

УДК 681.3

**Структура программного обеспечения систем автоматизации приборов для научных исследований. К и-
б и т к и н В. В. — В кн.: Научное приборостроение. Теоретические и экспериментальные исследования.
Л.: Наука, 1984, с. 139—145.**

**Рассматриваются основные компоненты программного обеспечения современных систем автоматизации
приборов для научных исследований. Выделяются и описываются унифицированные компоненты програм-
мно-го обеспечения: диалоговые системы, технологические средства программирования, управляющие про-
граммы. Анализируется взаимосвязь различных компонент программного обеспечения. Ил. — 1.**

В. В. Кибиткин

СТРУКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМ АВТОМАТИЗАЦИИ ПРИБОРОВ ДЛЯ НАУЧНЫХ ИССЛЕДОВАНИЙ

Современные системы автоматизации приборов для научных исследований (СА), как правило, строятся на базе малогабаритных цифровых электронных вычислительных машин (ЭВМ) и являются программно-управляемыми. В настоящее время для любой ЭВМ характерна тесная взаимосвязь между ее аппаратными средствами и программным обеспечением, реализуемым на этих аппаратных средствах, т. е. ЭВМ рассматривается как аппаратно-программный комплекс или вычислительная система.

С практической точки зрения СА представляют собой проблемно-ориентированные вычислительные системы. Проблемная ориентация заключается в том, что в соответствии с требованиями прибора выбирается конфигурация технических средств СА с необходимыми характеристиками по производительности, разрабатываются специализированные внешние устройства для получения данных от прибора и для управления его работой, а также создается программное обеспечение (ПО), реализующее алгоритмы управления прибором и обработки данных. Таким образом, СА можно рассматривать как специализированные вычислительные системы (СВС) и для их построения можно использовать основные положения теории СВС.

В соответствии с этой теорией СА можно представить в виде многоуровневой иерархической системы. Выбор конкретных уровней иерархии определяется поставленной проблематикой. При рассмотрении проблематики программного обеспечения систем автоматизации, СВС могут быть представлены в следующем виде (см. рисунок).

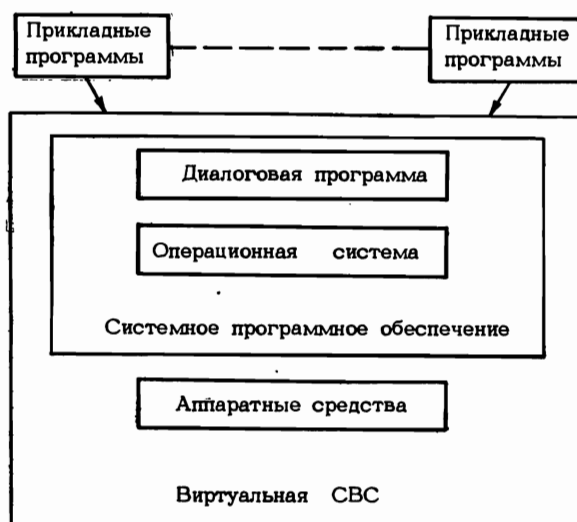


Схема виртуальной СВС.

На нижнем уровне иерархии находятся аппаратные средства. В данном случае они рассматриваются только с точки зрения их использования программистами и поэтому при описании аппаратные средства характеризуются их архитектурой: системой команд, перечнем и функциями программно-доступных регистров, системой прерываний и др. Принципиально описания архитектуры достаточно для того, чтобы разработчик СА мог создавать программное обеспечение. Однако программисту работать на уровне аппаратуры практически нецелесообразно. Дело в том, что в ПО СА можно выделить ряд типовых компонент, которые, с одной стороны, мало зависят от конкретной СА, а с другой — крайне трудоемки с точки зрения их разработки. Поэтому имеет смысл разрабатывать эти компоненты как унифицированные для всех СА с обязательным обеспечением возможности легкой адаптации созданных унифицированных компонент к конкретной СА при минимальном увеличении непроизводительных расходов аппаратных ресурсов, вызванном унификацией программных модулей.

