

Лекция 3. Виды обеспечения САПР

1 Техническое обеспечение САПР

2 Программное обеспечение САПР

3 Лингвистическое обеспечение САПР. Языки программирования

1 Техническое обеспечение САПР

Основу технического обеспечения САПР составляет, как правило, персональный компьютер. Конструктивно он представляет собой системный блок, состоящий из корпуса с блоком питания, в котором установлены: материнская плата с процессором, оперативной памятью, видеокартой (видеоплатой), при необходимости звуковой картой и сетевой картой; жесткий диск (винчестер); привод для компакт – дисков; привод для дисков. Кроме этого в состав компьютера обязательно входит монитор (дисплей), клавиатура, манипулятор «мышь». Это основные устройства компьютера. Кроме них применяются периферийные устройства: принтер, плоттер, звуковые колонки, микрофон, цифровой фотоаппарат, цифровая видеокамера.

На практике в настоящее время широко применяются локальные вычислительные сети (ЛВС). Это принадлежащая одной организации коммуникационная система, связывающая различные аппаратные средства: компьютеры, принтеры, плоттеры. Слово «локальная» указывает на близость расположения компьютеров. Диапазон действия ЛВС колеблется от нескольких метров до 8 – 10 км.

ЛВС предоставляет пользователям следующие возможности:

- **обмен информацией** (сообщениями электронной почтой, файлами текстовых документов, чертежей и программ);
- **разделение ресурсов компьютеров**, т.е. совместное использование баз данных и программ, хранящихся на любом из компьютеров сети (либо на удаленном мощном компьютере – сервере с жестким диском большой емкости);
- **вывод информации**, например, на лазерный принтер или плоттер, подключенный только к одному из компьютеров сети.

ЛВС состоит из следующих основных элементов:

- файлового сервера;
- рабочих станций;
- сетевой операционной системы;
- несущей среды (кабелей), сетевых карт и других аппаратных средств.

Сервер – это мощная ПЭВМ, на жестком диске которой хранятся прикладные программы, базы данных и т.д., необходимые для работы пользователей сети. Сервер, предназначенный только для обслуживания сетевых запросов, называется выделенным. При генерации (установке) сети можно сформировать и совмещенный сервер, на котором можно работать как на рабочей станции.

Рабочие станции – подключенные к сети ПЭВМ, на которых работают отдельные пользователи.

Каждая рабочая станция и сервер в ЛВС должны иметь специальное программное обеспечение: сетевую оболочку или операционную систему.

Кабель в ЛВС определяет физическую среду передачи информации. Существует три типа кабелей:

1. **Витая пара** (физически этот кабель состоит из четырех витых пар в оплётке, одна пара используется для передачи информации в одном направлении, вторая – в другом направлении, две оставшиеся пары предназначены для передачи служебных сообщений по сети, на практике иногда они не используются).
2. **Коаксиальный** (типа телевизионного).
3. **Волоконно – оптический** (физически это кварцевая нить в полимерной оплётке, оплётка предназначена для придания гибкости кабелю; передача информации по кабелю производится световыми излучениями с **разной** длиной волны, за счет чего образуется ряд информационных каналов).

Скорость передачи информации – важнейший показатель эффективности сети, она измеряется в Мбит/с, Гбит/с. Скорость передачи информации по витой паре составляет от 10 до

100 Мбит/с, по коаксиальному кабелю – от 0,5 до 10 Мбит/с, по волоконно – оптическому теоретически – сотни Гбит/с, практически – около 2 Гбит/с (за счет более низкой пропускной способности приемных и передающих устройств).

Сетевые карты физически могут быть встроены в материнскую плату или устанавливаться в разъемы системного блока компьютера. Их тип определяется выбранной топологией сети.

