

# **3. Средства информационных и коммуникационных технологий**

# **3.1 История компьютера**

Информационные проблемы появились у людей на самой ранней фазе их развития и сопровождают их по сей день. Нашим далеким предкам жизненно важно было знать, например, сколько еды нужно заготовить на случай холодной зимы, как поведет себя тот или иной хищник в разных условиях, какие условия обитания имеются в новых для них местах и т.д. Наверное, они разбирались в понятиях «равно», «не равно», «больше», «меньше», т.е. оперировали информационными признаками предметов, событий.

*Из истории известно, что с самых древних времен люди делали попытки как-то облегчить свою умственную деятельность. Например, зарубками, которые делались каменными топорами на стенах пещер, они отмечали количество убитых животных, фактически используя стену пещеры как запоминающее устройство. Огнем зажженных в условленных местах костров, звуками ударов передавали другим информацию о тех или иных событиях, причем на значительные расстояния.*



Чем больше развивался человек, тем большее количество информации ему было необходимо помнить, получать, передавать и обрабатывать. Как мы знаем, счет применяется тысячи лет. Первое устройство, механизующее счет, – счеты – создано задолго до начала нашей эры. Так, разновидность счетов – абак – использовалась в Древнем Египте и была знакома китайцам еще в VI в. до н.э. В самых ранних вычислительных устройствах большее внимание уделялось механизации арифметических операций, однако хранение результатов оказалось делом не менее важным. Функции «памяти» выполнял материал – глиняные дощечки, береста, а позднее бумага. Скорость операций была невелика, память использовалась медленно, и, главное, не было острой необходимости в поиске путей механизации управления последовательностью операций.

Прорыв в этой области произошел в ткацком деле. В 1801 г. в ткацком станке Жаккарда использовались перфокарты – кусочки картона с пробитыми в них отверстиями для управления командами, обеспечивающими автоматическое переплетение

в ткани на станке.



Английский математик Чарльз Бэббидж (1792 – 1871), исследуя астрономические таблицы, обнаружил в них множество ошибок, вызванных неточностью расчетов. Он понял, что абсолютно точные таблицы можно получить только с помощью расчетов, выполненных машиной, которая получает описание необходимых действий от человека, выполняет их и выдает ему результат. Чарльз Бэббидж выдвинул идею создания программно-управляемой счетной машины, имеющей арифметическое устройство, устройство управления, ввода и печати. Первая спроектированная Бэббиджем машина – разностная машина – работала на паровом двигателе. Она высчитывала таблицы логарифмов методом постоянной дифференциации и заносила результаты на металлическую пластину. Работая модель, которую он создал в 1822 г., была шестицифровым калькулятором, способным производить вычисления и печатать цифровые таблицы.



Ада Лавлейс (1815–1852), дочь гениального поэта лорда Байрона, писала программы в кодах этой машины, и поэтому по праву может считаться первым в мире программистом. Именно в ее честь разработанный в США в 1960-х гг. язык программирования был назван Ada. В материалах Бэббиджа и комментариях Лавлейс намечены также понятия, как подпрограмма и библиотека подпрограмм, модификация команд и индексный регистр, которые стали употребляться только в 1950-х гг. Сам термин «библиотека» был введен Бэббиджем, а термины «рабочая ячейка» и «цикл» предложила Ада Лавлейс.

*В 1844 г. Ада писала о машине Бэббиджа в своей работе «Примечания переводчика»: «Многие лица, недостаточно знакомые с математикой, считают, что роль машины сводится к получению результатов в цифровой форме, а природа самой обработки данных должна быть арифметической и аналитической. Это заблуждение. Машина может обрабатывать и объединять цифровые величины точно так, как если бы они были буквами или любыми другими*

*символами общего характера, и фактически она может выдать результаты в алгебраической форме».*



*Там же она образно описала значение перфокарт. «Карты только указывают сущность операций, которые должны быть совершены, и адреса переменных, на которые эти действия направлены. Можно сказать достаточно точно, что аналитическая машина ткёт алгебраические удары, как ткацкий станок Жаккарда – цветы и листья».*