Созданное унифицированное ПО может поставляться как готовое комплектующее изделие наряду с аппаратными средствами, и с точки зрения разработчика СА унифицированное программное обеспечение и аппаратные средства образуют единую специализированную вычислительную систему, обладающую уже функционально новыми возможностями, принципиально облегчающими процесс создания ПО СА. Подобный аппаратно-программный комплекс, рассматриваемый как единая СВС, называется виртуальной СВС.

Понятие виртуальной СВС является очень гибким, поскольку конкретная виртуальная СВС полностью определяется выбранным набором иерархически

упорядоченных средств, включенных в нее, а этот набор средств в свою очередь определяется классом пользователей данной виртуальной СВС. Так, на рисунке изображена виртуальная СВС, предоставляемая для выполнения прикладной программы. Если бы прикладные программы были сюда включены в качестве еще одного уровня, то была бы получена виртуальная СВС, предоставляемая конкретному пользователю системы автоматизации, например физику-экспериментатору, т. е. с позиций пользователя была бы получена система автоматизации, предоставляемая именно ему. Таким образом, виртуальная СВС является полезной абстракцией, помогающей описывать различные уровни функциональных возможностей СВС, получаемых за счет использования различных подмножеств программного обеспечения СВС.

Программное обеспечение СА разбивается на уровни в соответствии с функциональным назначением, взаимосвязями и степенью унификации отдельных программных модулей ПО. Проблемная ориентация ПО воплощается прежде всего в прикладных программах, непосредственно реализующих алгоритмы управления прибором и алгоритмы обработки данных, получаемых от него. Однако эффективная разработка и использование прикладных программ практически невозможны без создания служебного программного обеспечения, называемого системным программным обеспечением (СПО). Назначение СПО — создавать среду для выполнения прикладных программ, но непосредственно с алгоритмами этих программ СПО не связано, поэтому СПО может быть в значительной степени унифицировано, а его разработка — централизована. Это позволит разработчику СА сосредоточить внимание на вопросах эффективного управления прибором и будет способствовать резкому снижению трудоемкости создания ПО СА и существенному повышению его эффективности и надежности.

Для каждого конкретного эксперимента и для отдельных его этапов от системы автоматизации прибора может потребоваться выполнение, вообще говоря, различных действий и тем самым реализация различных подмножеств прикладных программ. Вследствие этого, пользователь прибора должен иметь удобные средства для связи с системой автоматизации прибора в ходе эксперимента, которые реализуются диалоговой системой (ДС). ДС в соответствии с командами пользователя вызывает на исполнение те или иные прикладные программы, в силу чего можно считать, что прикладные программы выполняются в среде ДС.

Языковые средства диалога являются специфическими для каждого прибора (и может быть даже для отдельных типов экспериментов, проводимых с его помощью), но средства планирования и вызова прикладных программ на исполнение (так называемый монитор диалога) могут быть сделаны унифицированными. Кроме того, языковые средства диалога могут быть построены на унифицированной основе, позволяющей легко создавать дополнительные специфические языковые возможности и встраивать их в существующую диалоговую систему. Поэтому диалоговую систему можно рассматривать как верхний уровень системного программного обеспечения, непосредственно предшествующий уровню прикладных программ.

При построении и использовании прикладных программ, так и диалоговой системы возникает ряд общих проблем, решаемых нижним слоем ПО СА — операционной системой (ОС). Программы ОС непосредственно и целиком используют функциональные возможности аппаратных средств и образуют собой среду, в которой выполняются любые диалоговые системы и прикладные программы. ОС непосредственно расширяет функциональные возможности аппаратуры и поэтому разработка ОС является наиболее трудоемкой по сравнению с другими компонентами ПО СА. В то же время ОС значительно слабее этих компонент отражает проблемную ориентацию СА. В силу отмеченных причин к ОС следует предъявлять наиболее высокие требования по унификации.