В ЛВС компьютеры располагаются сравнительно недалеко друг от друга. Для связи на большом расстоянии можно использовать аппаратуру обычных телефонных линий, которая, правда, поддерживает относительно низкую скорость передачи информации. Дополнительным устройством при этом является модем. Когда с компьютера информация передается по телефонной линии, передаваемые сигналы подвергаются **модуляции**, а когда принимается – **демодуляции**. Отсюда название – модем. Назначение модема – замена двоичного сигнала компьютера (сочетания 0 и 1) аналоговым сигналом с частотой, соответствующей рабочему диапазону телефонной линии.

Конструктивно модем – это печатная плата, вставляемая в компьютер или присоединяемая к нему, связанная с кабелем, подключаемым к телефонной розетке.

Телефонные сети начинают переводиться на цифровые сигналы, совместимые с сигналами компьютеров. Поэтому необходимость в модемах в перспективе отпадет.

2 Программное обеспечение САПР

Программное обеспечение (ПО) САПР – совокупность машинных программ и сопутствующих им эксплуатационных документов, необходимых для выполнения автоматизированного проектирования.

ПО подразделяется на **общее** и **прикладное** (специальное). В свою очередь, общее ПО можно подразделить на **общесистемное** программное обеспечение и **языки** (среды, студии) **программирования**.

К программному обеспечению предъявляются следующие требования:

- **экономичность** (эффективность по быстродействию и затратам памяти);
- **удобство использования**, применение простых проблемно-ориентированных языков, наличие средств диагностики ошибок пользователя и т.п.;
- **надежность и правильность получения результатов проектирования**; универсальность по отношению к тем или иным ограничениям решаемых задач;
- **открытость** (адаптируемость) относительно внесения изменений в процессе эксплуатации программ;
- **сопровождаемость**, характеризующая работоспособность программ при внесении изменений в них;
- **мобильность** при перестройке программ с ЭВМ одного типа на ЭВМ другого типа.

Программное обеспечение целесообразно разрабатывать на основе принципов модульности и иерархичности. Принципы модульности и иерархичности позволяют организовать коллективную параллельную разработку различных частей программного обеспечения, создавать открытые программные системы, облегчают их комплексную отладку и информационное согласование.

Общесистемное программное обеспечение САПР

Общесистемное ПО служит для организации функционирования технических средств. **Операционная система** является основным компонентом системного программного обеспечения САПР.

Операционная система – это комплекс программ, который загружается при включении компьютера. Она производит диалог с пользователем, осуществляет управление компьютером, его ресурсами (оперативной памятью, местом на дисках и т.д.), запускает прикладные программы на выполнение и т.д.

Операционные системы включают в себя программы двух групп.

- **обрабатывающие** программы, составляющие подсистему подготовки программ пользователя (внешнее программное обеспечение);
- **управляющие** программы, образующие группу исполнения программ пользователя (внутреннее программное обеспечение).

К обрабатывающим программам относятся трансляторы с алгоритмических языков, библиотеки стандартных программ и системные обслуживающие программы.

Управляющая часть программного обеспечения имеет иерархическую организацию и в общем случае в ней можно выделить различные уровни: уровень операционных систем вычислительной сети, операционных систем отдельных ЭВМ, мониторных систем САПР и мониторов отдельных ППП.

Основные функции управляющей части, связь с пользователем в режиме диалога, планирование вычислительного процесса, распределение вычислительных ресурсов, динамическое распределение памяти и др.

Группа управляющих программ включает в себя программы управления задачами, заданиями и данными.

Программа управления задачами (супервизор, диспетчер, монитор, резидентная программа) находится в оперативной памяти и выполняет все необходимые диспетчерские функции — переключение с выполнения одной программы на другую, распределения ресурсов времени и оперативной памяти между программами. Именно супервизор реализует мультипрограммный режим работы ЭВМ или режим разделения времени.

Программы управления заданиями выполняют интерпретацию директив языка управления заданиями — ввод, трансляция, загрузка в память ЭВМ, решение, вывод информации.