В 1890 г. в США была проведена перепись населения. Одним из тех, кому была поручена обработка огромного объема информации, полученной в результате этой переписи, был доктор Герман Холлерит. Он придумал специальный код (код Холлерита), который использовал для нанесения на перфокарты для обработки результатов переписи, им же был разработан проект счетной машины, использующей перфокарты. Такая машина была создана, и на ней была решена эта задача. Холлерит развивал успех – появилось целое семейство таких машин разного функционального назначения: перфораторы, контрольные, табуляторы. Они получили название счетно-перфорационных машин (СПМ). Эти машины широко

использовались в экономике и управлении до 1970-х гг.



В нашей стране на всех крупных предприятиях и в учреждениях имелись машинно-счетные станции (МСС) или машинно-счетные бюро (МСБ), оснащенные СПМ, на которых решались учетные, бухгалтерские, финансовые и другие задачи.

Первая треть XX в. ознаменовалась созданием и внедрением целого ряда вычислительных устройств. Большой вклад в это внес математик Алан Тьюринг, который в 1937 г. опубликовал работу с ясным и глубоким описанием универсальной схемы вычислений. Для этого он использовал некую гипотетическую, или, говоря сегодняшним языком, виртуальную машину с удивительно простой структурой (она так и вошла в историю как «машина Тьюринга»). Правда эта машина так и осталась на бумаге, поскольку из-за своей медлительности уже тогда была экономически невыгодна, но она привлекла к себе внимание многих талантливых ученых и инженеров и ускорила процесс создания первой универсальной вычислительной машины. И сегодня гипотетическая машина Тьюринга и сходная с ней машина Поста используются в теоретической информатике для доказательства

теорем, анализа алгоритмов и т.д. Существуют даже

Компьютерные программы, моделирующие эти машины.





В 1944 г. под руководством Говарда Айкена из компании IBM была создана электронно-вычислительная машина Harvard Mark I на релейных логических элементах. В 1946 г. американцы Джон Эккерт и Джон Мочли создали ЭВМ на электровакуумных приборах (электронных лампах) – ENIAC. Обе эти ЭВМ были ориентированы в первую очередь на выполнение научно-технических расчетов. Однако появление технологии ЭВМ первого поколения связывают с началом серийного производства UNIVAC в 1951 г.

Нужно сказать, что в нашей стране также было много замечательных достижений в этой области. Так, создание первого арифмометра связывают с именем замечательного русского математика П.Л. Чебышева. Что же касается ЭВМ, то в конце 1940-х гг. группа ученых во главе с академиком С.А. Лебедевым создала в Киеве малую электронно-счетную машину (МЭСМ), в 1952 г. – большую электронно-счетную машину (БЭСМ). Во второй половине 1950-х гг. пошла в серию новая советская ламповая ЭВМ «Урал-1». Чуть позже появилась отечественная ЭВМ «Стр



И американские, и наши ЭВМ первого поколения имели десятки тысяч электронных ламп, занимали огромные (до 300 м<sup>2</sup>) площади и обладали быстродействием в несколько сотен операций в секунду, что по тем временам было просто чудом.

В 1948 г. американские физики Джон Бардин, Уолтер Браттейн и Уильям Шокли изобрели полупроводниковый триод – транзистор, что дало толчок к развитию ЭВМ второго поколения, где на смену электронным лампам пришли транзисторы, которые имели значительно большую продолжительность службы, более высокую надежность, потребляли меньше энергии, были намного компактнее и гораздо дешевле в производстве. Созданные на базе полупроводников ЭВМ второго поколения, начало массового производства которых относят к 1960 г., имели быстродействие в десятки тысяч операций в секунду, занимали площадь уже не в сотни, а в десятки квадратных метров и гораздо реже, нежели машины первого поколения, выходили из строя.



