесообразным осуществлять текущий контроль знаний студентов с использованием ЭВМ?

Нам кажется, что разумно привлекать к составлению программ в системе РИГА студентов педагогических специальностей. Разбор и дальнейшее внедрение таких программ в учебный процесс под руководством ведущих преподавателей повысит педагогическое мастерство будущих учителей.

Л И Т Е Р А Т У Р А

1. Программирование в диалоговой системе РИГА /Сост. Кузьмин Ю.Я. Рига, 1983. 31 с.

Статья поступила 2 декабря 1983 года.

ДИАЛОГОВЫЙ ЗАДАЧНИК ПО ФИЗИКЕ ТВЕРДОГО ТЕЛА

А.В.Афанасьева
НИИ физики твердого тела ЛГУ им. Петра Стучки

В настоящей работе рассматривается вопрос автоматизированного обучения методике решения задач качественного и количественного характера. Главная особенность качественной задачи [1,2] состоит в том, что в ней внимание обучаемого акцентируется на качественной стороне физических явлений, свойств тел, вещества, процессов и т.п. Следует отличать качественную задачу от вопроса по проверке формальных знаний (например, что называется ампером?). Цель последних - закрепить формальные знания обучаемых; ответы на такие вопросы в готовом виде имеются в учебнике, и обучаемый должен лишь вспомнить их. В качественной задаче задается такой вопрос, ответ на который обучаемый должен составить сам, на основе условия задачи и своих знаний по физике. Решаются качественные задачи путем логических умозаключений, базирующихся на законах физики.

Качественные задачи по физике способствуют углублению и закреплению знаний. Они служат средством проверки знаний и практических навыков обучаемых [3]. К количественным относятся задачи на построение графиков (например, построить траекторию движения; нахождение вектора γ), на получение количественных соотношений между физическими величинами. Особенность таких задач заключается в том, что внимание студента акцентируется на знании законов, формул, уравнений, постоянных и размерностей физических величин.

Решение физических задач является одной из форм контроля и закрепления знаний подучаемых студентом на лекциях и занятиях в автоматизированной системе обучения. В ряде

вузов эта форма контроля осуществляется с помощью ЭВМ. В отечественных и зарубежных вузах ЭВМ чаще всего используют для выдачи студентам различных вариантов задач для контрольных работ и домашних заданий [4-6]. Можно использовать ЭВМ как тренажер для решения задач [6,7]. В этом случае студенту многократно выдаются типовые задачи с меняющимися условиями. Например, в системе PLATO разработаны задачки-тренажеры по механике, математике и другим областям знания [8].

Новым направлением, получившим развитие сравнительно недавно является использование ЭВМ для обучения решению задач. Пройдя такое обучение, студенты приобретают и закрепляют полученные знания в конкретной области физики.

Средства многопультной диалоговой системы РИГА [9] позволяют создавать комплексы из текстовых и программных единиц информации. При этом существует возможность в нужных местах текста производить вызов программ. Сами тексты могут включать диалог на текстовом уровне. Указанные возможности системы легко могут быть использованы для создания диалоговых задачников качественного и количественного характера. В настоящей работе предлагается диалоговый задачник по курсу физики полупроводников.

Известно, что в курсе физики полупроводников рассматривается ряд фундаментальных проблем: основы зонной теории полупроводников, колебания атомов кристаллической решетки, статистика электронов и дырок в полупроводниках, механизм рассеяния носителей заряда в кристаллические явления, поверхностные явления и люминесценция в полупроводниках, диффузия и дрейф неравновесных носителей заряда и др. [10].

Среди них можно выделить наиболее важные: статистика электронов и дырок в полупроводниках и кинетические явления в полупроводниках.

При изучении этих вопросов необходимой практической основой является решение физических задач. Оно учит анализировать изучаемые явления, выделять обуславливающие их основные факторы. Решение задач в известной мере приближается к модели научного физического исследования [11].

Предлагается использовать следующие три типа задач количественного характера:

1. Задача на эффект Холла.
2. Задача на вычисление концентрации акцепторов легированного полупроводника.
3. Задача на вычисление концентрации носителей заряда в полупроводнике с конкретно заданными геометрическими размерами.

Программированный обучающий задачник, предлагаемый в данной работе, охватывает три указанных выше типа задач, к каждому из которых представлен одной задачей.

Первая связана с эффектом Холла - одним из главномагнитных эффектов. Его применяют для определения типа носителей заряда, измерения сил магнитных полей. Для решения такой задачи необходимо иметь представление о самом эффекте, знать фактор Холла и его зависимость от механизма рассеяния.

Вторая задача относится к одной из важных тем по физике полупроводников: статистика э. электронов и дырок в полупроводниках.

Третья является типичной задачей из группы задач по проводимости полупроводников.

Процесс обучения методике решения задач выглядит следующим образом. Вначале система предьявляет правила решения задачи. Например:

Правила:

1. Полученный ответ введите в машину с точностью до 0,5 %.
 2. Десятичное основание обозначайте буквой "E". Например, три целых и пять десятых на десять в шестой степени введите: 3,5E+6.
 3. Единицы измерения не вводите.
 4. Пользуйтесь калькулятором ЭВМ.
- Затем генерируется сама задача:
- З а д а ч а.** (введите ответ, либо <·>, либо <=>).
Вычислить концентрацию носителей заряда ($I/cm^2 \cdot s$) данного полупроводника, если рассеяние происходит на

акустических фонах. Подвижность Холла равна 2400. ($\text{см}^2/\text{Вс}$), а удельная проводимость 16,0 ($\text{I}/\text{Ом}\cdot\text{см}$).

При успешном решении задачи система выдает следующую, в противном случае она начинает процесс обучения: после неудачного решения предлагается подсказка и снова предьявляется эта же задача. В данном случае это выглядит следующим образом:

Вы ошиблись.

Для верного решения необходимо вспомнить, что фактор Холла (R) зависит от типа рассеяния. В данном случае рассеяние происходит на акустических фонах, следовательно, $R = I, I\bar{6}$.

Придется повторить задачу.

Число предьявляемых подсказок зависит от степени сложности задачи. В случае неудачного решения за последний подсказкой выдается полное решение задачи.

Для того, чтобы можно было проследить за количеством обращений к программированному "Задачнику по физике полупроводников", первая строка имеет вид:

4-й раз используется K4I0I.

Автора задачника также может заинтересовать количество обращений пользователей к разным местам задачника. Например, к подсказке. С этой целью нужные места задачника помечаются счетчиком $\#A000000$, который увеличивается на 1 всякий раз, как пользователь попадает на это место задачника.

Обращение к задаче, написанной на ФОРТРАНЕ, и анализ результатов организованы следующим образом.

1. $\#P4I03$

2. $\#A00004$

3. $\#K 0_{I4I}-I$

4. $\#A00000$

5. Молодец!

