

Инструментальная база информационных технологий

- | | | |
|---------------------------------------|-----------------|-----------------------|
| 1. Программные
технологий | средства | информационных |
| 2. Технические
технологий | средства | информационных |
| 3. Методические
технологий | средства | информационных |

Информационные технологии функционируют на основе инструментальной базы, включающей программные, технические и методические средства.



1. Программные средства информационных технологий

Программные средства информационных технологий можно разделить на две большие группы: **базовые** и **прикладные**.

Базовые программные средства относятся к инструментальной страте информации информационных технологий и включают в себя:

- операционные системы (ОС);
- языки программирования;
- программные среды;
- системы управления базами данных (СУБД).

Прикладные программные средства предназначены для решения комплекса задач или отдельных задач в различных предметных областях.

ОС предназначены для управления ресурсами ПК (ЭВМ) и процессами, использующими эти ресурсы. В настоящее время существуют две основные линии развития ОС: Windows и Unix. Генеалогические линии данных ОС развивались следующим образом:

1. CP/M → QDOS → 86-DOS → MS-DOS → Windows.
2. Multics → UNIX → Minix → Linux.

Каждый элемент линии имеет свое развитие, например, Windows развивался в такой последовательности: Windows 95, 98, Me, NT, 2000, Vista, XP, Win 7, Win 8, Win 10.

Соответственно Linux развивался следующим образом: версии 0.01, 0.96, 0.99, 1.0, 1.2, 2.0, 2.1, 2.1.10, 2.4, 2.6, 3.x.

Каждая версия может отличаться добавлением новых функциональных возможностей (сетевые средства, ориентация на разные процессоры, многопроцессорные конфигурации и др.).

Большинство **алгоритмических языков программирования** (Си, Паскаль) созданы на рубеже 60 – 70-х годов (за исключением Java). Позже, периодически появлялись новые языки программирования, однако на практике они не получили широкого и продолжительного распространения. Другим направлением в эволюции современных языков программирования были попытки создания универсальных языков (Алгол, PL/1, Ада), объединяют в себе достоинства ранее разработанных.

Появление ПК и ОС с графическим интерфейсом (Mac OS, Windows) привело к смещению внимания разработчиков программного обеспечения в сферу визуального или объектно-ориентированного программирования, сетевых протоколов, баз данных. Это привело к тому, что в настоящее время в качестве инструментальной среды используется конкретная среда программирования (Delphi, Access и др.) и знания базового языка программирования не требуется. Можно считать, что круг используемых языков программирования стабилизировался..

Анализ синтаксиса и семантики языков программирования показывает, что их родственные конструкции различаются главным образом «внешним видом» (набором ключевых слов или порядком следования компонентов). Содержимое практически идентично, за исключением небольших различий, не имеющих существенного значения. Таким образом, конструкции современных языков имеют общее содержание (семантику), различный порядок следования компонент (синтаксис) и разные ключевые слова (лексику). Следовательно, различные языки предоставляют пользователю одинаковые возможности при различном внешнем виде программ.

Стандартизацию языков программирования в настоящее время осуществляют комитеты ISO/ANSI, однако их деятельность направлена в основном на неоправданное синтаксическое расширение языков. Для исключения существующих недостатков предложены способы задания семантического и синтаксического стандартов языков программирования

Семантическое описание любой конструкции языка (оператор, тип данных, процедура и т.д.) должно содержать не менее трех обязательных частей:

- список компонент (в Типе указателя это компоненты Имя типа и Базовый тип);
- описание каждой компоненты;
- описание конструкции в целом.

Для синтаксического описания обычно используется формальное описание конструкции, например, в виде БНФ. Синтаксическое описание присутствует в любом языке, начиная с Алгола.

Среди большого числа языков самую заметную роль в развитии программирования сыграли три пары: Алгол-60 и Фортран, Паскаль и Си, Java и Си++. Эти языки не случайно объединены в пары, так как противостояние заложенных в них идей способствовало прогрессивному развитию.

