

75-50

На правах рукописи

Каминский Леонид Георгиевич

СИСТЕМЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ФИЗИЧЕСКИХ ЭКСПЕРИМЕНТОВ

01,01,10 - "Математическое обеспечение вычислительных комплексов
и АСУ"

А в т о р е ф е р а т

диссертации на соискание учёной степени
кандидата физико-математических наук

(на русском языке)

Дубна 1975

Работа выполнена в Институте физики высоких энергий (г.Серпухов).
Официальные оппоненты: доктор физико-математических наук, профессор Л.Н.Королёв, кандидат физико-математических наук А.И.Волков.
Ведущее предприятие - Институт Кибернетики АН УССР, г. Киев.
Автореферат разослан "16" апреля 1975 г.
Защита диссертации состоится "3" июня 1975 г.
в 11 часов на заседании Учёного совета Лаборатории вычислительной техники и автоматизации Объединенного института ядерных исследований по адресу: г. Дубна, Московской области, ОИЯИ.

С диссертацией можно ознакомиться в библиотеке ОИЯИ.

Учёный секретарь совета

Т.Л.ПУЗЫНИНА

Пузынина

Каминский Леонид Георгиевич

**СИСТЕМЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ФИЗИЧЕСКИХ ЭКСПЕРИМЕНТОВ**

**01.01.10 - "Математическое обеспечение вычислительных комплексов
и АСУ"**

А в т о р е ф е р а т

**диссертации на соискание учёной степени
кандидата физико-математических наук**

(на русском языке)

**Объединённый институт
ядерных исследований
БИБЛИОТЕКА**

Данная диссертация посвящена разработкам программного обеспечения ряда физических экспериментов, проводимых в ИФВЭ, и касается в основном вопросов построения программного обеспечения электронных экспериментов, организации систем обработки снимков с больших пузырьковых камер, технологии и методов работы с большими программами. В основу диссертации положены работы ^{/1-9/}, выполненные автором совместно с сотрудниками ИФВЭ в течение 1971-1975 гг. в Институте физики высоких энергий (г. Серпухов).

В главе I обсуждаются вопросы применения ЭВМ в экспериментальной физике высоких энергий.

В главе II рассматриваются методы, применяемые при построении операционных систем - важной части программного обеспечения.

В главе III описывается система программного обеспечения общего назначения для ЭВМ ИР 2100 - основной машины ИФВЭ, используемой в электронных экспериментах.

В главе IV кратко описывается система обработки filmовой информации с больших пузырьковых камер, а также описывается операционная система, предназначенная для организации вычислительных процедур в больших программах обработки.

В главе V описывается система хранения и модернизации больших программ (РАТСНУ), предназначенная для обеспечения эксплуатации системы обработки.

В Заключении дан перечень основных результатов диссертации.

Созданные системы программного обеспечения физического эксперимента характеризуются применением современных методов программирования, приспособленностью для дальнейшего развития и для работы в условиях полной автоматизации.

Многие компоненты этих систем представляют интерес для включения их в системное программное обеспечение ЭВМ общего назначения.

Описанное в работе программное обеспечение в настоящее время находится в эксплуатации в ИФВЭ и частично в других физических центрах. С его помощью проводятся, в частности, эксперимент СИГМА^{/17/} и эксперименты с большой пузырьковой камерой "Мирабель".

Все основные результаты, включенные в диссертацию, были опубликованы в работах^{/1-14/} и обсуждались на семинарах в ОИЯИ (г. Дубна), ИТЭФ (г. Москва), ИФВЭ (г. Серпухов), ИФВЭ АН Каз.ССР (г. Алма-Ата). В обсуждениях также принимали участие ряд сотрудников ИАЭ им. И.В. Курчатова (г. Москва) и РТИ АН СССР (г. Москва).

I.

В современной экспериментальной физике ЭВМ применяются для^{/15,16/} проектирования экспериментов, сбора экспериментальной информации, контроля, управления и настройки экспериментального оборудования, обработки экспериментальной информации, развития специализированного математического обеспечения, симуляции информационных условий эксперимента.

Одним из важнейших требований к математическому обеспечению физического эксперимента является его мобильность, т.е. возможность легкого приспособления к конкретной физической задаче и способность к изменениям во время эксперимента.

