Курс лекций «ЭВМ и периферийные устройства» («Архитектура вычислительных машин и систем»)

Лектор - доцент, к.т.н.

Кузьмин Александр Александрович

Введение. Часть 1.

Определения важнейших понятий организации и архитектуры ЭВМ

ЭВМ и информация

Электронная вычислительная машина (компьютер) – автоматическое устройство, назначение которого состоит в выполнении формализованных манипуляций над информацией.

Почему абак, счёты, арифмометр и даже калькулятор не являются компьютерами?

Определение может быть и недостаточно строгое, но очень «плотное», так как указывает на важнейшие понятия: автоматизацию и информацию.

Что такое «информация»? Можете ли Вы привести определение этого понятия?

(Ознакомьтесь с приложением П0.1.)

Определения из основной литературы

Термином вычислительная машина будем обозначать комплекс технических и программных средств, предназначенный для автоматизации подготовки и решения задач пользователя. (Учебник Б.Я.Цилькера и С.А.Орлова)

Вычислительная машина – это система, выполняющая заданную в программе, четко определенную последовательность операций в соответствии с выбранным алгоритмом обработки информации. (Учебник В.Ф.Мелехина и Е.Г.Павловского)

Основой построения (появления) ЭЦВМ явилась следующая совокупность (последовательность!) формализаций:

- множество задач (по-видимому, все задачи обработки информации) может быть решено в численном виде; ход их решения представляет последовательность сравнительно простых действий над числами;
- числа могут быть представлены в различных системах счисления и зашифрованы (закодированы);
- действиям над числами соответствуют логические преобразования над кодами;
- коды могут быть представлены в виде дискретных электрических сигналов;
- разработаны методы логического преобразования кодов, в виде обработки электрических сигналов, однозначно соответствующие действиям над числами.

Определения алгоритма и вычислимость

Понятие «алгоритм» столь же фундаментально, как и «информация» и столь же неопределимо. Но попытки дать определения неизбежны, вот одна из них:

- алгоритм – это точное общепонятное предписание, определяющее процесс преобразования исходных данных в искомый результат.

(Ознакомьтесь с приложением П0.2.)

Работы Тьюринга и Поста были связаны с проблемой формальной вычислимости функций. В качестве решения этой проблемы был сформулировано положение, получившее наименование тезис Чёрча.

(Ознакомьтесь с приложением П0.3.)

Функция Аккермана – тест для компьютера

Функция Аккермана индуктивно задана на парах неотрицательных целых чисел:

$$A(0,n) = n+1; \qquad A(m+1,0) = A(m,1);$$

$$A(m+1,n+1) = A(m, A(m+1,n)),$$

где m, n больше или равно 0.

Можно получить следующие аналитические выражения:

$$A(1,n) = n+2; \qquad A(2,n) = 2n+3; \qquad A(3,n) = 2^{n+3} - 3; \qquad \ldots$$

Высокая рекурсивность этой функции используется для проверки компиляторов и вообще компьютеров выполнять рекурсию. Функция очень быстро возрастает по мере увеличения m, но ещё существеннее факт сложности (продолжительности) рекурсивного вычисления.

Функция Аккермана одной переменной:

$$Ack(n) = A(n,n)$$

Функция Аккермана (продолжение)

Функция Аккермана — пример всюду определённой вычислимой функции, которая не является примитивно рекурсивной. Она принимает два неотрицательных целых числа в качестве параметров и возвращает натуральное число, обозначается A(m, n). Эта функция растёт очень быстро, например, число

$$A(4,4) = 2^{2^{10^{19729}}}$$

настолько велико, что ***количество цифр в записи этого числа*** многократно превосходит количество атомов в наблюдаемой части Вселенной.

Функция Аккермана в силу своего определения имеет очень глубокий уровень рекурсии, что можно использовать для проверки способности компилятора оптимизировать рекурсию. Вычисление A(1, n) занимает линейное время n. Вычисление A(2, n) требует квадратичное время. Время вычисления A(3, n) пропорционально 4 в степени n+1.

Значение A(4, 2) невозможно посчитать с помощью простого рекурсивного применения за разумное время.

(Подробнее свойства этой функции описаны, например, в Википедии.)

Свойства алгоритмов

Существенные (определяющие) свойства:

- **Определенность** (однозначность), не оставляющая места произволу.
- **Массовость**, т.е. возможность исходить из любых исходных данных.
- **Результативность**, т.е. свойство так определять процесс, который всегда приводит к искомому результату за конечное число шагов.