Важно различать язык программирования и его реализацию. Сам язык – это система записи, набор правил, определяющих синтаксис и семантику программы. Реализация языка – это программа, которая преобразует запись высокого уровня в последовательность машинных команд. Существуют два способа реализации языка: **компиляция** и **интерпретация**.

При **компиляции** специальная рабочая программа (компилятор) осуществляет перевод рабочей программы в эквивалентную на машинном коде и в дальнейшем ее выполнение совместно с данными. В методе интерпретации специальная программа (интерпретатор) устанавливает соответствие между языком и машинными кодами, применяя команды к данным. Любой язык программирования может быть как интерпретируемым, так и компилируемым, но в большинстве случаев есть свой предпочтительный способ реализации. К сожалению, в настоящее время не существует универсального компилятора, который мог бы работать с любым существующим языком. Это объясняется отсутствием единой семантической базы. Хотя современные языки программирования похожи друг на друга, идентичность их далеко не полная. Таким образом, существует общая семантическая зона, в которую входят конструкции, принадлежащие всем языкам программирования (или большинству из них), и область объединения, содержащая конструкции специфические для данного языка, поэтому создание универсального компилятора возможно двумя путями:

1.Использование общих конструкций (область пересечения), исключение специфических конструкций языков (область объединения). Это приведет к обеднению всех языков программирования.

2.Использование всех имеющихся конструкций (область объединения + область пересечения). Такой подход приведет к значительному расширению семантической базы и использованию дополнительных ресурсов.

С точки зрения информационных технологий программирование имеет промышленный характер, который соответствует традиционным стадиям жизненного цикла программного продукта:

- анализ требований;
- разработка спецификаций;
- проектирование;
- макетирование;
- написание исходного текста;
- отладка;
- документирование;
- тестирование и сопровождение.

Наряду с этим направлением развивается так называемое исследовательское программирование.

Программные среды реализуют отдельные задачи и операции информационных технологий. К их числу относятся:

1. Текстовые процессоры.
2. Электронные таблицы.
3. Личные информационные системы.
4. Программы презентационной графики.
5. Браузеры.
6. Редакторы WEB-страниц.
7. Почтовые клиенты.
8. Редакторы растровой графики.
9. Редакторы векторной графики.
10. Настольные издательские системы.
11. Средства разработки.

2. Технические средства информационных технологий

Компьютеры являются ядром любой информационной системы. Первоначально они были созданы для реализации большого объема вычислений, представляющих длинные цепочки итераций. Главным требованием при этом были высокая точность и минимальное время вычислений. Такие процессы характерны для числовой обработки.

По мере внедрения ЭВМ, их эволюционного развития, стали возникать другие области применения, отличные от вычислений, например, обработка экономической информации, создание информационно-справочных систем, автоматизация учрежденческой деятельности и т. п. В данном случае не требовались высокая точность и большой объем вычислений, однако объем обрабатываемой информации мог достигать миллионов и миллиардов записей. При этом требовалось не только обработать информацию, а предварительно ее найти и организовать соответствующую процедуру вывода. Указанные процессы характерны для нечисловой обработки, требующей в большинстве случаев больших затрат машинного времени. Рассмотренные аспекты оказали решающее влияние на развитие архитектуры ЭВМ.

ЭВМ классической (фоннеймановской) архитектуры состоит из пяти основных функциональных блоков:

- запоминающего устройства (ЗУ);
- устройства управления;
- устройств управления и арифметически-логического устройства, рассматриваемых вместе и называемых центральным процессором;
- устройства ввода;
- устройства вывода.

В фоннеймановской архитектуре для обработки огромного объема информации (миллиарды байт) используется один процессор. Связь с данными осуществляется через канал обмена. Ограничения пропускной способности канала и возможностей обработки в центральном процессоре приводят к тупиковой ситуации при нечисловой обработке в случае увеличения объемов информации. Для выхода из тупика было предложено два основных изменения в архитектуре ЭВМ:

- использование параллельных процессоров и организация параллельной обработки;
- распределенная логика, приближающая процессор к данным и устраняющая их постоянную передачу.