Выполнять эти требования можно разными способами, но в любом случае ЭВМ должна обладать развитым программным обеспечением общего назначения.

Важной задачей является обеспечение физика-экспериментатора простыми и удобными способами взаимодействия с машиной. Решением её является создание операционных систем – от самых простых, управляющих вводом/выводом, до наиболее сложных диалоговых систем.

Программно-аппаратный комплекс ЭВМ целесообразно строить на основе модульного принципа^{/19/}. Последовательно придерживаясь принципов модульного программирования^{/7/}, возможно обеспечить гибкость даже для больших и громоздких программ.

Успешная реализация принципов модульного программирования существенно зависит от корректности сопоставления определённой функции программного комплекса адекватной языковой модели^{/20/}. Выделение такой модели позволяет ввести понятие системного анализатора – центральной части операционной системы. Подавая на вход анализатора грамматику языка, синтаксис которого определяет логику работы, семантика – функции, а прагматика – задачу, решаемую модулем, удаётся стандартизировать разработку модуля операционной системы^{/1/}.

Совокупное применение принципов модульного программирования и лингвистического подхода реализуется при создании универсального (в рамках определённого класса языковых моделей) синтаксического анализатора.

Использованный метаязык R-грамматик^{/18/} допускает обобщение создаваемых алгоритмов, например, для построения программ статистического анализа (типа SUMX), информационно-поисковых систем и т.п.

Рассмотрим одно из возможных представлений R-метаязыка^{/1/}.

Поставим в соответствие данному языку конечный автомат с магазинной памятью, который может производить либо порождение, либо распознавание этого языка. Тогда R-метаязык описывает ориентированный граф переходов, разрешенных для этого автомата.

Граф переходов может быть задан в виде списка возможных переходов, т.е. правил языка. Элементы списка представляют собой комплексы

правил. Комплексы представляются вершинами графа, а правила – исходящими из них рёбрами. В случае, когда одно из правил комплекса не применимо, происходит переход к следующему правилу в этом комплексе до тех пор, пока не найдётся применимое правило, или весь комплекс не исчерпается. В последнем случае фиксируется ошибка во входной строке. Применимость правила определяется по совпадению входного символа с записанным при этом правиле.

В случае, когда правило применимо, происходит переход к рассмотрению очередного комплекса, имя которого задается в этом правиле, и из входной строки выбирается для анализа следующий символ. Одновременно над магазинами производятся операции, которые указаны в правилах. В магазин может быть занесено имя комплекса, и вводится специальная операция над магазином – переход по содержимому магазина к указанному в нем комплексу.

Переход к подграфу с возвратом по магазину используется либо для выделения подграмматик языка, либо для организации рекурсивного вызова подграмматик.

Описанный метод производит синтаксический анализ входного языка. Если каждому правилу сопоставить некоторую последовательность приказов, которые будут исполнены, если правило применимо, то получим транслятор с входного языка на язык указанных приказов. Процедуры, выполняющие эти приказы, называются семантическими подпрограммами.

В некоторых случаях проверка применимости правила возможна только на уровне семантики, поэтому из семантической подпрограммы возможен выход, который указывает, что правило отвергнуто на этом уровне.

Каждому правилу ставится в соответствие следующая информация:

- символ входной строки, для которого применимо данное правило;
- имя семантической подпрограммы и список её фактических параметров;
- имя следующего комплекса правил;
- адрес возврата, который надо занести в магазин, для случая, когда осуществляется переход на какой-либо подграф данного графа.

Анализ входной строки распадается на 5 последовательных этапов:

1. Ввод нового символа из входной строки в том случае, если предшествующее правило применилось.
2. Проверка применимости текущего правила по совпадению символов входной строки с указанным в правиле.
3. Выполнение семантической подпрограммы с возможным выходом в случае неприменимости правила.
4. Регистрация факта применимости правила. Определение имени следующего комплекса правил (возможно получение этого имени из магазина).
5. Возможное занесение в магазин имени того комплекса, к которому надо вернуться после выхода из вызванного подграфа.

Переход к этапу 1.

Перед входом в анализатор производятся необходимые начальные установки, при этом считается, что предыдущее правило применилось.

Выход из анализатора осуществляется либо с помощью специальной семантической подпрограммы, либо при регистрации факта исчерпания магазина возвратов.