6. $\#:2$

7. $\#47, I, 0$

8. Что же вы! Вспомните, что...

В первой строке происходит вызов программы-задачи и ее решение студентом. После завершения задачи студент снова попадает в текстовую часть задачника, т.е. на вторую строку. При этом программа-задача сообщает системе результат решения (решено/нерешено). Во второй строке зафиксирована очередная попытка решения задачи (счетчик \mathcal{N} увеличится на 1), а затем произойдет условный переход (третья строка) на четвертую строку, если задача была решена верно, либо на четвертую строку после \mathcal{Y} , если она не решена.

При верном решении будет увеличен счетчик всех верных решений этой задачи (\mathcal{N} в четвертой строке) и выдан текст: "Молодец!". После этого текста, при попытке студента пойти дальше, система будет искать строку, начинающуюся символами "2", так как инструкция $\mathcal{Z}:2$ означает поиск такой строки, которая начнет новую задачу.

Если задача не была решена, то инструкция \mathcal{Y} перейдет на четыре строки ниже, т.е. к $\mathcal{Z}+7, 1, \mathcal{B}$, что зафиксирует ошибку в ответе конкретного студента. Эта ошибка занесется в седьмой (из 50 возможных) счетчик ошибки. По счетчику система автоматически формирует оценку знаний студента и, кроме того, она указывает на конкретный вопрос, с которым не справился студент.

Увеличив счетчик ошибок, система выдает текст подсказки: "Что же Вы! Вспомните, что...". Таким образом образуется диалог со студентом, во время которого система выдает текст, запускает программы задач, узнает результат, выдает подсказку и снова запускает программу-задачу.

Сами задачи реализованы в виде двух программных моделей:

1) подпрограмма, осуществляющая задание условий задачи, выдчу текста и расчет ответа;

2) программа, принимающая ответ студента, сравнивающая его с расчетным, подсчитывающая число попыток решений и сообщаящая системе РИГА, решена или нет задача. Эта же программа работает в режиме калькулятора, что облегчает студенту проведение расчетов. В целом, программа составля-

ет около 90 % всего алгоритма задачи, что облегчает автору программирование задач. Его работа сводится к созданию подпрограмм типа:

```
0001 FTN4
0002 C# АВТОР: АФАНАСЬЕВА А.В.  ДАТА: 2/3/83
0003 C# НАЗНАЧЕНИЕ: ЗАДАЧА НА ЭФФЕКТ ХОЛЛА
0004 C# ЗАГРУЗИТЬ (RE,RTASK)
0005     SUBROUTINE P1S(LUV,UN)
0006     P1=RND(2000.,3000.,100.,1)
0007     P2=RND(10.,20.,1.,1)
0008     P3=1.18
0009     P4=1.6E-19
0010 C****ВЫДАЧА ТЕКСТА ЗАДАЧИ:
0011 101 CONTINUE
0012     WRITE(LUV,102)P1,P2
0013 102  FORMAT("ЗАДАЧА (ВВЕДИТЕ ОТВЕТ, ЖИВО (.), ЖИВО(=)"/
0014     "-----"/
0015     "  ВЫЧИСЛИТЬ КОНЦЕНТРАЦИЮ НОСИТЕЛЕЙ ЗАРЯДА (1/СМ**3)  "/
0016     "  ДАННОГО ПОЛУПРОВОДНИКА , ЕСЛИ РАССЕЯНИЕ ПРОИСХОДИТ НА"/
0017     "  АКУСТИЧЕСКИХ ФОНОНАХ. ПСАВИЗНОСТЬ ХОЛЛА РАВНА ",F5.1,""/
0018     "  (СМ**2/В*С) , А УДЕЛЬНАЯ ПРОВОДИМОСТЬ ",F5.1," (1/ОМ*СМ).")
0019 C#РАСЧЕТ ОТВЕТА:
0020     DN=(P3*P2)/(P4*P1)
0021     RETURN
0022     END
```

Здесь приведен листинг подпрограммы для задачи на эффект Холла. Строка № 6 задает подвижность носителей случайным образом, но в пределах от 2000 до 3000 с шагом 100 см²/В·с. Аналогично задается удельная проводимость. В строке № 20 рассчитывается ответ.

В результате выполненной работы показана возможность создания программированных задачникoв качественного и количественного характера, а также найдены некоторые типовые решения программирования задач, что может быть полезно при создании задачникoв на другие темы.

ЛИТЕРАТУРА

- I. Тульчинский М.Е. Качественные задачи по физике. М., 1976. 112 с.
2. Тульчинский М.Е. Занимательные задачи - парадоксы и софизмы по физике. М., 1971. 128 с.
3. Тульчинский М.Е. Качественные задачи по физике. Калуга, 1969. 108 с.
4. ЭВМ в курсе общей физики / Под ред. Матвеева А.И. М., 1982. 232 с.
5. Smith James H. Problem Sets in Elementary Classical Mechanics. - Urbana, Illinois, 1976, 38 p.
6. Bitzer Donald. The Wide World of Computer - Based Education. - Urbana. Illinois, 1976. 43 p.
7. Талызина И.Ф., Габай Г.В. Пути и возможности автоматизации учебного процесса. М., 1977. 57 с.
8. Nibello Louis V. Keith Bailey. Catalogue of PLATO Mathematics Lessons. - Urbana. Illinois, 1975. 122 p.
9. Программирование в диалоговой системе РИГА. / Сост. Кузьмин Д.Я. Рига, 1983. 31 с.
10. Шалимова К.В. Физика полупроводников. М., 1976. 416 с.
- II. Новодворская Е.М., Дмитриев Э.М. Методика проведения упражнений по физике во втузе. М., 1981. 318 с.

Статья поступила 20 февраля 1984 года.

ПОИСК ФАЙЛОВ НА ДИСКЕ

А.Б.Езепс

НИИ физики твердого тела ЛГУ им. Петра Стучки

Эффективность работы любой ЭВМ определяется прежде всего ее операционной системой и набором сервисных программ, позволяющих быстро и просто получать информацию о необходимых файлах. Мини-ЭВМ типа М-6000, СМ-1, СМ-2, СОУ-1 имеют специальную систему управления файлами (СУФ), дающую пользователям ЭВМ возможность производить операции с файлами: создавать, уничтожать, переименовывать, копировать и выводить файлы, а также запрашивать каталоги файлов. Эти операции довольно удобны при работе в однопултовом режиме и при не очень больших количествах файлов [1,2].

Системы коллективного пользования, имеющие много пультов для одновременной работы пользователей с ЭВМ, предъявляют новые требования к программному обеспечению. В частности, должны быть средства удобного представления общей информации по файлам, поиска файлов по отдельным элементам имени, частичного или полного вывода или уничтожения всех отобранных по отдельным признакам файлов.

