

Лекция 6. Математическое обеспечение АСУ

**6.1. Классификация средств математического
обеспечения АСУ**

6.2. Основные классы задач АСУП

**6.3. Языки программирования для описания
задач в АСУП**

Под математическим обеспечением АСУ
понимается совокупность различных
математических методов, моделей, алгоритмов и
комплексов программ, обеспечивающих
функционирование АСУ в соответствии с ее
целевым назначением.

Под термином математическое обеспечение АСУ понимают
математическое, лингвистическое и программное обеспечение
АСУ.

Особенностью математического обеспечения АСУ является:

- увеличение относительной стоимости *математического*
обеспечения по сравнению с комплексом технических средств
(КТС) АСУ;
- разумная типизация (унификация) прикладного программного
обеспечения;
- широкое применение ППП, стандартных оболочек и др.

- *Математическое обеспечение (МО)* можно разделить на три части:
- *МО ЭВМ (или внутреннее);*
- *специальное математическое обеспечение (или внешнее);*
- *программные средства телеобработки данных*

Внутреннее МО включает операционные системы (MS DOS), системы программирования и тесты (программы проверки исправности работы устройств ЭВМ),

- **Операционная система (ОС)** - набор программ, управляющих процессом решения задач. **Оптимальная загрузка всех узлов**
- **ЭВМ** и внешних устройств является **основной задачей ОС.**
- В состав ОС входит ряд программ, из которых основными являются: **диспетчер, супервизор, служебные программы.**
- ***Диспетчер*** - программа, обеспечивающая определенный режим работы ЭВМ.
- ***Супервизор*** - программа, обеспечивающая работу, задаваемую машине человеком-оператором в рамках установленного для нее режима.
- ***К служебным*** - относятся программы ввода исходных данных; программы редактирования и выдачи результатов; программы общения ОС с человеком-оператором и др.
- **ОС различают по целевому назначению на:**
- **общие** — для решение широкого круга задач и **проблемные.**
- В зависимости от организации решения задач на ЭВМ различают следующие **режимы работы ОС: индивидуальный, пакетный, мультипрограммирование, разделение времени.**

- **При индивидуальном режиме ЭВМ** постоянно или на время решения задачи находится полностью в распоряжении одного потребителя.
- **Пакетная обработка** предполагает, что пользователь не имеет непосредственного доступа к ЭВМ. Подготовленные им задачи в виде программ и исходных данных загружаются оператором в ЭВМ и решаются пакетами.
- **Мультипрограммирование** предполагает возможность одновременно решать несколько задач по различным программам с учетом приоритета.
- *При этом в каждый момент времени решается одна задача. Если при решении задачи появилась необходимость решения другой с более высоким приоритетом, то решение задачи прерывается, решается вторая задача, после ее решения продолжается решаться первая с того места, где произошла остановка и т.д.*
- **Режим разделения времени** предполагает одновременное решение нескольких задач.

Основными целями ОС являются:

- увеличение производительности вычислительных систем (ВС) путем обработки непрерывного входного потока заданий и совместного использования ресурсов ВС одновременно выполняющимися в ОП задачами (эффект мультипрограммирования);
- планирования ВС в соответствии с приоритетами отдельных заданий, ведение учета и контроля использования ресурсов;
- обеспечение программистов средствами разработки и отладки программ;
- обеспечение оператора средствами управления ВС.

Система программирования предназначена для автоматизации процесса программирования задач, она содержит трансляторы алгоритмических языков различных уровней и типов и обслуживающие программы.

Система служебных программ (тестов) предназначена для контроля правильности функционирования ВС, обнаружения неисправностей и анализа видов и причин сбоев.

Специальное (внешнее) МО включает ППП, программы конкретных задач АСУП, системную диспетчерскую программу.

ППП - функционально законченные комплексы программных средств, ориентированные на решение определенного класса задач.