Описанный синтаксический анализатор стал ядром операционной системы СНР^{/2/} для машины НР 2100.

III

On-line ЭВМ в экспериментальной физике высоких энергий является частью большой системы, включающей детектирующую аппаратуру, сложное электронное оборудование и группу физиков. Основное назначение ЭВМ в on-line эксперименте сводится к сбору информации и контролю экспериментального оборудования. Однако в экспериментах с использованием большого количества детекторов и сложной электроники (например, эксперимент СИГМА^{/17/} или др.) контроль сильно усложняется из-за чрезвычайно большого количества подлежащих контролю параметров с не всегда известными

допустимыми пределами дрейфа и, возможно, сильными корреляциями. В таких экспериментах контроль установки осуществляется физиками по качеству и непротиворечивости калибровочных физических результатов, получаемых при обработке небольшой выборки из собираемой информации. Поэтому на on-line ЭВМ необходимо уметь решать задачи обработки данных, не прерывая основного режима накопления информации.

Для реализации такого режима работы ЭВМ необходима удобная операционная система с мультипрограммированием.

Стандартная операционная система RTE^{/21/} (Real-Time-Executive) для ЭВМ HP 2100 ориентирована на перфоленту и мало приспособлена для оперативной отладки программ и небольшого счёта.

Созданная операционная система^{/2/} включает в себя файловую память общего пользования на диске, личное хранилище файлов на магнитной ленте, язык диалога, редактор текстовых файлов и телетайп в качестве терминального устройства. Эта система дополняет стандартную RTE-систему. По соображениям удобства в качестве основы системного языка использован язык GEORGE^{/22/}. Поэтому система и получила название GHP (GEORGE HEWLETT-PACKARD).

Операционная система GHP имеет дело с тремя типами файлов: текстовым, библиотечным и двоичным.

Текстовые файлы создаются занесением в них какой-либо информации. Библиотечный файл заполняется компиляторами системы RTE. Двоичные файлы создаются стандартным загрузчиком RTE-системы.

Файлы первых двух типов можно хранить на личной магнитной ленте пользователя. Файлы третьего типа хранить на ленте не имеет смысла.

Как и в операционной системе GEORGE команды в системе GHP состоят из глагола, который может сопровождаться параметрами. Параметрами, как правило, являются идентификаторы файлов.

Редактор EHP^{/3/} представляет собой обычную фоновую программу по отношению к стандартной операционной системе RTE^{/21/}.

Редактируемый и результирующий файлы являются обычными текстовыми файлами системы GHP. Редактор EHP ориентирован на использование

только в on-line режиме, поэтому не имеет редактирующего файла, а все инструкции на редакцию вводятся последовательно с телетайпа.

Язык редактора ЕНР представляет собой последовательность редактирующих инструкций. Каждая фраза может содержать целое число элементарных инструкций, разделенных запятыми, и передается одной строкой терминала от приглашающего символа до посылки конца строки.

On-line редактирование заключается в передвижении по редактируемому файлу специального указателя (pointer), перемежающемся с приемом, обработкой и выполнением редактирующих инструкций.

Указатель имеет формат - номер рекода. номер символа -, т.е. он указывает порядковый номер символа в текущем рекорде редактируемого файла.

В процессе редактирования редактор ЕНР может находиться в двух состояниях - приём инструкций и приём данных.

Продвижение указателя по редактируемому файлу осуществляется специальными редактирующими инструкциями.

Каждая инструкция редактора ЕНР представляет собой состоящее из одного символа мнемоническое сокращение названия операции и спецификации к ней. Инструкции ЕНР можно разделить на следующие категории:

- перемещающие указатель T, P (Transcribe, Position);
- вставка символов или строк I (Insert);
- преобразование строк A, B, R (After, Before, Replace);
- управляющие выводом сообщений редактора L, W (List, Window);
- заканчивающие редактирование E, Q (End, Quit).

В процессе редактирования ЕНР может выдавать на терминал сообщения об ошибках.

Ошибочная инструкция на выполнение не поступает, ошибка не является фатальной, т.е. редактирование может быть продолжено. При обнаружении ошибки во фразе, содержащей набор инструкций, все последующие за ошибочной инструкции не выполняются.