В имеющейся СУФ М-6000, СМ-1, СМ-2, СОУ-1 указанные выше требования выполнимы только с большими затратами ресурсов. Так, вывод первых строк файлов по отдельным элементам их имени можно произвести лишь в несколько приемов. На первой стадии нужно найти эти файлы, распечатав каталог. На второй стадии - провести распечатку каждого файла в отдельности. СУФ не имеет возможности скатого представления каталогов файлов. Как правило, при распечатке выдается несколько страниц. Нет возможности упорядочения каталогов по алфавиту, что очень удобно для представления пользователю. Нет возможности найти в системе файл, если, например, известно, что 2-й и 4-й символы их имени - цифры.

Все это привело автора настоящей статьи к разработке специальной программы ПОИСК ФАЙЛОВ НА ДИСКЕ.

Программа работает на трех уровнях. На первом уровне можно получить общие сведения о запрошенном диске: узнать количество дорожек для имен файлов, первую дорожку имен файлов, номер первой доступной дорожки и сектора для содержимого файлов, количество неупакованных файлов и количество экстенгов (продолжений файлов).

Чтобы получить эту информацию, пользователь должен знать номер диска, на котором будет произведен поиск. Номер принимает значения: 1, 3, 21, Программа запросит номер диска, после чего выдаст сведения первого уровня и выйдет на второй уровень. На первом уровне программа выполняет директивы пользователя.

На втором уровне программа выдает сведения об именах файлов, которые хранятся на диске, запрошенном пользователем. В зависимости от заданных директив можно вывести каталог имен и типов файлов на диске и на печатающее устройство, проделать поиск файлов по шаблону, при выводе каталога не распечатывать имена экстенгов, перед выводом упорядочить файлы по их именам.

На третьем уровне выполняется директива распечатки первых строк файлов. Здесь ведется работа с содержимым файлов. Пользователь может указать, с какого файла и сколько строк выводить, с нумерацией или без нумерации строк, плотный вывод или через строчку, выводить каждый файл на новом листе или все файлы подряд. Можно удалить файлы, которые соответствуют заданному шаблону. Для получения сведений о файлах на другом диске, нужно ввести директиву НАЧАЛО и задать номер диска.

Далее следует описание директив пользователя.

ЭКС - (экстенги). По этой директиве удаляются из списка имена экстенгов. Экстенги появляются следующим образом: если при выводе информации на диск окажется, что размер файла исчерпан, СУФ автоматически расширит

ет файл, создавая экстенды (новые группы смежных дорожек). На том же диске, где и главная часть файла, создаются метки экстендов файла. Параметры, указанные при создании главной части файла (имя файла и др.), автоматически переписываются в метки экстендов файла. Чтобы список имен файлов плотнее выводился, директива ЭКС удаляет имена экстендов.

ШАБ - (шаблон). Производит поиск имен и типов файлов по заданному пользователем шаблону. Максимальная длина шаблона - семь позиций: шесть позиций для имен файлов и одна для типов.

Каждая позиция может содержать латинскую букву, любую цифру, пробел, букву В или букву Ц. Пробел означает, что в этой позиции может быть любой символ, В - что в этой позиции может быть только буква и Ц - только цифра. Можно задавать неполный шаблон - недостающие позиции автоматически дополняются справа пробелами. Например, шаблон <F41> воспринимается программой как <F41 > и этот шаблон найдет имена всех файлов, которые начинаются с F41. Шаблон <F41 3> найдет все имена файлов третьего типа на языке ФОРТРАН пользователя 4I (если принята такая кодировка файлов). Шаблон <39> укажет все имена файлов пользователя 39, а шаблон ЦЦ сообщит все имена, содержащие цифры во второй и третьей позициях. Шаблон ББББ найдет имена, в первых четырех позициях которых содержатся буквы.

АЛФ - (алфавит). Упорядочивает список имен файлов по алфавиту.

КАТ - (каталог). Выдает на пульт пользователя или на печатающее устройство сжатый список имен и типов файлов, по 8 имен в строке. Вывод происходит кадрами. Директива КАТ имеет свой диалог с пользователем: пользователь сообщает, сколько еще имен имеется для распечатки и предлагается ввести "+" для вызова следующего кадра, "-" для вывода предыдущего кадра и "Копия"

для вывода кадра на печатающее устройство. После слова "Копия" можно ввести комментарии (до 72 символов), которые будут распечатаны как заглавие кадра. Если ввести ".", произойдет выход из директивы KAT на второй уровень.

ФАЙ - (файлы). Директива производит удаление файлов и вывод первых строк файлов; работает она не с именами файлов, а с содержимым файлов и имеет свой внутренний диалог, который образует третий уровень. Директива имеет сложную форму исполнения. После ее ввода ЭВМ начинает диалог с пользователем. Вводом ответов на вопросы ЭВМ пользователь может задать желаемый формат для вывода файлов. Предусмотрена также возможность удаления файлов. Далее следует описание диалога.

"Будем удалять файлы?" - ответ "да" приведет в конце диалога к удалению всех указанных файлов. При ответе "нет" будут выполнены операции вывода первых строк.

"Номер первого выводимого файла?" - нужно ввести порядковый номер № I первого подлежащего выводу файла согласно каталогу. Ввод числа $\leq I$ вызовет вывод с начала. При № I) фактического числа файлов - вывод последнего.

"Номер последнего выводимого файла?" - нужно ввести номер № 2 последнего файла. Ввод числа $> =$ фактического количества файлов вызовет выдачу до конца. При вводе №2 = №1 - вывод первого.

"Число выводимых строк?" - нужно ввести число строк, которые будут выведены для каждого запрошенного на вывод файла. Ввод числа $\leq I$ выдаст первую строку. Ввод числа большего количества фактической длины файла - выдаст весь файл.

"Печатать?" - ответ "да" приведет к выдаче файлов на печать, иначе - на дисплей.

"Пропускать строку?" - ответ "да" вызовет вывод файлов на печать через строку, иначе - нормальный вывод. При выводе на дисплей этот запрос не выдается.