Программы конкретных задач АСУП можно условно **разделить на 3 класса:**

- **программы общие** для всех отраслей (промышленности, транспорта, торговли и др.);
- **программы общие для предприятий** авиационной отрасли;
- **программы специфические** для каждого предприятия (АРЗ, авиационного производственного объединения и др.).
- **К 1 классу задач** относятся задачи: (расчет заработной платы, учет кадров, учет материальных ценностей и т.д.).
- **Ко второму** - задачи диспетчерского управления (расчет режимов работы оборудования, расчет выпуска АТ и др.).
- **К третьему** - специфические задачи ремонта АТ (выпуск запасных частей при ремонте, подготовка АТ к вылетам и др.).
- Большое количество различных по целям и значению программ требует их организации в масштабах всей системы и это выполняется с помощью **системной диспетчерской программы.**

- *МО строится на основе **типизации алгоритмов** по классам задач и **унификации** методов решения родственных задач. Такой подход позволяет **удешевить МО**, а также создать **единые модели** для решения различного класса задач.*
- **К первому классу задач относятся задачи первичного учета (массовые)** (повторяемость расчетов с абонентами - миллионы в год, расчетов по заработной плате - сотни тысяч в год и т.п.).
- **Примеры задач первичного учета:** суточный, декадный, месячный и годовой **учет** поступления и расхода ГСМ по авиакомпаниям, отрядам и др.; суточный и недельный, месячный **налет** ВС; **учет** и анализ отказов авиационного оборудования; **учет** движения и запасов материальных средств и др.

Первичный учет позволяет накопить попутно большое количество информации, последующее обобщение которой позволит получить ***полноценные статистические данные, необходимые для принятия решений.***

Эти задачи образуют **класс учетно-статистических задач**, к которым примыкают и **задачи нормативного планирования.**

Математической характеристикой этих задач является большое количество логических операций при небольшом объеме простых математических операций.

В числе задач этого класса можно отметить: ***составление всех форм статистической и бухгалтерской отчетности; расчет себестоимости продукции; расчеты потребностей в ГСМ и т.д.***

Обширную группу среди перечисленных составляют бухгалтерские задачи, характеризующиеся большим числом операций сложения, вычитания, логических операций (сортировка, группировка, сравнение) и формированием таблиц заданной формы.

Математическое моделирование широко применяется в значительной в трех принципиально разных классах **задач: в сложных неэкстремальных расчетах, прогнозировании и оптимизации.**

В АСУ за человеком остаются **функции принятия решений** на основе данных выданных АСУ, непосредственное **наблюдение** за управляемым процессом (объектом) (контроль), разработка и установление решающих правил (критериев, нормативов, предельных уровней контролируемых величин), **совершенствование управления и его формы, анализ результатов** работы ЭВМ и подготовка мероприятий по совершенствованию работы системы.

6.3 Языки программирования для описания задач в АСУП

ЯЗЫКИ ВЫСОКОГО УРОВНЯ (т.е. немашинные языки), которые стали своеобразным связующим мостом между человеком и машинным языком компьютера. Языки высокого уровня работают через **трансляционные программы**, которые вводят "исходный код" (гибрид английских слов и математических выражений, который считывает машина), и в конечном итоге заставляет компьютер выполнять соответствующие команды, которые даются на машинном языке. Существует **два основных вида трансляторов: интерпретаторы**, которые сканируют и проверяют исходный код в один шаг, и **компиляторы**, которые сканируют исходный код для производства текста программы на машинном языке, которая затем выполняется отдельно.

1.1. Интерпретаторы

Одно, часто упоминаемое преимущество интерпретаторной реализации состоит в том, что она допускает "непосредственный режим". Непосредственный режим позволяет вам задавать компьютеру задачу вроде PRINT 3.14159*3/2.1 и возвращает вам ответ, как только вы нажмете клавишу ENTER (это позволяет использовать компьютер стоимостью 3000 долларов в качестве калькулятора стоимостью 10 долларов). Кроме того, интерпретаторы имеют специальные атрибуты, которые упрощают отладку. Можно, например, прервать обработку интерпретаторной программы, отобразить содержимое определенных переменных, бегло просмотреть программу, а затем продолжить исполнение.