В процессе проведения on-line эксперимента часто возникает необходимость в проведении небольших, но точных расчётов, которые могут носить как численный, так и аналитический характер. Для этой цели должен быть разработан и включен в программное обеспечение эксперимента специализированный разговорный фортрано-ориентированный язык для оперативных вычислений с соответствующим интерпретатором.

При разработке такого языка основное внимание должно уделяться адекватности языка и решаемой проблемы, причём лексика этого языка должна представляться пользователю естественной. Физики-экспериментаторы естественным языком для вычислений считают ФОРТРАН. По этой причине ФОРТРАН (по возможности с минимальными отклонениями) был положен в основу языка, описанного в работе ^{/4/}.

Программное обеспечение ^{/5/} для вывода информации на графический дисплей с запоминанием основано на принципе: иметь оптимальный набор процедур для изображения элементарных объектов и адекватные этому набору простые правила изображения сложных объектов, т.е. должен быть создан необходимый минимум программ с хорошо продуманным способом обращения к ним, при помощи которого можно просто изобразить любой требуемый для решаемой проблемы объект.

Библиотека графических подпрограмм разбита на несколько уровней.

Нулевой уровень – это своего рода программный интерфейс между рабочей программой и изображением на экране. Набор программ нулевого уровня представляет минимальные возможности для графического вывода информации, наилучшим образом реализуя возможности, заложенные в электронике вывода на дисплей.

Программы первого уровня – это более сложные программы, служащие для изображения текстов и часто встречающихся геометрических объектов. Они подготавливают информацию для программ нулевого уровня и через них обращаются к дисплею.

Программы второго уровня – это менее определенный круг программ, потому что их набор полностью определяется характером решаемых на

ЭВМ задач. Обычно это сложные программы для наглядного представления физической информации с более или менее ограниченной универсальностью.

Для нужд эксперимента разработан ^{/6/} универсальный способ расширения системы RTE, позволяющий динамически формировать набор дополнений к диспетчеру из независимых модулей, которые работают после обращения к диспетчеру из программы пользователя, но до включения привилегированного режима.

Реализация основана на глобальном (в пределах данной программы пользователя) перехвате обращений к диспетчеру и замене их обращениями к специальной системе подпрограмм. Этот перехват и переключение обращений происходит динамически и не требует изменения самой программы пользователя, что особенно удобно при использовании стандартных библиотечных процедур, либо в случае, когда программой пользователя является, например, компилятор. При помощи этой системы реализованы форматный вывод информации на дисплей и форматный обмен информацией с текстовыми дисковыми файлами системы RTE.

Описанная система находится в эксплуатации в ИФВЭ более полутора лет. За это время она зарекомендовала себя как удобное и надежное средство проведения физического эксперимента. Найденные в процессе создания и эксплуатации системы решения могут оказаться полезными для других разработок.

IV

Важное место в сфере применения ЭВМ занимает обработка результатов эксперимента. Наибольший опыт в создании и использовании систем обработки накоплен в экспериментах с пузырьковыми камерами. Системы обработки снимков с пузырьковых камер имеют дело с чрезвычайно большим объемом информации. Их характеризуют следующие особенности:

- программы обработки работают с набором событий;
- анализ событий, как правило, многошаговый, с хранением промежуточных данных на накопителях ЭВМ;
- в процессе обработки некоторые шаги повторяются многократно.

Важнейшим элементом современной организации обработки экспериментальных данных является введение стандартов:

- для методики обработки;
- для вычислительных алгоритмов;
- для программирования.

Поэтому в физических центрах существуют системные соглашения и средства для обмена программами и данными. Опыт использования традиционных систем обработки, а также прогресс в области программирования привели к формулировке общих системных принципов ^{/23/} построения новых систем обработки для больших пузырьковых камер. Такие принципы впервые были реализованы в системе программ ЦЕРНа Large Bubble Chamber Programs ^{/23/}.

Участие ИФВЭ в международном сотрудничестве СССР-Франция и СССР-ЦЕРН в экспериментах на Серпуховском ускорителе с пузырьковой камерой "Мирабель" потребовало создания на ЭВМ БЭСМ-6 специальных системных средств и методики адаптации программ указанной выше системы.