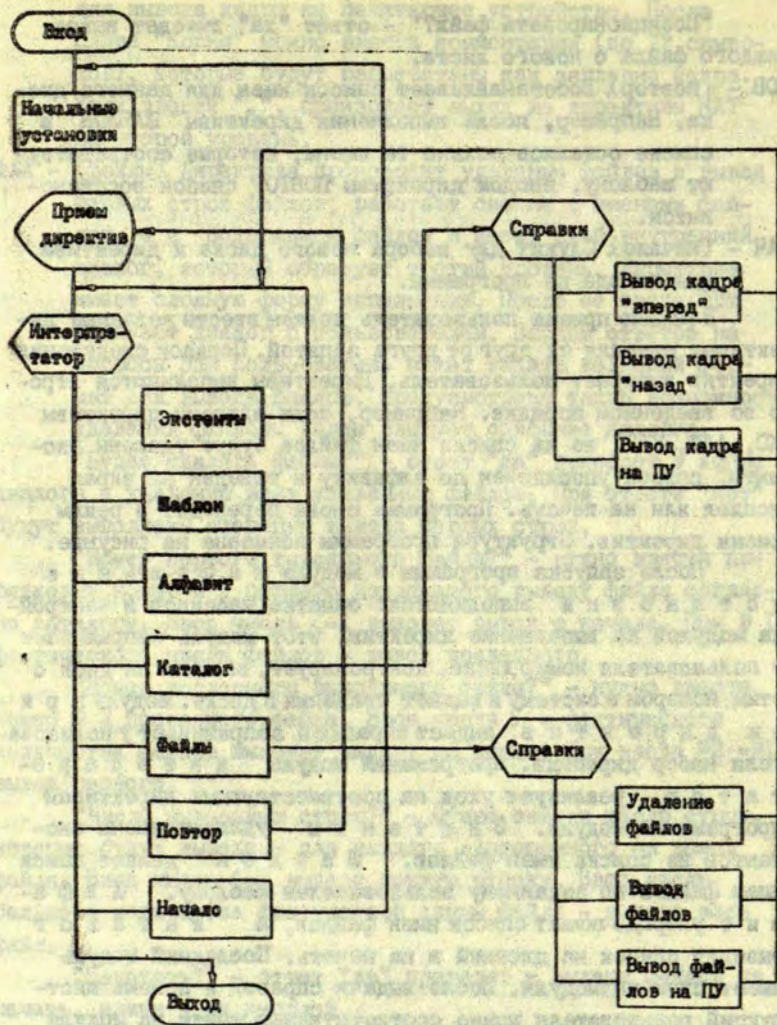
"Позиционировать файл?" - ответ "да" вызовет вывод каждого файла с нового листа.

ПОВ - (повтор). Восстанавливает список имен для данного диска. Например, после выполнения директивы ШАБЛОН в списке остались только те имена, которые соответствуют шаблону, вводом директивы ПОВТОР список восстановится.

НАЧ - (начало). Служит для выбора нового диска и директива для выхода из программы.

В режиме приема пользователь должен ввести желаемые директивы, отделяя их друг от друга запятой. Порядок следования директив выбирает пользователь. Директивы выполняются строго во введенном порядке. Например, если введены директивы ЭКС, АЛФ, КАТ, то из списка имен файлов будут удалены экстененты, список упорядочен по алфавиту и выведен на экран дисплея или на печать. Программа снова переходит в режим приема директив. Структура программы показана на рисунке.

После запуска программы в модуле начальные установки выполняются: очистка массивов и настройка модулей на выполнение директив, этот модуль запрашивает у пользователя номер диска, контролирует, включен ли диск с этим номером в систему и выдает сведения о диске. Модуль приема директив выдает справки и запрашивает у пользователя набор директив. Программный модуль интерпретатор реализует уход на соответствующие директивам программные модули. Экстененты удаляет имена экстенентов из списка имен файлов. Шаблон делает поиск имен файлов по заданному пользователем шаблону. Алфавит упорядочивает список имен файлов, а каталог выводит список на дисплей и на печать. Последний модуль имеет свои подмодули. После выдачи справок и приема инструкций пользователя можно соответственно выйти на модули вперед кадра вперед, вывод кадра назад и вывод кадра на печатающее устройство.



Р и с. Структура программы.

Модуль файлы удаляет указанные в диалоге с пользователем файлы или выводит первые строки файлов. Форма вывода указывается пользователем. Модуль повтор восстанавливает список имен в начальном виде, а программный модуль начало позволяет выбрать другой диск.

Л И Т Е Р А Т У Р А

1. Дисковая операционная система реального времени. Программное обеспечение М-6000 АСВТ-М. Руководство по пользованию, ч.1. 2-я ред. Бм., Бг. 249 с.

2. АСВТ-М. Программное обеспечение М-6000. Система управления файлами для дисковой операционной системы реального времени. Краткое описание и руководство по пользованию. Бм., Бг. 157 с.

Статья поступила 10 января 1984 года.

КОНТРОЛЬ ДИСПЛЕЕВ В МНОГОПУЛЬТОВОЙ СИСТЕМЕ

Я.Я.Гутанс
НИИ физики твердого тела ЛГУ им. Петра Стучки

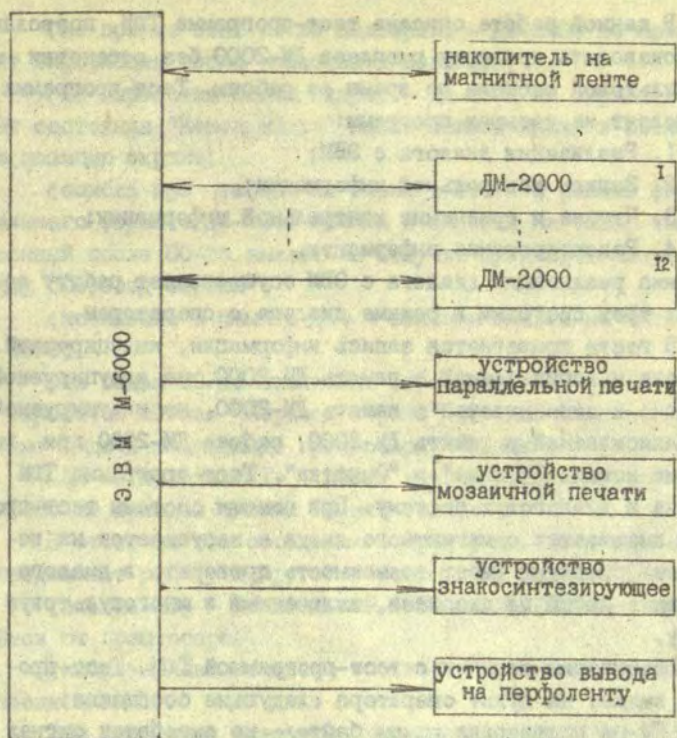
В настоящее время стали популярны диалоговые системы коллективного пользования. Многопультовые диалоговые системы позволяют автоматизировать сбор и обработку данных, обучение пользователей ЭВМ работе с операционными системами, трансляторами и техническими средствами. При работе в многопультовой системе возникает необходимость без остановки системы проверять внешние устройства ввода-вывода информации (дисплеи, печатающие устройства и т.д.) (рис.). При этом в случае нескольких дисплеев особо важно тестировать неисправный дисплей, который может находиться в следующих аварийных ситуациях, указанных в таблице.