Программы управления данными обеспечивают поиск, хранение, загрузку в оперативную память и обработку файлов.

Функции и структуры операционных систем имеют различия в одно- и многопроцессорных вычислительных системах, многомашинных комплексах и вычислительных сетях. Соответственно этому операционная система должна обеспечить одно- или мультипрограммный режим работы ЭВМ, режим мультиобработки задач, совместное функционирование уровней и подсистем САПР через специальную мониторную систему.

Операционная система типа Windows предоставляет следующие возможности для программистов:

1. **Независимость программ от внешних устройств.** DOS — программа может работать с аппаратными средствами компьютера (монитором, клавиатурой, принтером и т.д.) непосредственно, минуя Windows. Windows — программа может обращаться к внешним устройствам только через Windows. Это снимает с программиста проблему обеспечения совместимости с конкретными внешними устройствами, т.к. ее берет на себя Windows. Поэтому любая Windows — программа может работать с любым внешним устройством, если с ним может работать Windows. Программы (драйверы) для поддержки наиболее распространенных устройств входят в Windows, а для остальных устройств — поставляются вместе с этими устройствами.

2. **Наличие средств для построения пользовательского интерфейса.** В Windows входят все необходимые средства для построения пользовательского интерфейса: окон, меню, запросов, списков и т.д. При этом стиль пользовательского интерфейса практически стандартен и считается одним из лучших.

3. **Доступность всей оперативной памяти.** В отличие от MS DOS средства управления оперативной памятью Windows обеспечивают доступность всей оперативной памяти компьютера (а не только ее части), что облегчает создание больших программ.

Другие возможности: обмен данными между приложениями Windows, организация встроенных справочных программ и т.д.

Для пользователей Windows предоставляет следующие возможности:

1. **Единый пользовательский интерфейс.** Т.к. Windows предоставляет программисту все необходимые средства для создания пользовательского интерфейса (окон, меню и т.д.), то программисты пользуются ими, а не изобретают собственные средства. Вследствие этого пользовательский интерфейс Windows — программ в

значительной степени унифицирован, и пользователям не требуется изучать для каждой программы новые принципы организации взаимодействия с этой программой.

2. **Многозадачность.** Windows обеспечивает возможность одновременного выполнения нескольких программ, переключения с одной задачи на другую, управления приоритетами выполняемых программ.

3. **Поддержка мультимедиа.** При подключении соответствующих устройств Windows может воспринимать звуки от микрофона, компакт – диска, изображения от цифрового фотоаппарата, цифровой видеокамеры или с компакт – диска, выводить звуки на колонки или в наушники, выводить на экран монитора движущиеся изображения.

Другие возможности: совместимость с DOS – приложениями, удобство поддержки устройств, поддержка масштабируемых шрифтов и т.д.

Для создания программ используются языки (среды, студии) программирования.

Прикладное программное обеспечение САПР

Прикладное программное обеспечение представляют пакеты прикладных программ (ППП) для выполнения различных проектных процедур. Оно разрабатывается на основе единого внутреннего представления графической и текстовой информации, единого входного языка, строится по модульному принципу и ориентировано на использование непрограммистом-проектировщиком.

Различают несколько типов ППП в зависимости от состава пакета. Пакеты прикладных программ **простой** структуры характеризуются наличием только обрабатывающей части — набора функциональных программ (модулей), каждая из которых предназначена для выполнения только одной проектной процедуры. Объединение нужных модулей осуществляется средствами операционной системы ЭВМ.

Пакеты прикладных программ **сложной** структуры и **программные** системы появились в результате развития прикладного программного обеспечения. В первых из них имеется собственная управляющая часть — **монитор**, во вторых, кроме того, **языковой процессор** с проблемно-ориентированным входным языком. Программные системы вместе с соответствующим лингвистическим и информационным обеспечением называют **программно-методическими комплексами САПР**.