В результате использования таких системных средств в ИФВЭ на ЭВМ БЭСМ-6 был адаптирован ^{/10/} весь комплекс программ для обработки filmовой информации с больших пузырьковых камер. Запуск этой системы обработки позволил приступить к широкой программе физических исследований при помощи жидководородной камеры "Мирабель", экспонируемой в пучках Серпуховского ускорителя.

Основные программы обработки filmовой информации очень громоздки. Поэтому здесь особенно важен вклад системных программистов в организацию системы обработки. Разработка современной системы обработки должна начинаться с создания специализированной операционной системы для организации вычислительных процедур.

В организации вычислительного процесса важную роль играют интерпретирующие операционные системы (интерпретаторы), предоставляющие программисту аппарат распределения памяти, вызова вычислительных процедур, средства отладки программ и т.п. Для общения с интерпретаторами обычно не разрабатываются специальные языки, а оформляется оно в рамках используемого языка программирования.

Операционная система, обладающая такими свойствами, была разработана /23/ в ЦЕРНе и реализована /7,11/ на ЭВМ БЭСМ-6.

Анализ системы обработки и опыт работы с программами позволяют сделать следующие выводы:

1) динамическое распределение памяти оказывается удобным и мощным инструментом, и применение его оправдано даже в тех случаях, когда структура данных, может быть, его и не требует, потому что при этом облегчается внедрение модульного принципа (динамическая память рассматривается как псевдоустройство ввода/вывода для связи между модулями);

2) необходимость обработки сложных структур данных требует создания специального механизма для их представления в динамической памяти, который должен охватывать наиболее часто встречающиеся типы структур данных и способы работы с ними;

3) способы общения программы с пользователем, включающие в себя язык управляющих карт, средства отладки и диагностики ошибок, должны быть тщательно продуманы и обладать возможностью расширения;

4) эффективная работа программ может оказаться невозможной, если при её использовании не будет достигнута согласованность работы операционных систем обработочной программы и ЭВМ.

у

Проблема организации хранения текстов больших программ в современном вычислительном центре имеет важное значение. Система хране-

ния должна позволять пользователям иметь различные варианты программ с простым и удобным доступом к любому из них, а также давать возможность дальнейшего усовершенствования программ.

В связи с вводом ^{/10/} в эксплуатацию на ЭВМ БЭСМ-6 цепочки универсальных программ обработки экспериментальных данных, общий объем текстов которых достигает ста тысяч перфокарт, потребовалось создание специальной системы для хранения и развития основных версий этих программ, а также создания рабочих вариантов программ для конкретных экспериментов.

Аналогичные проблемы возникают при создании операционных систем ЭВМ, автоматизированных систем управления и во многих других приложениях ЭВМ.

При современных тенденциях в программировании большие программы обычно представляют собой пакеты модулей и наборы библиотек, а рабочий вариант программы компонуют только из необходимых модулей для данного конкретного случая. Весь набор таких модулей вместе с исправлениями и добавлениями удобно хранить на каком-либо одном накопителе. Такие удобства предоставляет система PATCHY ^{/8,24/}. В ней предполагается хранение произвольной текстовой информации вместе с исправлениями и добавлениями в едином файле (называемом PAM-Patchy Master), а специальная программа-сборщик PATCHY собирает пользователю требуемый вариант текста.

Подготовку и редактирование PAM-файла осуществляет специальная программа-редактор UPPA ^{/9,25/} (UPdate PAm file).

В заключение кратко формулируем основные результаты диссертации:

1. Проведен анализ использования ЭВМ в экспериментальной физике высоких энергий и требований, предъявляемых к программному обеспечению.
2. Проведен анализ теоретических работ по лингвистическому подходу к операционным системам и разработан метод построения операционных систем для программного обеспечения экспериментов в ИФВЭ.

3. На основе разработанного метода создано программное обеспечение для ЭВМ HP 2100, включающее операционную систему GHP, on-line редактор текстов ENP, язык для оперативных вычислений и дисплейную библиотеку. Оно может быть использовано также для ЭВМ M-6000.

4. Разработан универсальный способ расширения системы RTE для ЭВМ HP 2100, позволяющий динамически формировать набор дополнений к диспетчеру из независимых модулей, которые могут объединяться в переключаемые списочные структуры.