Т а б л и ц а

Возможные аварийные ситуации дисплея

Ситуации	Автономный режим проверки	Динамический режим проверки	В многопультовой системе
S.1.	да	нет	нет
S.2.	нет	да	да
S.3.	нет	да	да

В таблице приняты следующие обозначения ситуаций:
 S.1. - неисправность индикации и пульта ввода информации,
 S.2. - неисправность набора и редактирования информации,
 S.3. - неисправность обмена информации с ЭВМ.



Р и с. Внешние устройства ввода-вывода информации ЭВМ М-6000.

Обнаружение этих ситуаций производится с помощью специальных тест-программ. Ввод информации с пульта дисплея ДМ-2000 в память и индицирование этой информации на экран проверяется в автономном режиме без участия ЭВМ. Тест, который поставляется заводом вместе с дисплеем ДМ-2000 позволяет производить проверку дисплея только в динамическом режиме. При попытке использования в многопультной системе она останавливается, прерывается работа всех пользователей.

В данной работе описана тест-программа TDM, позволяющая производить проверку дисплеев ДМ-2000 без остановки многопультной системы во время ее работы. Тест-программа TDM состоит из четырех программ:

1. Реализация диалога с ЭВМ;
2. Запись контрольной информации;
3. Чтение и сравнение контрольной информации;
4. Редактирование информации.

Программа реализации диалога с ЭВМ осуществляет работу остальных трех программ в режиме диалога с оператором.

В тесте проверяется запись информации, индуцируемой на экране и записываемой в память ДМ-2000, не индуцируемой на экране и записываемой в память ДМ-2000, не индуцируемой и не записываемой в память ДМ-2000, работа ДМ-2000 при выполнении команд "Чтение" и "Очистка". Тест-программа TDM включена в диалоговую систему. При помощи системы тест-программа вызывается с магнитного диска и запускается на исполнение. Оператор имеет возможность проверить в диалоговом режиме любой из дисплеев, включенный в многопультную систему.

Рассмотрим вариант с тест-программой TDM. Тест-программа выдает на пульт оператора следующие сообщения:

< ДМ не подтвердил прием байта > - не выработан сигнал готовности от передающей карты;

< не выработана готовность в режиме приема > - не выработан сигнал готовности от приемной карты;

< сигнал ОШ-Т при выдаче на ПР. > - ошибка по паритету в информации, принятой от ДМ-2000, или наличие сигнала "Ошибка";

< ошибка при приеме от процессора > - ДМ-2000 выработал байт состояния в ответ на байт, выданный из процессора;

< КОП-Т сопровождает не последний байт > - последний байт от ДМ-2000 не сопровождается сигналом "Конец операции";

< ошибочный начальный байт > - ДМ-2000 выработал ошибочный начальный байт;

⟨ не принят байт после повторов ⟩ - ДМ-2000 не принял байт информации после нескольких повторов;

⟨ не выработан конец кадра ⟩ - ДМ-2000 не выработал байт состояния "Конец кадра" после записи знака в последнюю позицию экрана;

⟨ ошибка при записи на экран в ФФ ⟩ - в режиме фиксированного формата ДМ-2000 принял и записал на экран знак, выданный после 80-го знака, не получив предварительную команду "Перевод строки";

⟨ искажение в памяти ДМ ⟩ - искажен символ на экране ДМ-2000;

⟨ не выдан ПС по окончании строки при ЧТ. ⟩ - ДМ-2000 не выработал сигнал "Перевод строки" по окончании строки в режиме выдачи по команде процессора "Чтение";

⟨ ДМ не прислал КТ ⟩ - после выдачи текста на экран ДМ-2000 не прислал сигнал "Конец текста".

В ответ на эти сообщения оператор вызывает инженера и продолжает работу с другого, исправного дисплея ДМ-2000.

⟨ ошибка 23 ⟩ - нелегальный байт от ДМ-2000 при выдаче байтов от процессора;

⟨ неопознанная ошибка ⟩ - ошибка, не принадлежащая к вышеописанным.

Эти сообщения означают, что дисплей ДМ-2000 не подключен к ЭВМ. Если тест не обнаружил ошибок при проверке устройства, на пульт оператора выдается сообщение:

тест закончил работу.

ДМ-2000 исправен.

Сообщение о характере неисправностей дисплея ДМ-2000 выдается на экран дисплея оператора, при этом все остальные дисплеи работоспособны, а тест-программа не мешает работе других пользователей.

Предполагается создание в многопультной диалоговой системе тест-программ для тестирования других устройств ввода-вывода, включенных в вычислительный комплекс М-6000.

Статья поступила 6 февраля 1984 года.

СВЯЗЬ ДИСПЛЕЯ ИБМЗ-00-013 С МИНИ-ЭВМ

И.А.Гужа, А.Я.Спилва
НИИ физики твердого тела ЛГУ им.Петра Стучки

В в е д е н и е

В системах коллективного пользования, учитывая разнообразные запросы пользователей, необходимо применять терминалы с широкими функциональными возможностями. Таким терминалом является дисплей типа ИБМЗ-00-013 [1]. Дисплей обеспечивает хранение 48 строк текста по 30 символов в строке с одновременным отображением на экране не менее 1920 символов при количестве 24 строки по 80 символов в каждой; отображает также одну служебную строку для индикации режимов его работы. Дисплей формирует изображение 192 символов (160 символов прописных и строчных букв русского и латинского алфавитов, арабских цифр, специальных символов и 32 служебных символа).

Выбор схемы интерфейса

Для связи мини-ЭВМ с дисплеями часто используют модули быстрой передачи данных (МБПД) [2], серьезным недостатком которых является то, что они состоят из двух плат: приемника и передатчика и, соответственно, занимают два места в сопряжении мини-ЭВМ. Кроме того, для стыковки МБПД с дисплеями ИБМЗ-00-013 необходимо дополнительное согласующее устройство.

Рассмотрим возможности связи с ЭВМ, заложенные в самом дисплее. Связь дисплея с ЭВМ может быть осуществлена через сопряжение типа "Стык-2", либо через токовую петлю. Сопря-

жение типа "Стык-2", кроме информации, позволяет использовать еще и специальные сигналы, которые оповещают о работе аппаратуры и линии связи. Недостатком данного сопряжения является довольно большое число соединительных проводов и необходимость иметь два дополнительных источника питания (+12 В, -6 В). При использовании токовой петли для связи необходимы только две витые пары проводов.

Из вышеизложенного следует, что целесообразным является разработка специального интерфейса, позволяющего принимать и передавать информацию по токовой петле. Подобного рода интерфейсы в последнее время разрабатывают на больших интегральных схемах (БИС) [3].