Основные требования к записи алгоритмов:

- наглядность;
- обозримость;
- однозначность толкования;
- обеспечение различной степени детализации.

Среди прочих используются следующие формы представления алгоритма:

- Текстуальная.
- Операторная.
- Графическая.
- В терминах алгоритмического языка.

(Существуют ли другие формы записи алгоритмов?)

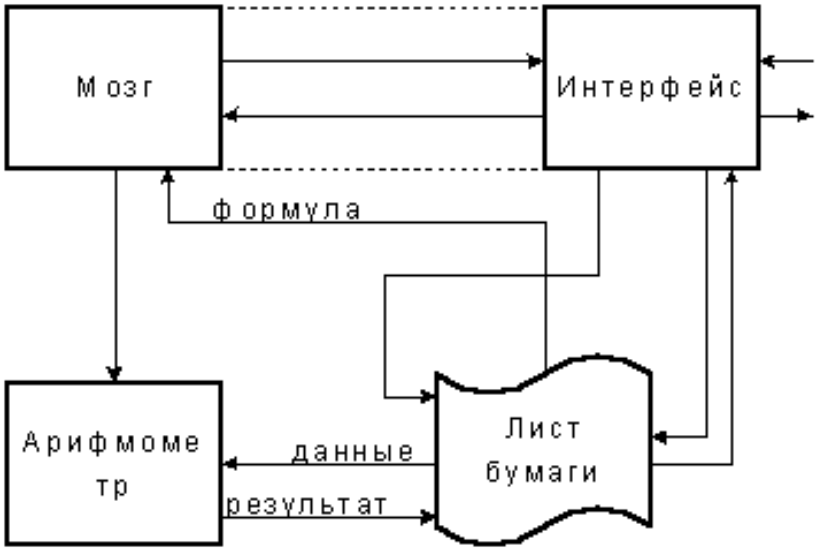
Определение ЭВМ

Техническое устройство, способное распознавать и интерпретировать формальный язык, на котором записан алгоритм, и выполнять предписанные действия, (реализовывать алгоритмические конструкции, операторы программы), и есть вычислительная машина.

(Ознакомьтесь с приложением П0.4.)

Модель вычислений

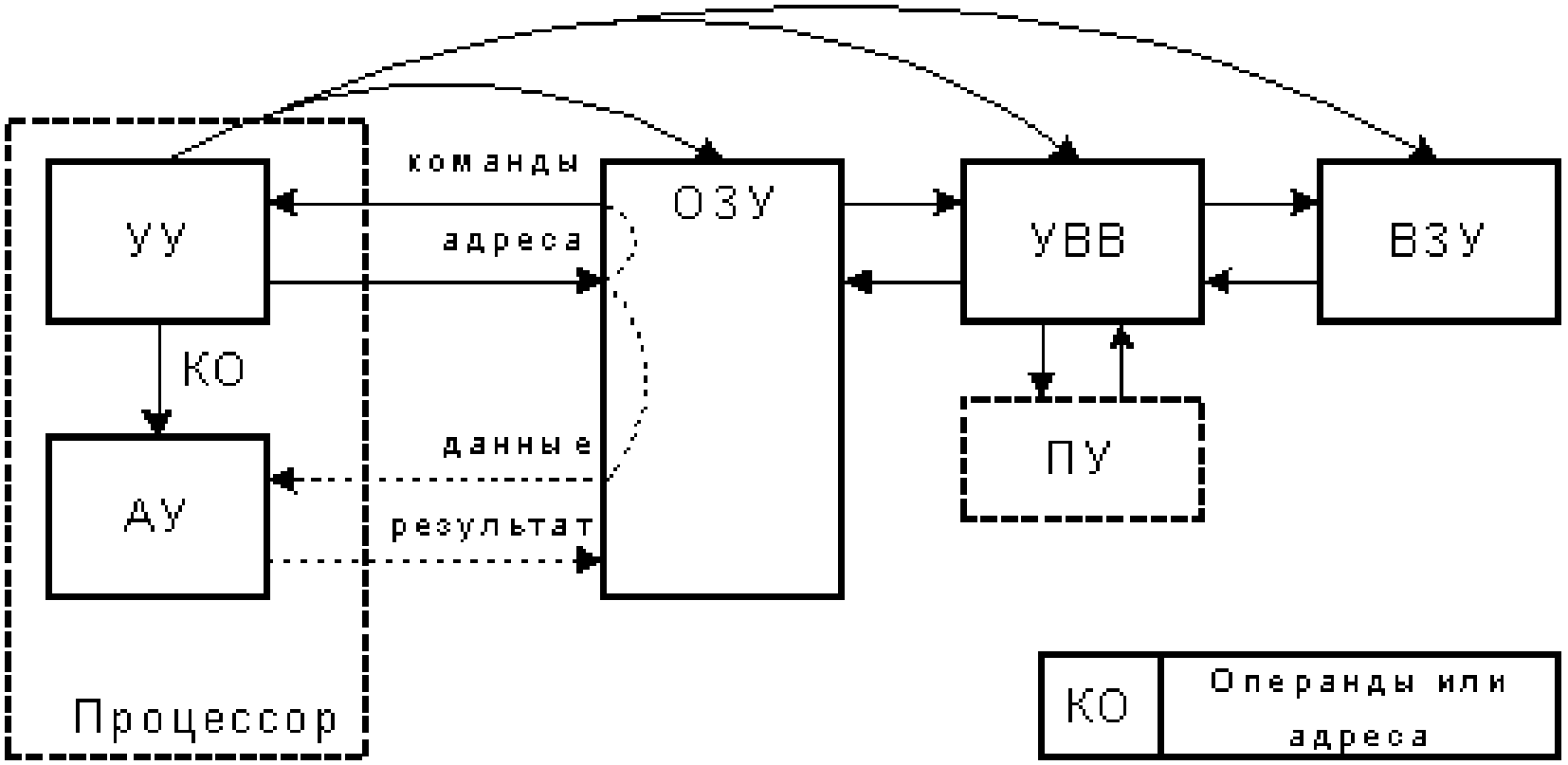
На лист бумаги заносим формулу вычислений, определяющую порядок действий, исходные данные и заготавливаем, например, таблицу для будущих результатов.