5. Осуществлено обеспечение системных средств на ЭВМ БЭСМ-6 при создании в ИФВЭ системы обработки снимков с больших пузырьковых камер и согласование операционной системы ЭВМ с требованиями системы обработки.

6. Реализована на ЭВМ БЭСМ-6 специализированная операционная система интерпретирующего типа, предназначенная для организации вычислительных процедур в программах обработки снимков с больших пузырьковых камер.

7. Реализована на ЭВМ БЭСМ-6 система хранения, модернизации и эксплуатации больших программ, включающая стандарты на структуру текстов, хранимых на внешних накопителях, программу-сборщик RATCHU и программу-редактор UPPA.

Л И Т Е Р А Т У Р А

1. Л.Г.Каминский, Препринт ИФВЭ 74-27, Серпухов, 1974.
2. Л.Г.Каминский, А.А.Лебедев. Препринт ИФВЭ 74-20, Серпухов, 1974.
3. Л.Г.Каминский, С.В.Клименко, А.А.Лебедев. Препринт ИФВЭ 74-73, Серпухов, 1974.
4. Л.Г.Каминский, С.В.Клименко, А.А.Лебедев. Препринт ИФВЭ 75-29, Серпухов, 1975.
5. Л.Г.Каминский, С.В.Клименко, А.А.Лебедев. Препринт ИФВЭ 75-17, Серпухов, 1975.
6. Л.Г.Каминский, С.В.Клименко, А.А.Лебедев. Препринт ИФВЭ 75-7, Серпухов, 1975.
7. Л.Г.Каминский, С.В.Клименко, А.А.Лебедев, Е.А.Слободюк. Препринт ИФВЭ 75-8, Серпухов, 1975.

8. Л.Г.Каминский, С.В.Клименко, С.В.Тарасевич. Препринт ИФВЭ 73-8, Серпухов, 1973.
9. Л.Г.Каминский, С.В.Клименко, В.М.Кубик. Препринт ИФВЭ 73-68, Серпухов, 1973.
10. В.В.Аммосов, В.А.Бумажнов, П.А.Горичев, П.Ф.Ермолов, Л.Г.Каминский и др. Препринт ИФВЭ 75-26, Серпухов, 1975.
11. П.А.Горичев, Л.Г.Каминский, Э.П.Кистенев и др. Препринт ИФВЭ 73-88, Серпухов, 1973.
12. Л.Г.Каминский, Э.П.Кистенев, Е.Д.Шербаков. Препринт ИФВЭ 72-8, Серпухов, 1972.
13. Ю.А.Белокопытов, Л.Г.Каминский, С.В.Клименко, А.А.Лебедев. Препринт ИФВЭ 74-89, Серпухов, 1974.
14. В.В.Гусев, Л.Г.Каминский. Кибернетика, 6, Киев, 1971.
15. L.Kowarski. Proceedings of the 1970 CERN Computing and Data Processing School, Villa Monastero, Varenna, Italy. CERN 71-6, Geneva, 1971, p. 207.
16. Г.И.Забиякин. Сообщение ОИЯИ 10-4231, Дубна, 1968.
17. Ю.М.Антипов и др. Препринт ИФВЭ 72-87, Серпухов, 1972.
18. И.В.Вельбицкий. Кибернетика, 3, Киев, 1973.
19. А.Н.Коновалов, Н.Н.Яненко. Сб. "Комплексы программ математической физики", Новосибирск, 1972.
20. В.М.Глушков, И.В.Вельбицкий, А.А.Стогний. Кибернетика, 3, Киев, 1972.
21. Real-Time Executive Software System, Hewlett-Packard. Part N^o 02005-90001, Oct. 1971.
22. ICL 1900 Series Operating System GEORGE 3 and 4. Manual. Tech. Pub. 4267, 1971.
23. M.Aderholz et al. Proc. 10th Int. Conf. on Data Handling Systems in High-Energy Physics, Cambridge, 1970, CERN 70-21, Vol. I, p. 467.
24. E.Pagiola. Program PATCHY. CERN Computer 6000 Series Program Library II, CERN, Geneva, 1968.
25. Program UPPA. CERN Computer 6000 Series Program Library II, CERN, Geneva, 1968.

Рукопись поступила в издательскую группу

3 апреля 1975 года.