Выделяют системный уровень разработки прикладного программного обеспечения, уровень прикладных программ и уровень подпрограмм (модулей).

Связи между отдельными программными модулями могут быть реализованы по управлению, информации, размещению и воздействию.

Связи модулей по управлению могут быть двух типов: последовательные связи между модулями без возврата в предыдущий модуль и иерархические связи с подчиненностью модулей различных уровней

Связи модулей по информации проявляются в передаче числовых массивов в несколько модулей пакета. Этот аспект взаимодействия модулей затрагивает проблемы построения информационного обеспечения САПР.

Связи модулей по размещению указывают группы модулей, одновременно размещаемых в оперативной памяти на различных этапах проектирования.

Связи модулей по воздействию отражают такие воздействия одних программ на другие, которые приводят к изменению самих программ, например воздействие языковых процессов на рабочие программы. Внутри рабочих программ связи модулей по воздействию стараются исключить.

К настоящему времени разработано большое количество пакетов прикладных программ САПР электрических и электронных средств. В качестве примеров можно привести ДИСП, САМРИС-2, СПАРС, АРОПС, КРОСС и др. Из зарубежных систем можно отметить пакеты MicroCAP, PSPICE, P-CAD, SPADE и др.

Значительное число этих пакетов ориентировано на автоматизацию проектирования печатных плат, цифровых и аналоговых интегральных схем, операционных усилителей, низкочастотных радиотехнических устройств. В то же время весьма недостаточно разработано пакетов программ проектирования радиочастотных, в том числе мощных устройств, радиоэлектронных средств СВЧ, пакетов, посвященных комплексному построению и интеграции

радиочастотных средств, включающих в себя как усилители, так и пассивные радиочастотные устройства, вплоть до антенн и СВЧ-устройств.

Развитие программного обеспечения САПР требует все более значительных затрат высококвалифицированного труда. Стоимость многих промышленных САПР составляет миллионы долларов. Поэтому актуальной становится разработка САПР второго порядка, или САПР САПРов. Пока таких систем еще не существует, но прогресс в этом направлении имеется. В отличие от традиционных САПР в таких системах результат имеет нематериальный (информационный) характер. Различие результатов вызвано различными языками описания предметных областей: в одном случае — чертежи, схемы, устройства, а в другом — программа проектирования. Однако и в том, и в другом случае возможен единый системный методологический подход к проектированию; становится актуальным создание и развитие банка инженерных знаний, необходимых для проектирования.

3 Лингвистическое обеспечение САПР. Языки программирования

Лингвистическое обеспечение — совокупность языков, используемых в процессе разработки и эксплуатации САПР.

Под «языком» понимается любое средство общения, любая система символов и знаков для представления и обмена информацией.

Лингвистическое обеспечение образуется следующими языками:

- программирования;
- управления;
- проектирования.

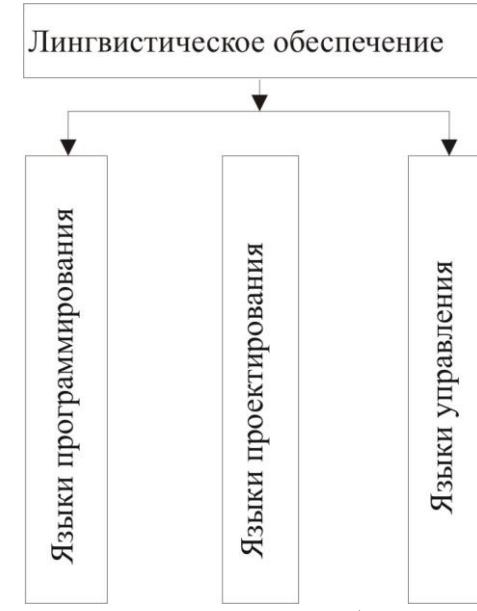


Рисунок 1 – Лингвистическое обеспечение

Языки программирования необходимы для создания программного обеспечения при разработке САПР. В принципе языки программирования относят и к программному обеспечению САПР. Здесь мы их подробно рассматривать не будем, информация о них приведена в специальной литературе. Напомним лишь, что к наиболее распространенным языкам программирования относятся Pascal, Fortran, Basic, Си (различных версий). В настоящее время на их базе разработаны и повсеместно используются среди программирования такие, как, соответственно, Delphi, Visual Fortran, Visual Basic, Visual Си (также различных версий).