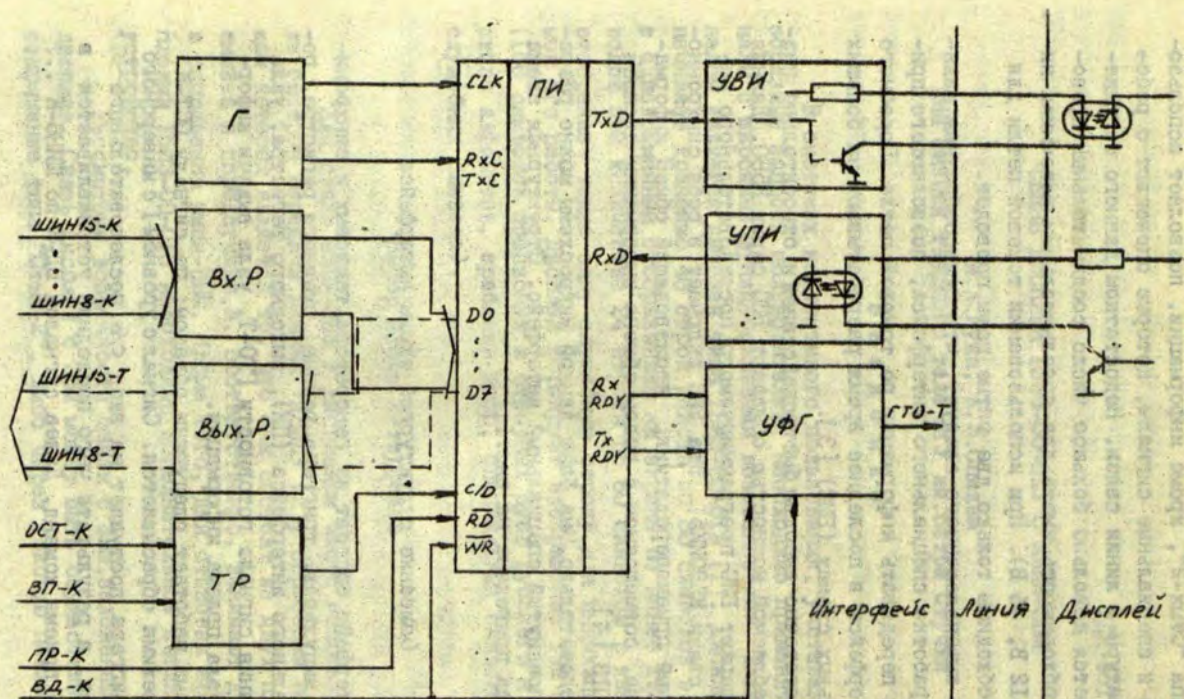
Достаточно высоким быстродействием и относительно низкой потребляемой мощностью обладают БИС серии К580. Данная серия содержит БИС программируемого последовательного интерфейса связи К580ИК51. Эта БИС включает в себя синхронно-асинхронный приемопередатчик, производящий обмен информации в виде, совмещаемом с видом обмена информации дисплея ИИЭ-00-013 [4].

Следовательно, на базе данной микросхемы можно разработать упомянутый специальный интерфейс. Структурная схема интерфейса приведена на рис.1.

Описание структурной схемы интерфейса

Интерфейс состоит из генератора тактовых и синхронизирующих импульсов, триггера режима, входного регистра, последовательного интерфейса (ПИ), выходного регистра, узла формирования сигнала готовности ГТО-Т, узла выдачи информации и узла приема информации.

Схема работает следующим образом: по сигналу ОСТ-К триггер режима сбрасывается. Сигнал с уровнем I с инверсного выхода триггера поступает на вход с/о последовательного интерфейса, в результате чего последний устанавливается в состояние приема команд. Далее сигналом ВД-К по ШИНВ-К ...



Р и с. 1. Структурная схема интерфейса связи.

Г - генератор тактовых и синхро-импульсов, ВХ-Р - входной регистр, ВЫХ.Р - выходной регистр, ПИ - последовательный интерфейс, ТР - триггер режима, УПИ - узел приема информации, УВИ - узел выдачи информации, УФГ - узел формирования сигнала готовности.

ШИН15-К через входной регистр записываются управляющие слова. Для нормальной работы ПИ необходимо записать два управляющих слова. Первое слово определяет скорость обмена информацией и формат информационных слов. Оно имеет значение 316(8). Второе управляющее слово определяет режим работы ПИ и имеет значение 67(8). После этого сигналом ВП-К триггер режима устанавливается в значение 1. На вход c/\bar{c} поступает 0 с его инверсного выхода. С этого момента интерфейс может работать в режимах выдачи и приема информации.

В режиме приема информация через входной регистр поступает на вход $R \times D$ ПИ, который в ответ выставляет сигнал готовности $R \times R \bar{D}$ равный 1. Последний через узел формирования готовности поступает на ЭВМ в качестве сигнала ГТО-Т. ЭВМ, получив ГТО-Т = 1, вырабатывает сигнал приема ПР-К, который поступает на выходной регистр в качестве строба и на вход $\bar{R} \bar{D}$ ПИ.

В режиме выдачи информации, последняя с ШИНВК ... ШИН15К, стробируемая сигналом ВД-К, поступает через входной регистр на входы $D_0 \dots D_7$ ПИ. Одновременно сигнал $\bar{w} \bar{k}$ устанавливается равным 0, чем обуславливается преобразование выданной информации в последовательный код, который с выхода $T \times D$ через выходной регистр поступает на линию.

Усовершенствование блока клавиатуры

Клавиатура дисплея I5ИЭ-00-013 имеет некоторые недостатки: если, набирая информацию в верхнем регистре, необходимо ввести один символ из нижнего регистра, надо нажать клавиши НР и нужный символ, затем для перехода в верхний регистр надо нажать еще клавишу ВР. При работе на дисплее наиболее часто используют строчные буквы латинского алфавита, цифры и спецсимволы. При этом выявляется еще один недостаток клавиатуры - необходимость часто переключать регистры, потому что строчные буквы латинского алфавита находятся на верхнем регистре, а спецсимволы - на нижнем. Размещение клавиш блока клавиатуры показано на рис.2.

+;	!;	"	#	¤	%	&	'	()	0	=		ТАБ.	ГТ
И	Л	С	У	К	Е	Н	Г	Ш	Э	Х	:	ПС*	БК
С	В	У	В	А	Р	Р	О	Л	Д	Ж	Э	>	3Б
ВР	Ж	Г	С	И	И	Т	Д	В	Ю	>	/	?	НР
РУС												ЛАТ	

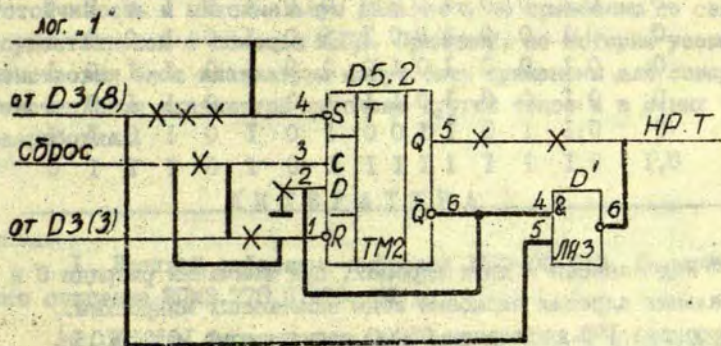
7	8	9
4	5	6
1	2	3
0	.	:

Р и с. 2. Изменения функциональных значений клавиш.