Другой недостаток фоннеймановской архитектуры связан с организацией процесса обращения к ЗУ, осуществляемого путем указания адреса для выборки требуемого объекта из памяти. Это приемлемо для числовой обработки, но при нечисловой обработке обращение должно осуществляться по содержанию (ассоциативная адресация). Поскольку для нечисловой обработки в основном используется та же архитектура, необходимо было найти способ организации ассоциативного доступа. Он осуществляется путем создания специальных таблиц (справочников) для перевода ассоциативного запроса в соответствующий адрес. При такой организации обращения к ЗУ, называемом эмуляцией ассоциативной адресации, в случае работы с большими объемами информации резко падает производительность ЭВМ. Это связано с тем, что нечисловая обработка – это не только просмотр, но и обновление данных.

Для преодоления ограничений организации памяти были предложены ассоциативные запоминающие устройства.

Таким образом, ЭВМ для нечисловой обработки должна удовлетворять следующим требованиям: **ассоциативность, параллелизм, обработка в памяти.**

Кроме этого на более высоком уровне к архитектуре предъявляются следующие требования:

- перестраиваемость параллельных процессоров и запоминающих устройств;
- сложные топологии соединений между процессорами;
- мультипроцессорная организация, направленная на распределение функций.

Перечисленные выше ограничения и требования были реализованы в машинах баз данных (МБД).

Приведем классификацию архитектур ЭВМ:

- архитектура с одиночным потоком команд и одиночным потоком данных (SISD);
- архитектура с одиночным потоком команд и множественным потоком данных (SIMD);
- архитектура с множественным потоком команд и одиночным потоком данных (MISD);
- архитектура с множественным потоком команд и множественным потоком данных (MIMD).

К **классу SISD** относятся современные фоннеймановские однопроцессорные системы. В этой архитектуре центральный процессор работает с парой «атрибут – значение». Атрибут (метка) используется для локализации соответствующего значения в памяти, а одиночная команда, обрабатывающая содержимое накопителя (регистра) и значение – результат. В каждой итерации из входного потока данных используется только одно значение.

К **классу SIMD** относят большой класс архитектур, основная структура которых состоит из одного контроллера, управляющего комплексом одинаковых процессоров. В зависимости от возможностей контроллера и процессорных элементов, числа процессоров, организации поиска и характеристик маршрутных и выравнивающих сетей выделяют четыре типа SIMD:

- *матричные процессоры*, организованы так, что при выполнении заданных вычислений, инициированных контроллером, они работают параллельно. Предназначены для решения векторных и матричных задач, относящихся к числовой обработке;

- *ассоциативные процессоры*, обеспечивающие работу в режиме поиска по всему массиву за счет соединения каждого процессора непосредственно с его памятью. Используются для решения нечисловых задач;
- *процессорные ансамбли*, представляющие совокупность процессоров, объединенных определенным образом для решения заданного класса задач, ориентированных на числовую и нечисловую обработку;
- *конвейерные процессоры* (последовательные и векторные) осуществляющие выполнение команд и обработку потоков данных по принципу, аналогичному транспортному конвейеру. В этом случае каждый запрос использует одни и те же ресурсы. Как только некоторый ресурс освобождается, он может быть использован следующим запросом, не ожидая окончания выполнения предыдущего. Если процессоры выполняют аналогичные, но не тождественные задания, то это последовательный конвейер, если все задания одинаковы – векторный конвейер.

К ~~классу MISD~~ может быть отнесена единственная архитектура конвейер, но при условии, что каждый этап выполнения запроса является отдельной командой.

К **классу MIMD**, хотя и не всегда однозначно, относят следующие конфигурации:

- мультипроцессорные системы;
- системы с мультиобработкой;
- вычислительные системы из многих машин;
- вычислительные сети.