Аналогия со столь формальным процессом вычисления наблюдается в работе ЭВМ. При этом заметим, что технические средства должны обеспечивать по крайней мере следующее:

- **хранение и выдачу исходных, промежуточных данных и результатов;**
- **хранение порядка действий;**
- **обмен с внешним миром, в частности с человеком, решающим задачу;**
- **непосредственное выполнение отдельных операций;**
- **управление (координацию) всем ходом процесса.**

Удовлетворять этим требованиям может следующая структура:

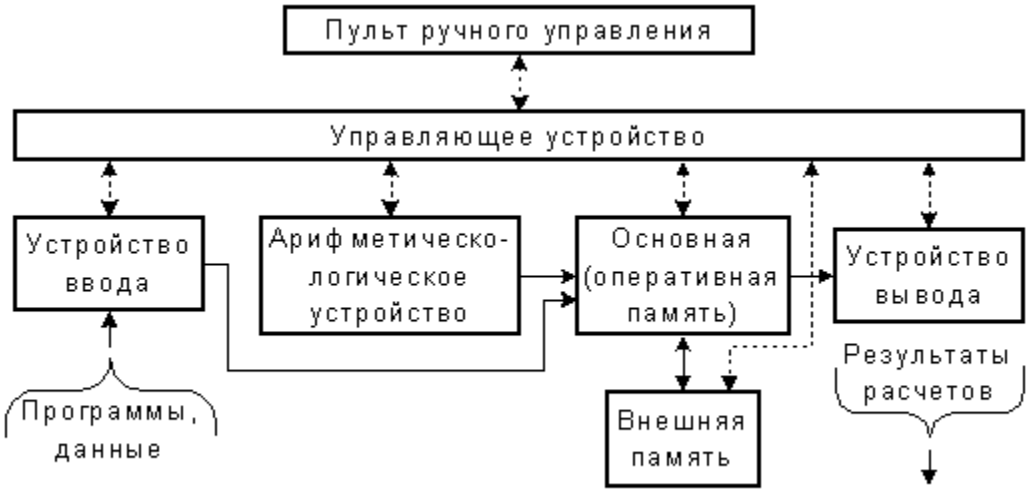


Базовая организация ЭВМ

Любая универсальная ЭВМ состоит, таким образом, из четырех основных частей (по фон Нейману – из 5-ти):

- **Арифметико–логическое устройство (АЛУ), предназначенное для непосредственного выполнения отдельных операций над кодами чисел (операндами).**
- **Запоминающее устройство (ЗУ), предназначенное для приема, хранения и вывода кодов чисел и программ. Система памяти ЭВМ строится по иерархическому принципу СОЗУ – ОЗУ – ВЗУ.**
- **Устройство управления (УУ), предназначенное для автоматического (без участия человека) управления работой всех других блоков в процессе вычислений. Обычно тоже строится по иерархии.**
- **Устройство ввода-вывода (УВВ), предназначенное для восприятия вводимых данных, представляемых, в частности, и в каком-либо «неэлектронном виде» (оптические свойства на CD и DVD, пробивки на перфокарте, перфоленте и др.), преобразования их в электрические сигналы и передачи их в ЗУ, а также для обратных процедур.**

(Фон Нейман устройства ввода и вывода рассматривал как два независимых блока).