Задачи, решаемые операционной системой, можно разбить на две группы: автоматизация создания ПО и организация вычислительного процесса в СВС в процессе ее функционирования.

Актуальность первой группы задач обуславливается тем, что программирование на машинном языке требует от программиста исключительно высокого

уровня детализации описания реализуемых алгоритмов. В результате этого трудоемкость и стоимость создания ПО (по сравнению с аппаратными средствами) являются чрезвычайно высокими. Даже при использовании средств автоматизации разработки ПО в настоящее время стоимость ПО составляет преобладающую часть общей стоимости ЭВМ, а без средств автоматизации — развитие и усложнение ПО практически невозможно. Это положение одинаково относится ко всем классам цифровых ЭВМ, начиная от ЭВМ общего назначения и кончая микроЭВМ. Следует отметить, что для мини-микроЭВМ проблема создания ПО усугубляется ограниченностью их ресурсов.

Общепринятым современным подходом к решению проблемы разработки ПО ЭВМ является инженерный подход, состоящий в рассмотрении ПО как одного из видов (хотя и специфического) промышленной продукции. При таком подходе ПО должно создаваться на основе типовых приемов и унифицированных средств, совокупность которых получила название — «технология программирования». Поэтому средства ОС, решающие задачу автоматизации создания ПО можно назвать технологическими средствами программирования. Основными группами этих средств являются языки программирования, средства отладки программ и методология программирования.

Языки программирования, позволяющие записывать алгоритмы на символическом уровне, близком к естественному языку, являются основой автоматизации программирования. В операционную систему включается несколько языков программирования, каждый из которых отражает некоторую ориентацию или на класс решаемых задач, или на требования к эффективности получаемых программ. Второй тип ориентации обуславливает разделение языков программирования на два класса:

— языки программирования высокого уровня, максимально близкие к естественному языку, но мало учитывающие особенности аппаратных средств и в силу этого дающие малоэффективные программы с точки зрения использования аппаратуры СВС;

— языки программирования низкого уровня, близкие к машинному языку и максимально учитывающие особенности аппаратных средств, но трудоемкие в использовании.

Для каждого языка программирования создается транслятор, обеспечивающий автоматический перевод программ, записанных на данном языке программирования, в программы на машинном языке (которые непосредственно выполняются аппаратными средствами СВС). Одному языку обычно соответствует несколько трансляторов: отладочный, оптимизирующий и др.

ПО СА представляет собой комплекс программных модулей, имеющих определенную автономию и, возможно, написанных на разных языках программирования. Для обеспечения работы комплекса как единого целого входящие в него модули должны быть связаны перед выполнением, т. е. в командах передачи управления между модулями вместо символических имен модулей должны быть поставлены адреса их точек входа. Данную функцию выполняют редакторы связей. Они же позволяют строить и выполнять прикладные программы с оверлейной структурой, т. е. программы, отдельные части которых динамически вызываются в оперативную память в ходе выполнения, замещая в памяти друг друга. Оверлейные программы резко уменьшают потребный объем оперативной памяти за счет того, что они не должны там целиком размещаться в ходе выполнения.

Редакторы связей совместно с языками программирования образуют системы программирования. Кроме них в системы программирования включаются также загрузчики, обеспечивающие загрузку программного обеспечения СВС в оперативную память перед выполнением и передачу управления первой выполняемой программе ПО.

Средства отладки программ представляют собой наиболее разнородную группу технологических средств программирования. Сюда прежде всего входят средства документирования, обеспечивающие хранение текстов программ в ЭВМ и их оперативную модификацию в ходе отладки. Средства дампования и трассировки позволяют проследивать ход выполнения программы и получать распечатки содержимого оперативной памяти в указанных точках программы.

Отладчики позволяют в диалоговом режиме следить за выполнением отлаживаемой программы, корректировать ее в ходе выполнения и вновь запускать после корректировки. Тест-мониторные системы служат для тестирования программных модулей и автоматизации анализа результатов тестирования.