1.2. Компиляторы

Компилятор-это транслятор текста на машинный язык, который считывает исходный текст. Он оценивает его в соответствии с синтаксической конструкцией языка и переводит на машинный язык. Другими словами, компилятор не исполняет программы, он их строит. Интерпретаторы невозможно отделить от программ, которые ими прогоняются, компиляторы делают свое дело и уходят со сцены. При работе с компилирующим языком, таким как Турбо-Бейсик, вы придете к необходимости мыслить о ваших программах в признаках двух главных фаз их жизни: периода компилирования и периода прогона.

2. КЛАССИФИКАЦИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

2.1. Машинно – ориентированные языки

Машинно – ориентированные языки – это языки, наборы операторов и изобразительные средства которых существенно зависят от особенностей ЭВМ (внутреннего языка, структуры памяти и т.д.). Машинно – ориентированные языки позволяют использовать все возможности и особенности Машинно – зависимых языков:

- **высокое качество создаваемых программ (компактность и скорость выполнения);**
- **возможность использования конкретных аппаратных ресурсов;**
- **предсказуемость объектного кода и заказов памяти;**
- **для составления эффективных программ необходимо знать систему команд и особенности функционирования данной ЭВМ;**
- **трудоемкость процесса составления программ (особенно на машинных языках и ЯСК), плохо защищенного от появления ошибок;**
- **низкая скорость программирования;**
- **невозможность непосредственного использования программ, составленных на этих языках, на ЭВМ других типов.**

Машинно-ориентированные языки по степени автоматического программирования подразделяются на классы:

2.1.1. Машинный язык

компьютер имеет свой определенный **Машинный язык** (далее **МЯ**), ему предписывают выполнение указываемых операций над определяемыми ими операндами, поэтому **МЯ** является командным. Однако, некоторые семейства ЭВМ (например, ЕС ЭВМ, IBM/370/ и др.) имеют единый **МЯ** для ЭВМ разной мощности. В команде любого из них сообщается информация о местонахождении операндов и типе выполняемой операции.

2.1.2. Языки Символического Кодирования

Языки Символического Кодирования (далее **ЯСК**), так же, как и **МЯ**, являются командными. Однако коды операций и адреса в машинных командах, представляющие собой последовательность двоичных (во внутреннем коде) или восьмеричных (часто используемых при написании программ) цифр, в **ЯСК** заменены на символы (идентификаторы), форма написания которых помогает программисту легче запоминать смысловое содержание операции. Это обеспечивает существенное уменьшение числа ошибок при составлении программ.

2.1.3. Автокоды

Есть также языки, включающие в себя все возможности **ЯСК**, посредством расширенного введения *макрокоманд* - они называются Автокоды.

Развитые автокоды получили название **Ассемблеры**. Сервисные программы и пр., как правило, составлены на языках типа **Ассемблер**

2.1.4. Макрос

Язык, являющийся средством для замены последовательности символов описывающих выполнение требуемых действий ЭВМ на более сжатую форму - называется **Макрос** (средство замены).

В основном, **Макрос** предназначен для того, чтобы сократить запись исходной программы. Компонент программного обеспечения, обеспечивающий функционирование макросов, называется **макропроцессором**.

2.2. Машинно – независимые языки

Машинно – независимые языки – это средство описания алгоритмов решения задач и информации, подлежащей обработке. Они удобны в использовании для широкого круга пользователей и не требуют от них знания особенностей организации функционирования ЭВМ и ВС.