Базовая организация ЭВМ (продолжение)

Основные характеристики фон-неймановской архитектуры:

- **наличие единого вычислительного устройства, включающего процессор, средства передачи информации и память;**
- **линейная структура адресации памяти, состоящей из слов фиксированной длины;**
- **низкий уровень машинного языка, команды которого осуществляют простые операции над элементарными операндами;**
- **централизованное последовательное управление.**

Процессор – законченный блок ЭВМ, в состав которого входят УУ, АЛУ и отдельные регистры. Он объединяет в себе оборудование, посредством которого программа интегрируется в вычислительный процесс.

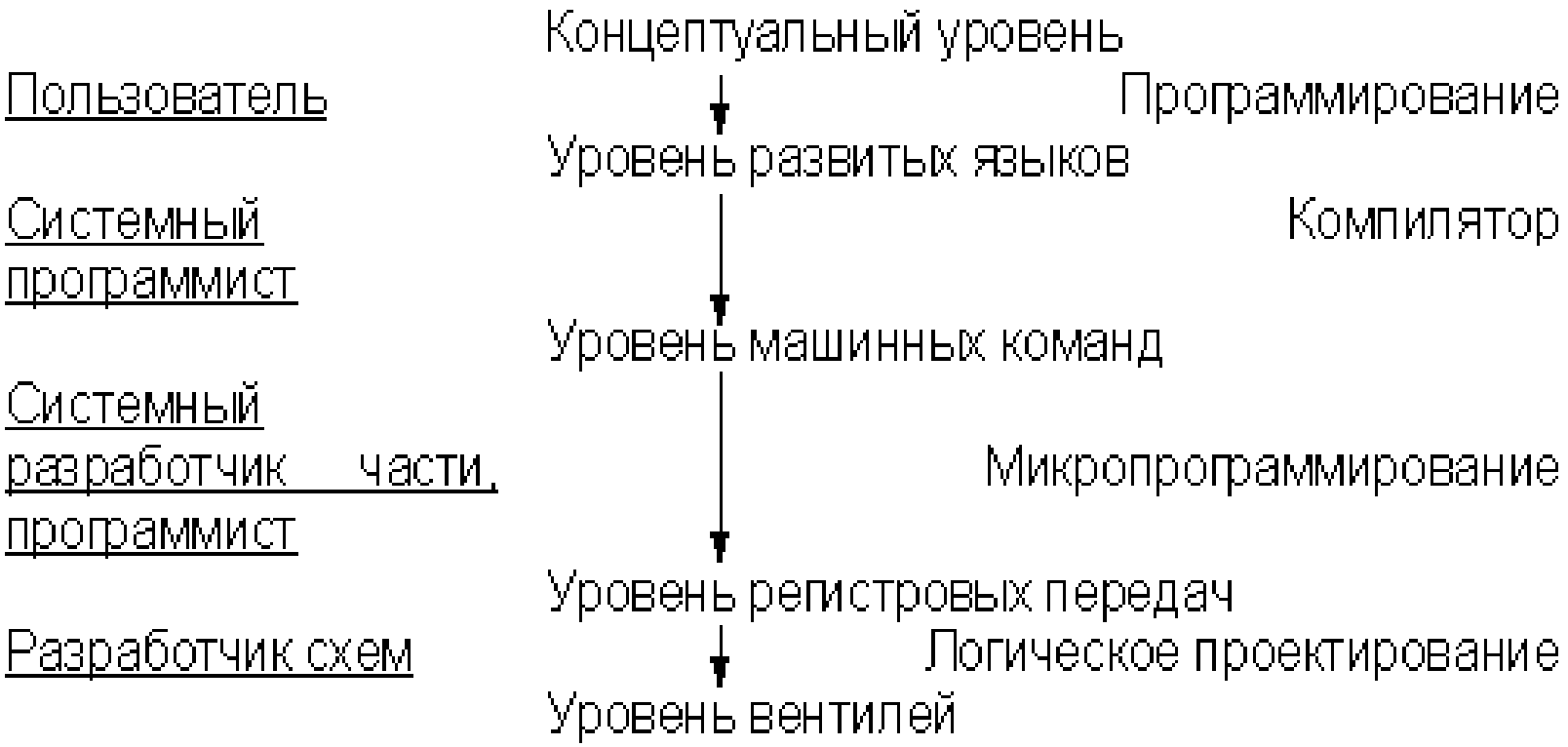
Архитектура – представление аппаратной и программной составляющих частей системы и взаимосвязи между ними как единого целого в противоположность рассмотрению компонентов системы по отдельности.