Как видно из рисунка, цифры находятся и на основной клавиатуре (вместе с спецсимволами), и на дополнительной. Для их ввода можно использовать только дополнительную клавиатуру, а соответствующие клавиши основной клавиатуры использовать для ввода спецсимволов независимо от включенного регистра. Поэтому на основной клавиатуре цифры заменены спецсимволами из нижнего регистра. Произведены также некоторые другие изменения функциональных значений клавиш, способствующие созданию удобства для работы на дисплее.

Все изменения показаны на рис.2 утолщенными линиями. Для осуществления этих изменений заменены имеющиеся в блоке ППЗУ, в которых хранятся коды символов клавиш, на ППЗУ с информацией согласно таблице.

Для изменения алгоритма переключения нижнего и верхнего регистров изменена схема переключения регистров (рис.3).



Р и с. 3. Изменения в узле управления блока клавиатуры.

Т а б л и ц а

Измененные разряды микросхем ППЗУ DS1 и DS2

Новое обозн. клавиши	А д р е с								К о д							
	HP-T	ЛAT-T	32	16	8	4	2	1	DS1				DS2			
	7	6	5	4	3	2	1	0	01	02	03	04	01	02	03	04
+	0	0,1*	0	0	0	0	0	0	0	0	1	0	1	0	1	1
!	0	0,1	0	0	0	0	0	1	0	0	1	0	0	0	0	1
"	0	0,1	0	0	0	0	1	0	0	0	1	0	0	0	1	0
#	0	0,1	0	0	0	0	1	1	1	0	1	0	0	0	1	1
\$	0	0,1	0	0	0	1	0	0	0	0	1	0	0	1	0	0
%	0	0,1	0	0	0	1	0	1	1	0	1	0	0	1	0	1
&	0	0,1	0	0	0	1	1	0	1	0	1	0	0	1	1	0
'	0	0,1	0	0	0	1	1	1	0	0	1	0	0	1	1	1
(0	0,1	0	0	1	0	0	0	0	0	1	0	1	0	0	0
)	0	0,1	0	0	1	0	0	1	1	0	1	0	1	0	0	1
-	0	0,1	0	0	1	0	1	0	0	0	1	0	1	1	0	1
=	0	0,1	0	0	1	0	1	1	1	0	1	0	1	1	0	1
;	0	0,1	1	0	1	1	0	0	1	0	1	0	1	0	1	1
.	0,1	0,1	1	1	1	1	1	1	0	0	1	0	1	1	1	0

* Код записан в двух адресах, при значении разряда 0 и 1.

В остальных адресах записаны коды замененных микросхем.

В микросхеме DS3 по адресу 01000 записан код 10101010 (#).

Вновь введенные элементы и связи показаны утолщенными линиями, а прерванные связи зачеркнуты. Измененная схема работает следующим образом: при поступлении сигнала "Сброс" триггер D5.2 устанавливается в состояние логического 0. На выходе 6 элемента D' тоже появляется нулевой уровень и в блоке клавиатуры включается верхний регистр. Так как триггер D5.2 включен по счетной схеме, то каждый импульс сигнала ВР изменяет его состояние на противоположное и, следовательно, меняет верхний и нижний регистры после каждого нажатия на клавишу ВР.

При поступлении сигнала НР (нажата клавиша НР), на выходе 6 элемента D' появится уровень логической 1, и установится нижний регистр. Такое состояние сохранится до отпущения клавиши НР.

Разработанный интерфейс может найти применение также и для связи двух ЭВМ. Связь имеет более повышенную помехоустойчивость и максимальную дальность по сравнению со связью, осуществляемой с помощью МВЦД. Принципы, по которым усовершенствован блок клавиатуры могут быть применены для совершенствования клавиатуры дисплеев других типов и в новых разработках.

Л И Т Е Р А Т У Р А

1. Дисплей алфавитно-цифровой И5ИЗ-00-013. Техническое описание ШПМЗ.778.012Т0. 61 с.
2. Модуль быстрой передачи данных А723-1 2.000.002РЭ. 52 с.
3. Intel 8080 Microcomputer Systems. Users Manual. Santa Clara, California, 1975.
4. Ķīpa A. Elektroniskie skaitļotāji. Rīga, 1983. 236 lpp.

Статья поступила 17 февраля 1984 года.

СО Д Е Р Ж А Н И Е

1.	ВОСТРЫ М., КИРПА А.Н., ГВОЗДЕВ С.В., ЛЫСОВ Б.И., ПОЗДЕЕВ В.А. Развитие дисковой операционной системы реального времени для мини-ЭВМ типа М-6000.....	3
2.	ПАВЛИК О.В. Организация диалога в системе автоматизации времяпролетного масс-спектрометра.....	12
3.	ГВОЗДЕВ С.В., КРУГЛОВА Н.А. Обработка данных во время эксперимента.....	19
4.	КУЗЬМИНА Л.М. Развитие системы ВАРИАТОР для обработки спектрометрической информации.....	25
5.	БОКАНС Я.Я., КУЗЬМИНА Л.М. Диалоговая подсистема РАДУГА.....	31
6.	КУЗЬМИН Ю.Я. Диалоговая система РИГА.....	37
7.	КУЗЬМИН Ю.Я. Структура диалоговых тестов в системе РИГА.....	50
8.	КАРТАШЕВА Л.И. БЕЛОВА И.В. Контроль знаний студентов в системе РИГА.....	65
9.	АФАНАСЬЕВА А.В. Диалоговый задачник по физике твердого тела.....	72
10.	ЕЗЕПС А.Б. Поиск файлов на диске.....	79
11.	ГУТАНС Я.Я. Контроль дисплеев в многопультной системе.....	86
12.	ГУЖА И.А., СПИЛОВА А.Я. Связь дисплея ИБ ИЭ-00-013 с мини-ЭВМ.....	90

Р е ф е р а т ы

ЛМ 681.32

Васильев И.А., Пирса А.И., Тарасов С.В., Давид Б.В., Понкина Э.А.
РАДИКАЛЬНЫЕ МЕТОДЫ ОБЪЕДИНЕНИЯ СИСТЕМ РАДИО И ТЕЛЕВИЗИОННОЙ
ТЕЛЕВИЗИОННОЙ ТЕХНИКИ И-3000.

Обсуждаются достоинства и недостатки диалоговой системы
связи в системе радиального приема для микр-ЭВМ типа И-3000.
Предлагается структура системы, реализующая диалоговую
связь с компьютером в режиме реального времени. Рассмотрены
варианты реализации системы в виде модуля и ее интеграция в
систему радиального приема.