Существуют различные уровни языков программирования: высокие, более удобные для пользователя, и более низкие, близкие к машинным языкам.

Программа, записанная на некотором языке программирования высокого уровня, называется исходной. Прежде чем исходная программа будет исполнена, она должна быть преобразована в

машинную форму, соответствующую ЭВМ данного типа. Подобные преобразования осуществляются специальными программами, называемыми **языковыми процессорами**.

Основные типы языковых процессоров — трансляторы и интерпретаторы; соответственно преобразования программ называют трансляцией и интерпретацией.

Трансляцией называют перевод всего текста программы на исходном языке (исходной программы) в текст на объектном языке (объектную программу). Если исходный язык является языком высокого уровня, а объектный — машинным, то транслятор называют **компилятором**. Если исходный язык — машинно-ориентированный (в автокоде), а объектный — машинный, то транслятор называют **ассемблером**. Если исходный и объектный языки относятся к одному уровню, то транслятор называют **конвертером**.

По методу трансляции (компиляции) сначала исходная программа переводится на машинный язык, а затем скомпилированная рабочая программа исполняется.

При **интерпретации** перевод исходной программы в рабочую совмещены во времени; очередной оператор исходной программы анализируется и тут же исполняется.

В большинстве случаев применение трансляторов приводит к меньшим затратам машинного времени, но к большим затратам машинной памяти, чем при интерпретации.

Совокупность языка программирования и соответствующего ему языкового процессора называют **системой программирования**

Класс машинно-зависимых языков представлен ассемблером (макроассемблером). Он относится к языкам низкого уровня и используется для написания программ, явно использующих специфику конкретной аппаратуры.

К машинно-ориентированным языкам относится язык **СИ** (разработан в 1972 г.). В нем объединяются достоинства низкоуровневых возможностей ассемблеров и мощных выразительных средств языков программирования высокого уровня. Язык СИ является одним из претендентов на роль основного языка программирования в САПР и ориентирован на разработку системных программ. Он, в частности, послужил главным инструментом для создания операционных систем для ПЭВМ UNIX и MS DOS.

Язык **Фортран** является первым универсальным языком высокого уровня (с 1954 г.). Фортран — самый примитивный из распространенных алгоритмических языков общего назначения. Наиболее эффективен при численных расчетах, прост по структуре и эффективен при выполнении программ. Несмотря на свои недостатки, этот язык получил большое распространение при разработке прикладных программ для решения научных задач. Наиболее популярная в настоящее время версия этого языка — Фортран-77.

Идеи Фортрана получили развитие в языке PL/I (создан в 1964 г). В нем сделана попытка преодолеть некоторые недостатки, свойственные языкам для больших ЭВМ и использованы идеи структурного программирования. В настоящее время имеются различные версии этого языка: PL/M, PL/Z, PL/65 и др. Как язык программирования PL/M, в частности, значительно уступает конкурирующим с ним языкам Паскаль и Модула-2.

Язык **Паскаль** является одним из наиболее популярных языков программирования и используется для разработки системных и прикладных программ, в частности для персональных ЭВМ. Язык Паскаль создан вначале исключительно для учебных целей и изящно реализовал большинство идей структурного программирования. Достоинства языка оказались столь значительными, что он приобрел огромную популярность для самых различных приложений. В частности, компилятор Turbo Pascal, снабженный интерактивным редактором, позволяет создавать достаточно сложное программное обеспечение — системы управления базами данных, графические пакеты и др.