В отличие от вышеперечисленных технологических средств программирования, которые являются в основном комплексами программ, средства методологии программирования являются совокупностью способов и приемов, которых должен придерживаться программист в ходе своей работы для создания эффективных программ. Для облегчения использования данной группы средств применяется их программная поддержка, встраиваемая в качестве составной части в средства разработки и отладки программ. Средства методологии программирования оформляются в виде стандартов на программирование. Они охватывают собой и вопросы организации процесса проектирования ПО.

Функция организации вычислительного процесса в СВС выполняется отдельной компонентой ОС — управляющей программой (УП). Необходимость УП определяется тем, что основным режимом работы СВС является мультипрограммный режим, при котором в оперативной памяти одновременно находятся несколько программ, конкурирующих друг с другом за центральный процессор СВС, так что в тех случаях, когда одна программа не может выполняться на процессоре (например вследствие ожидания завершения операции ввода—вывода для нее), процессор передается для выполнения на нем другой программы. Таким образом, мультипрограммный режим работы СВС основывается на динамическом перераспределении ресурсов СВС между несколькими программами. Это перераспределение может производиться автоматически или по запросам от прикладных программ. В последнем случае УП обеспечивает удобный языковой интерфейс для формулирования запросов прикладным программистом.

В простейшем случае мультипрограммирование может быть организовано за счет того, что УП в соответствии с некоторым заложенным в нее алгоритмом диспетчеризации передает процессор от одной программы к другой; это дает возможность ускорить выполнение одних программ за счет замедления выполнения других. Однако общее суммарное время выполнения всего комплекса прикладных программ при этом не уменьшается. В то же время современные аппаратные средства СВС (такие, как каналы ввода—вывода или контроллеры, обеспечивающие прямой доступ к оперативной памяти) позволяют одновременно выполнять счет на процессоре и операции обмена с внешними устройствами. Поэтому на время выполнения операций обмена одной программой процессор может быть передан другой программе, и в данном случае мультипрограммирование уменьшает общее время выполнения программного комплекса, т. е. увеличивает производительность СВС.

Функция организации вычислительного процесса в СВС, выполняемая УП, кроме динамического распределения ресурсов СВС включает в себя еще две функции. Первая из них — это управление операциями обмена между оперативной памятью и внешними устройствами. Реализация этой функции сводится к включению в состав управляющей программы стандартных программ обмена, к которым обращаются все прикладные программы для выполнения любых операций обмена. Необходимо подчеркнуть, что операции обмена наиболее трудоемки в программировании, поскольку они должны максимально учитывать специфику конкретных аппаратных средств, и поэтому вопрос унификации данного типа программных средств особенно важен. И, наконец, последней основной функцией УП является обеспечение языкового интерфейса между УП и прикладными программами, с помощью которого облегчается использование ресурсов пользователями (например упрощается обработка наборов данных, расположенных на внешних запоминающих устройствах, за счет использования символических обращений к данным).

В соответствии со способом использования СВС можно выделить три типа операционных систем.

Первый класс ОС — исполнительные операционные системы (ИОС) — применяется при использовании СВС только для управления прибором и обработки получаемой от него информации. В этом случае СВС циклически реализует

один и тот же набор готовых программ (на машинном языке), поэтому технологические средства программирования в ИОС не входят. Фактически ИОС представляет собой совокупность управляющей программы и программы-загрузчика, служащей для загрузки готовых программ. В связи с тем что набор готовых программ, который реализуется в среде ИОС, известен заранее, появляется возможность точно определить требования к ресурсам СВС. Поэтому многие ресурсы СВС при ее работе в составе прибора можно распределить на этапе разработки прикладных программ, и степень динамичности распределения ресурсов, которую должна обеспечить ИОС, невелика. В предельном (но распространенном) случае ИОС должна выполнять только четыре функции:

- выполнение запросов от прикладных программ на их синхронизацию друг с другом;

- выполнение запросов от прикладных программ на передачу сообщений между ними;

- распределение времени процессора на основании приоритетов прикладных программ;

- управление функционированием внешних устройств в ходе операций обмена данными между этими устройствами и оперативной памятью.