Гарвардская архитектура – фон-неймановская архитектура, усовершенствованная путем введения раздельных трактов команд и данных. (Используется для устранения затруднений из-за общей памяти). Альтернатива – Принстонская архитектура.

Команда (машинная команда) – кодированное предписание, определяющее шаг процесса выполнения программы; содержит указание выполнения операции, адрес (адреса) операндов и служебные признаки.

Многоуровневая модель вычислительных процессов

С позиций использования ЭВМ и инструментальных средств проектирования вычислений оказывается возможным ввести несколько уровней абстрагирования представления разработки:



Семантический разрыв

Основу существования этого **неизбежного явления** составляет противоречие между языками программирования (ЯП) и архитектурными принципами организации ЭВМ. Это противоречие между средой использования и желаемой простотой организации обработки «двоичной информации», это смысловой (семантический) разрыв между теми средствами, которые используются для представления решаемых задач и теми средствами, что мы вынуждены использовать при реализации этих представлений.

Характерные особенности архитектуры современных ЭВМ (регистры общего назначения, индексные регистры, данные в форме с фиксированной и плавающей запятой, косвенная адресация, прерывания, асинхронный ввод-вывод, виртуальная память и т.д.) не изменились, а усугубились с 60-х годов.

Современные высокоуровневые ЯП стали ещё более формализованными и ориентированными на математические методы пользователя.

Следствия: сложность ОС, трансляторов и компиляторов, большой объём, высокая стоимость разработки и ненадежность прикладного ПО и т.д.

О классификации ЭВМ, комплексов и систем

Существуют многочисленные способы классификации ЭВМ, комплексов и систем. Большинство классификаций имеют весьма относительную ценность, поскольку все они выбирают отдельные критерии. Среди всех выделим только две классификации.

Первая связана с существованием трёх способов инициирования выполнения машинных команд:

1) в порядке расположения в программе, последовательно;
2) по мере готовности данных;
3) по мере потребности в результате выполнения.

Соответственно вычислительные системы классифицируются:

1) с логически-программным управлением;
2) с управлением потоками данных;
3) с редукционно-программным управлением.

Флинн М. (*Michael J. Flynn)* в 1966 году ввел и далее в 1972-м уточнил описание ЭВМ и вычислительных систем с точки зрения потоков команд (инструкций) и данных. Он также ввел понятия одиночного и множественного потоков, как для команд, так и для данных. Это привело к распределению всех архитектур ВМ и систем по 4-м классам:

	Одиночный поток данных (ОД)	Множественный поток данных (МД)
Одиночный поток команд (ОК)	ОКОД Однопроцессорная система (в 1972 году – IBM 360)	ОКМД Параллельный процессор и ассоциативный процессор (… - ILLIAC IV)
Множествен-ный поток команд (МК)	МКОД – «не бывает» МКОД-К Процессор поточной обработки (конвейерная система) (… - CDC STAR 100)	МКМД Многопроцессорная или многомашинная вычислительная система (… - Univac 1108)

С появлением и в начальный период применения в компьютерах микропроцессоров (1970 – 80 гг.) широко использовалась классификация архитектур вычислительных устройств по месту хранения операндов: *стековая; аккумуляторная; регистровая; с выделенным доступом к памяти*. В настоящее время грани между этими различиями архитектуры процессоров стерлись.

Одним с способов преодоления семантического разрыва стало появление в начале 80-х годов и используемой и в настоящее время особой **RISC**-архитектуры процессоров. Следствием успеха этой концепции было появление несколько искусственной классификации:

архитектура с сокращенным набором команд: RISC (Reduced (Restricted) Instruction Set Computer);

архитектура с полным набором команд: CISC (Complex Instruction Set Computer); так стали именовать «традиционную» ранее использовавшуюся организацию процессоров в противовес RISC;

архитектура с командными словами сверхбольшой длины: VLIW (Very Long Instruction Word), как развитие RISC-архитектуры.