Развитием Паскаля являются языки Модула-2 (в Европе) и **Ада** (в США). Язык Модула-2 обладает лучшими средствами для обработки больших программных комплексов и позволяет более эффективно использовать особенности аппаратуры. Таким образом, этот язык призван заполнить ниши между Паскалем и СИ. По оценке специалистов, язык Модула-2 через несколько лет станет наиболее популярным среди всех языков программирования.

Язык Ада можно назвать наиболее универсальным среди созданных языков. Однако трансляторы с этого языка пока не получили достаточного распространения.

Язык **Алгол** — общепризнанный язык для публикации алгоритмов решения научных задач, построен на четких и полных определениях. Для Алгола характерны строгие, но негибкие

структуры данных и программ. Алгол труден для реализации на большинстве ЭВМ, поэтому используются неполные варианты языка или его расширения.

Язык **Кобол**, разработанный для решения экономических задач, широко распространенный на больших и средних ЭВМ прошлых лет, на персональных ЭВМ почти не используется. На ПЭВМ имеются интегрированные системы, базы данных и другие типы прикладных систем, используемые в задачах экономического и управленческого характера.

Самыми распространенными на ПЭВМ являются различные версии языка Бейсик, простота которого делает его превосходным средством для начинающих программистов. В языке встроены удобные функции для работы с экраном дисплея, клавиатурой, внешними накопителями, принтером, каналами связи. Это позволяет относиться к Бейсику как к продолжению аппаратуры ПЭВМ. Системы Бейсика работают в режиме интерпретации, что способствует сокращению характерного цикла в работе программиста: составление программы — пробное исполнение — исправление ошибок — повторное исполнение.

Бейсик наряду с Паскалем принят во многих учебных заведениях как базовый язык для изучения программирования.

Язык **АПЛ** применяется для обработки структурных данных (векторов, матриц) и использует иероглифическую запись программных текстов. Из-за большого числа иероглифов (около 100) его иногда называют китайским Бейсиком.

К *классу проблемно-ориентированных* языков можно отнести Лого, CPSS, Форт и Смолток.

Язык **Лого** — диалоговый процедурный язык, реализованный на принципе интерпретации и работающий со списками, текстами, графическими средствами и др. Язык очень перспективен для обучения, создания электронных игрушек и т.д.

В системах программирования GPSS легко описывается и исследуется класс моделей массового обслуживания.

Развитием проблемно-ориентированных языков является объектно-ориентированный подход (языки Смолток, Форт, Модула и Ада). Отличительными особенностями таких языков являются модульность построения процедур, абстракция данных, динамическая связка программ, что позволяет отказаться от перекомпилирования всей программы при внесении изменений в отдельные модули, и использование механизма наследования иерархического типа.

К недостаткам таких языков относится некоторая замедленность выполнения программ из-за их динамической связи и сложность трансляторов.

Язык **Смолток** предназначен для решения нечисловых задач при построении систем искусственного интеллекта. В языке **Форт** применены структурное программирование и очень компактный машинный код.

Для разработки систем искусственного интеллекта также используются *функциональные языки Лисп, Пролог и Снобол*. Эти языки ориентированы на обработку символьной информации, требуют больших массивов данных и стали применяться в ПЭВМ в связи с появлением дешевой полупроводниковой памяти, позволяющей довести объем ОЗУ до нескольких Мбайт. Языки этого класса относятся к так называемым языкам представления знаний.

Язык **Лисп** применяется для программирования интеллектуальных задач — общение на естественном языке, доказательство теорем, принятие решений и т.п.

Язык **Пролог** приобрел в последние годы большую популярность в связи с японским проектом создания вычислительных систем пятого поколения. Он предназначен для создания широкого класса систем искусственного интеллекта, в том числе и персональных экспертных систем.