Подобные языки получили название **высокоуровневых языков программирования**. Программы, составляемые на таких языках, представляют собой последовательности операторов, структурированные согласно правилам рассматривания языка(задачи, сегменты, блоки и т.д.). Операторы языка описывают действия, которые должна выполнять система после трансляции программы на **МЯ**.

Программист получил **возможность не расписывать в деталях вычислительный процесс на уровне машинных команд, а сосредоточиться на основных особенностях алгоритма**.

2.2.1. Проблемно – ориентированные языки

С расширением областей применения вычислительной техники возникла **необходимость формализовать представление постановки и решение новых классов задач**. Необходимо было создать такие языки программирования, которые, используя в данной области обозначения и терминологию, позволили бы описывать требуемые алгоритмы решения для поставленных задач, ими стали **проблемно – ориентированные языки**. Эти языки, языки ориентированные на решение определенных проблем, должны обеспечить программиста средствами, позволяющими коротко и четко формулировать задачу и получать результаты в требуемой форме. Проблемных языков очень много, например: **Фортран, Алгол** – языки, созданные для решения математических задач;

Simula, Слэнг - для моделирования; **Лисп, Снобол** – для работы со списочными структурами.

- **2.2.2. Универсальные языки**

- **Универсальные языки** были созданы для широкого круга задач: **коммерческих, научных, моделирования** и т.д. Первый универсальный язык был разработан фирмой IBM, ставший в последовательности языков **Пл/1**. Вторым по мощности универсальный язык называется **Алгол-68**. Он позволяет работать с символами, разрядами, числами с фиксированной и плавающей запятой. **Пл/1** имеет развитую систему операторов для управления форматами, для работы с полями переменной длины, с данными организованными в сложные структуры, и для эффективного использования каналов связи. Язык учитывает включенные во многие машины возможности прерывания и имеет соответствующие операторы. Предусмотрена возможность параллельного выполнения участков программ.
- Программы в **Пл/1** компилируются с помощью автоматических процедур. Язык использует многие свойства **Фортрана, Алгола, Кобола**. Однако он допускает не только динамическое, но и управляемое и статистическое распределения памяти.

- **2.2.3. Диалоговые языки**

- Появление новых технических возможностей поставило задачу перед системными программистами — **создать программные средства, обеспечивающие оперативное взаимодействие человека с ЭВМ** их назвали **диалоговыми языками**.
- Эти работы велись в двух направлениях. Создавались **специальные управляющие языки** для обеспечения оперативного воздействия на прохождение задач, которые составлялись на любых ранее неразработанных (не диалоговых) языках. Разрабатывались также языки, которые **кроме целей управления обеспечивали бы описание алгоритмов решения задач**.

Необходимость обеспечения оперативного взаимодействия с пользователем потребовала сохранения в памяти ЭВМ копии исходной программы даже после получения объектной программы в машинных кодах. При внесении изменений в программу с использованием диалогового языка система программирования с помощью специальных таблиц устанавливает взаимосвязь структур исходной и объектной программ. Это позволяет осуществить требуемые редакционные изменения в объектной программе.

Одним из примеров диалоговых языков является **Бэйсик**.

Бэйсик использует обозначения подобные обычным математическим выражениям. Многие операторы являются упрощенными вариантами операторов языка **Фортран**. Поэтому этот язык позволяет решать достаточно широкий круг задач.

2.2.4. Непроцедурные языки

составляют группу языков, описывающих организацию данных, обрабатываемых по фиксированным алгоритмам (табличные языки и генераторы отчетов), и языков связи с операционными системами.

Позволяя четко описывать как задачу, так и необходимые для её решения действия, таблицы решений дают возможность в наглядной форме определить, какие условия должны быть выполнены прежде чем переходить к какому-либо действию. Одна таблица решений, описывающая некоторую ситуацию, содержит все возможные блок-схемы реализаций алгоритмов решения.

Табличные методы легко осваиваются специалистами любых профессий.

Программы, составленные на табличном языке, удобно описывают сложные ситуации, возникающие при системном анализе.