Основные принципы RISC-архитектур

Основной набор команд не должен интерпретироваться микро-командами, а должен выполняться аппаратным обеспечением.

Все **команды должны иметь одинаковую длину и минимальное число форматов** (обычно не более 2–3), это упрощает логику управления при выборе и при исполнении команды.

Любая **команда** основного набора **должна выполняться за один машинный цикл** (стандартом является команда сложения регистра с регистром); это достигается одновременным (параллельным) выполнением команд путем конвейеризации или использования нескольких обрабатывающих узлов.

Обращение к памяти производиться **только по специально выделенным командам** работы с памятью (Load – загрузка и Store – сохранение), а вся обработка данных должна вестись в регистровом формате; при этом количество регистров может быть велико.

Система команд должна обеспечивать поддержку компиляции с конкретного языка программирования (компиляторы для RISC на порядок сложнее, чем компиляторы для CISC).

Принципы построения универсальных ЭВМ

Работы фон Неймана, позволили сформулировать основные **принципы** построения универсальных ЭЦВМ:

- **Принцип программного управления, как основа построения автоматически действующих ЭВМ.**
- **Принцип хранимой программы (наравне с числами). Это позволяет производить над ней действия, как над числами; обеспечивается высокая скорость управления вычислительным процессом; удобно и просто менять программы (более гибки и универсальны).**
- **Использование многоуровневой (иерархической) памяти.**
- **Использование двоичной системы счисления для представления информации и программ. Это же позволяет применить аппарат булевой алгебры и свести все логические и арифметические операции к простейшим операциям.**
- **Принцип условного перехода. Переход обеспечивает независимость программ от исходных данных, т.е. делает программы более гибкими и универсальными.**

Состав программного обеспечения ЭВМ

Программное обеспечение (ПО) — это совокупность программ, позволяющих осуществить на компьютере автоматизированную обработку информации. ПО делится на системное (общее) и прикладное (специальное).

Системное ПО обеспечивает функционирование и обслуживание компьютера, а также автоматизацию процесса создания новых программ. К системному ПО относятся: операционные системы и их пользовательский интерфейс; инструментальные программные средства; системы технического обслуживания.

Операционная система (ОС) — обязательная часть специального ПО, обеспечивающая эффективное функционирование персонального компьютера в различных режимах, организующая выполнение программ и взаимодействие пользователя и внешних устройств с ЭВМ.

Пользовательский интерфейс (сервисные программы) — это программные надстройки ОС (оболочки и среды), предназначенные для упрощения общения пользователя с ОС.

Состав программного обеспечения ЭВМ (продолжение)

Интерфейсные системы — это сервисные системы ПО, чаще всего графического типа, совершенствующие не только пользовательский, но и программный интерфейс ОС, в частности, реализующие некоторые дополнительные процедуры разделения дополнительных ресурсов.

Утилиты автоматизируют выполнение отдельных типовых, часто используемых процедур, реализация которых потребовала бы от пользователя разработки специальных программ.

Инструментальные программные средства (ИПС) (системы программирования) — обязательная часть ПО, с использованием которой создаются программы. ИПС включают в свой состав средства написания программ (текстовые редакторы); средства преобразования программ в вид, пригодный для выполнения на компьютере (ассемблеры, компиляторы, интерпретаторы, загрузчики и редакторы связей), средства контроля и отладки программ.

Состав программного обеспечения ЭВМ (продолжение)

Пакет прикладных программ (ППП) — это совокупность программ для решения круга задач по определенной тематике или предмету.

Различают следующие типы ППП:

1. Общего назначения — ориентированы на автоматизацию задач пользователя (текстовые процессоры, табличные редакторы, системы управления базами данных, графические процессоры, издательские системы, системы автоматизации проектирования и т. д.).

2. Метод ориентированные — реализация экономико-математических методов решения задач (математического программирования, сетевого планирования и управления, теории массового обслуживания, математической статистики и т. д.).

3. Проблемно ориентированные — направленные на решение определенной задачи (проблемы) в конкретной предметной области (банковские пакеты, пакеты бухгалтерского учета, финансового менеджмента, правовых справочных систем и т. д.).

К **прикладному ПО** относятся сервисные программные средства, которые служат для организации удобной рабочей среды пользователя, а также для выполнения вспомогательных функций (информационные менеджеры, переводчики и т. д.).