Второй класс ОС — это технологические операционные системы (ТОС), служащие для разработки ПО СА. ТОС включает в себя управляющую программу и технологические средства программирования. Принципиальным отличием УП ТОС от УП ИОС является то, что УП ТОС работает с переменным набором программ, поскольку в ходе отладки программ СВС используется разными программистами, постоянно модифицирующими свои программы. Поэтому запросы на ресурсы СВС, используемой для разработки программ, носят нестационарный характер. В силу этого одним из основных требований к УП ТОС является обеспечение полной автоматизации распределения всех типов ресурсов и обеспечение значительно более высокой по сравнению с УП ИОС степени динамичности распределения ресурсов. Другими требованиями к УП ТОС являются:

- обеспечение защиты пользователей друг от друга и прежде всего обеспечение защиты личных массивов данных, хранимых на внешних запоминающих устройствах;

- наличие языков управления заданиями для указания требований к ресурсам со стороны отлаживаемых программ перед их прогоном на СВС;

- обеспечение независимости программ от внешних устройств;

- обеспечение разнообразных способов организации массивов данных на внешних запоминающих устройствах, включая как обязательный минимум библиотечную организацию;

- наличие средств, обеспечивающих среду для диалоговой отладки программ.

Практически ТОС представляет собой операционную систему, обеспечивающую работу в режиме разделения времени. При этом предполагается, что используется СВС расширенной комплектации, с несколькими терминалами и значительными объемами внешней памяти, что позволяет вести отладку одновременно нескольким пользователям. Многопользовательский режим отладки необходим не только для экономии затрат аппаратных средств, но и для обеспечения возможности совместного использования общих программ и данных несколькими пользователями (отладка каждым пользователем на собственной СВС приводила бы к большим трудностям по поддержанию всех копий общих программ и данных на одном и том же уровне обновления).

Выделение двух классов операционных систем СВС, используемых в СА, исполнительных и технологических, порождает две проблемы: проблему программной совместимости ИОС и ТОС и проблему их одновременного (в режиме мультипрограммирования) использования на СВС. Первая проблема означает, что программы, разработанные и отлаженные в среде ТОС, должны без всяких изменений выполняться в среде ИОС. Вторая проблема возникает в том случае, когда пользователь прибора хочет сам расширять прикладное программное обеспечение СА в ходе его эксплуатации, что влечет за собой необходимость обеспечить в мультипрограммном режиме как реализацию готовых программ, так и

отладку разрабатываемых программ. Обе проблемы могут быть эффективно решены при использовании ОС третьего класса — гипервизорной операционной системы (ГОС). Основой ГОС является гипервизор.

Под гипервизором понимается управляющая программа, которая создает среду для параллельного (в смысле мультипрограммирования) выполнения различных операционных систем. Для этого гипервизор преобразует физические ресурсы СВС в логические. Под логическими ресурсами понимаются некоторые абстрактные ресурсы, предоставляемые в распоряжение операционных систем, выполняемых в среде гипервизора. Основная функция гипервизора — моделирование логических ресурсов на реальных средствах СВС. Конкретные возможности гипервизора определяются теми операционными системами, для которых гипервизор должен создавать среду. В настоящее время используются гипервизоры, создающие среду, в которой выполняется одна ИОС и одна ТОС, т. е. под гипервизорной операционной системой понимается совокупность гипервизора, ИОС и ТОС. По определению гипервизор обеспечивает параллельное выполнение ИОС и ТОС. Программная же совместимость ИОС и ТОС обеспечивается за счет того, что использование гипервизора позволяет рассматривать множество запросов к управляющей программе ИОС как подмножество запросов к управляющей программе ТОС. Гипервизор является самым нижним уровнем программного обеспечения систем автоматизации приборов для научных исследований.