ЛМ 619.691.6:621.039.343

Васильев И.А., Пирса А.И., Тарасов С.В., Давид Б.В., Понкина Э.А.
РАДИКАЛЬНЫЕ МЕТОДЫ ОБЪЕДИНЕНИЯ СИСТЕМ РАДИО И ТЕЛЕВИЗИОННОЙ
ТЕЛЕВИЗИОННОЙ ТЕХНИКИ И-3000.

Обсуждаются достоинства и недостатки диалоговой системы
связи в системе радиального приема для микр-ЭВМ типа И-3000.
Предлагается структура системы, реализующая диалоговую
связь с компьютером в режиме реального времени. Рассмотрены
варианты реализации системы в виде модуля и ее интеграция в
систему радиального приема.

ДИАЛОГОВЫЕ СИСТЕМЫ АВТОМАТИЗАЦИИ ИССЛЕДОВАНИЙ

Сборник научных трудов

Редакторы: Д.Кузьмин, Т.Фадеева

Корректор и технический редактор М.Дреймане

Подписано к печати 25.04.84. ЯТ 09138. Ф/б 60x84/16.
Бумага № I. 6,8 физ.печ.л. 6,3 усл.печ.л. 5,0 уч.-изд.л.
Тираж 400 экз. Зак. № 789. Цена 75 к.

Латвийский государственный университет им. П.Стучки
Рига 226098, б. Райниса, 19

Отпечатано в типографии, Рига 226050, ул.Вейденбаума, 5
Латвийский государственный университет им. П.Стучки

УДК 681.32

Востры М., Кирпа А.Н., Гвоздев С.В., Лысов Б.И., Поздеев В.А.
РАЗВИТИЕ ДИСКОВОЙ ОПЕРАЦИОННОЙ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ
ДЛЯ МИНИ-ЭВМ ТИПА М-6000.

Обсуждаются достоинства и недостатки дисковой операционной системы реального времени для мини-ЭВМ типа М-6000. Предлагается операционная система, увеличивающая развитие возможностей ДОС РВ с сохранением в основном программного интерфейса между пользователем и операционной системой.

УДК 619.685.6:621.039.343

Павлюк О.В. ОРГАНИЗАЦИЯ ДИАЛОГА В СИСТЕМЕ АВТОМАТИЗАЦИИ
ВРЕМЯПРОЛЕТНОГО МАСС-СПЕКТРОМЕТРА.

Обосновывается необходимость применения директивного способа диалога для систем автоматизации времяпролетного масс-спектрометра. Исходя из задач, решаемых подобными системами, предложен набор директив, включающий средства формирования и отработки составных директив. Работа выполнена с использованием мини-ЭВМ СМ-4.

УДК 681.3.02

Гвоздев С.В., Круглова Н.А. ОБРАБОТКА ДАННЫХ ВО ВРЕМЯ
ЭКСПЕРИМЕНТА.

Обсуждается вопрос об обработке экспериментальной информации в автоматизированной системе в области физики твердого тела. Обработка разделяется на предварительную и детальную. Предварительная обработка рассматривается на примере автоматизированной системы СПЕКТР.

УДК 681.3.051

Кузьмина Л.М. РАЗВИТИЕ СИСТЕМЫ ВАРИАТОР ДЛЯ ОБРАБОТКИ СПЕКТРОМЕТРИЧЕСКОЙ ИНФОРМАЦИИ.

Рассматривается диалоговая система для обработки спектрометрической информации, состоящая из пакета диалоговых документированных программ, особенность которых заключается в наличии одновременно с программными модулями текстовых инструкций для обучения пользователей работе в системе.

УДК 53.05:535.37

Боканс Я.Я., Кузьмина Л.М. ДИАЛОГОВАЯ ПОДСИСТЕМА РАДУГА.

Рассматривается автоматизированная подсистема для измерения спектров поглощения, ее программное обеспечение и состав технических средств. Приведены математические формулы, по которым проводился расчет измеренных спектров.

УДК 681.3.1

Кузьмин Ю.Я. ДИАЛОГОВАЯ СИСТЕМА РИГА.

Обсуждаются особенности текстовых диалоговых систем и проблема защиты информации в них. В качестве примера рассмотрена система РИГА. Технология программирования этой системы показана на задачах текстовых имитаторов.

УДК 681.3.1

Кузьмин Ю.Я. СТРУКТУРА ДИАЛОГОВЫХ ТЕСТОВ В СИСТЕМЕ РИГА.

Рассмотрены варианты создания диалоговых тестов различного назначения: анкетирование, контроль знаний и обучение. Предложены способы формирования переменных тестов и решения проблемы встречных вопросов.

УДК 681.3.1

Карташева Л.И., Белова И.В. КОНТРОЛЬ ЗНАНИЙ СТУДЕНТОВ В СИСТЕМЕ РИГА.

Обосновывается необходимость создания обучающих и контролирующих программ в системе РИГА. На примере тестов, предназначенных для студентов, работающих в физическом практикуме, рассматриваются некоторые методические вопросы составления таких программ.

УДК 681.3.1

Афанасьева А.В. ДИАЛОГОВЫЙ ЗАДАЧНИК ПО ФИЗИКЕ ТВЕРДОГО ТЕЛА.

Показана возможность создания программированных задачник для самоконтроля обучаемых по вопросам физики полупроводников. Предложены некоторые типовые решения программированных задач.

В задачник включены задачи на вычисления и несколько задач качественного характера. В результате неверного решения ЭВМ выясняет причины ошибок и обучает методике решения задач.

УДК 681.3

Бэепс А.Б. ПОИСК ФАЙЛОВ НА ДИСКЕ.

Предлагаемая программа является удобным средством представления общей информации по файлам, производит поиск файлов по отдельным элементам имени, имеется возможность частичного вывода отобранных по определенным признакам файлов. Программа удобна при работе в многопультной системе и с большим количеством файлов.

УДК 681.3

Гутанс Я.Я. КОНТРОЛЬ ДИСПЛЕЕВ В МНОГОПУЛЬТНОЙ СИСТЕМЕ.

Рассматривается комплекс программ, позволяющих в многопультной диалоговой системе производить контроль неисправного дисплея ДМ-2000 без остановки системы.

УДК 681.327.8

Гужа И.А., Спилва А.Я. СВЯЗЬ ДИСПЛЕЯ ИБИЭ-00-013 С МИНИ-ЭВМ.

Приводится описание организации связи мини-ЭВМ типа М-6000 и СМ-1 с дисплеями ИБИЭ-00-013. Сообщаются данные по усовершенствованию блока клавиатуры дисплея, благодаря которому упрощается работа пользователей и продлевается срок службы блока клавиатуры.