Общим для данного класса является наличие ряда процессоров и мультиобработки. В отличие от параллельных матричных систем число процессоров невелико, а термин «мультиобработка» используют для обозначения функционально распределенной обработки (сортировки, слияния, ввода-вывода и др.)

Другим направлением развития вычислительной техники является нейрокомпьютеринг, основанный на нейронных сетях. Разработки проводятся в двух направлениях: аппаратном и программном. Нейрокомпьютеры обладают сверхвысокой производительностью, но благодаря сложным технологиям имеют очень высокую стоимость, поэтому они используются узким кругом пользователей для решения суперзадач.

В последние годы ведутся работы по созданию биокомпьютера на основе молекулярных технологий. Идея молекулярного вычислителя состоит в представлении «машинного» слова в виде состояний молекул. Несмотря на развитие средств вычислительной техники, наиболее популярными в настоящее время остаются компьютеры с традиционной фоннеймановской архитектурой. ЭВМ такой архитектуры в процессе эволюции последовательно прошли этапы аппаратной реализации от электронно-ламповой, далее транзисторной, интегрально-схемной до СБИС. В настоящее время наиболее распространенным типом ЭВМ являются персональные компьютеры (ПК), относящиеся к фоннеймановской архитектуре.

С точки зрения рынка аппаратных средств информационных технологий их можно разделить на три группы: компьютеры, сетевые средства, средства оргтехники. К распространенным аппаратным средствами относятся:

1. Настольные компьютеры (отечественной сборки и зарубежного производства).
2. Ноутбуки (переносные компьютеры).
3. Карманные компьютеры.
4. Процессоры.

5. Графические станции.
6. Мониторы.
7. Принтеры (струйные, лазерные и светодиодные).
8. Сканеры.
9. Системные платы.
10. Видеоадаптеры.
11. Звуковые платы.
12. Модемы.
13. Дисководы.
14. Дисководы на съемных носителях.
15. Внешние переносные дисководы.
16. Цифровые камеры.
17. Мыши.
18. Портативные MP3-плееры.
19. Платы для видеомонтажа.
20. TV-тюнеры.

3. Методические средства информационных технологий

Для большинства технологий характерной чертой их развития является стандартизация и унификация.

Стандартизация – нахождение решений для повторяющихся задач и достижение оптимальной степени упорядоченности.

Унификация – относительное сокращение разнообразия элементов по сравнению с разнообразием систем, в которых они используются.

Если в области традиционного материального производства уже давно сложилась система формирования и сопровождения стандартов, то в области информационных технологий многое предстоит сделать.

Главная задача стандартизации в рассматриваемой области – создание системы нормативно-справочной документации, определяющей требования к разработке, внедрению и использованию всех компонентов информационных технологий. На сегодняшний день в области информационных технологий наблюдается неоднородная картина уровня стандартизации. Для ряда технологических процессов характерен высокий уровень стандартизации (например для транспортирования информации), для других – он находится в зачаточном состоянии.

Многообразные стандарты и подобные им методические материалы делятся по следующим признакам:

1. По утверждающему органу:

- официальные международные стандарты;
- официальные национальные стандарты;
- национальные ведомственные стандарты;
- стандарты международных комитетов и объединений;
- стандарты фирм-разработчиков;
- стандарты «де-факто».

2. По предметной области стандартизации:

- функциональные стандарты (стандарты на языки программирования, интерфейсы, протоколы, кодирование, шифрование и др.);
- стандарты на фазы развития (жизненного цикла) информационных систем (стандарты на проектирование, материализацию, эксплуатацию, сопровождение и др.).

В зависимости от методического источника в качестве стандартов могут выступать метод, модель, методология, подход. Следует отметить, что указанные стандарты обладают разной степенью обязательности, конкретности, детализации, открытости, гибкости и адаптируемости.