Наряду с аппаратной составляющей развивалось и программное обеспечение. Дело в том, что для ЭВМ первого поколения, да и в значительной степени второго, программы писались в машинных кодах. Это означает, что у каждого типа ЭВМ была своя система команд, которую программист использовал при написании программы. Сами программы для удобства писались в большинстве случаев в восьмеричной системе счисления, а в машине, естественно, были представлены в двоичной, поскольку каждая восьмеричная цифра имеет соответствующую «триаду» двоичных – от 000 до 111. По своей архитектуре машины подразделялись на одно-, двух- и трехадресные. Например, отечественная ЭВМ «Урал» – одноадресная, «Минск» – двухадресная, а машины типа М-20, как БЭСМ, – трехадресные. Для одно- и двухадресных каждая команда имела модификации: например, для «Минск-22» – первый операнд по первому адресу, второй – по второму, а результат – в сумматор. Или первый операнд – в сумматоре, второй – по адресу, а результат – в адрес памяти и т. д.



*В этом смысле более логичными выглядели трехадресные системы команд: код команды (иногда со специальными признаками), первый операнд по первому адресу, второй – по второму, результат записывался по третьему адресу. Например, команда сложения могла выглядеть так: 01 2356 4563 7644. На простом языке это означало: взять число из адреса 2356 оперативной памяти, сложить (код команды – первый в записи) его с числом, находящимся по адресу 4563, а полученный результат записать по адресу, или, как говорили, в ячейку 7644. Причем команды сложения были тоже разные: с блокировкой нормализации, блокировкой округления, блокировкой того и другого и без блокировок. Кроме того, имелся специальный регистр адреса, содержимое которого программист мог использовать для модификации команд, например при организации циклов. Такие системы команд требовали от программиста не только хорошего знания всех нюансов, но и почти полного представления о работе аппаратуры. При этом возникал интересный момент: программист, пишущий, например, для БЭСМ-4, был беспомощен перед «Минск-22», поскольку*

*система команд, адресация, да и многое другое там было не таким.*



Это резко снижало эффективность работы. Еще в 1950-х гг. ученые задумывались над тем, как приблизить машинный язык к естественному. Первой важной разработкой в этом направлении было создание символического языка программирования, автором которого была Грейс Хоппер. В этом языке команды заменялись символами, которые мнемонически (Мнемозина – греческая богиня памяти) отражали их суть, адресам можно было давать имена и т.д. Этот язык был ближе человеку, нежели голые коды. Но ведь ЭВМ продолжала работать в кодах, как, впрочем, и сегодня. Потребовалась программа – переводчик с символического языка на язык кодов. Такие программы стали называть трансляторами, а впоследствии – ассемблерами, а символические языки – автокодами или, соответственно, языками ассемблера. Естественно, что как сам символический язык (язык ассемблера), так и транслятор (ассемблер) для каждого типа ЭВМ был свой и совершенно неприменимый на другом типе ЭВМ, т.е. эти языки программирования были машинно-ориентированными.



Это языки низкого уровня, т.е. наиболее приближенные к конкретной системе команд. Все это не давало возможность непосредственно использовать ЭВМ ученым, инженерам, экономистам, и они вынуждены были работать в паре с программистом, чтобы реализовать свои задачи. Такое положение, естественно, долго не могло оставаться неизменным, и уже в 1956 г. появился язык, максимально приближенный к естественному (английскому) языку, который включал в себя многие элементы алгебраических обозначений, – FORTRAN (FORmula TRANslation). В 1960 г. был создан язык Algol-60, который получил большое признание в научном мире, особенно среди математиков, затем появился язык COBOL, предназначенный для решения задач экономики. Языки стали появляться десятками, причем степень их ориентированности на конкретные проблемы (станки с числовым программным управлением, проектирование многослойных печатных плат, управление технологическими процессами и т.д.) возрастала неимоверно.



На рубеже 1960 – 1970-х гг. был создан универсальный язык программирования PL/1 (от англ. Programming Language – язык программирования). Это был мощный язык, получивший широчайшее распространение в мире и у нас в стране.

Вычислительная техника, естественно, тоже продолжала развиваться. Успехи микроэлектроники, создавшей микросхемы, которые включали в себя сотни, а впоследствии и тысячи электронных схем, дали мощный толчок для создания в конце 1960-х гг. ЭВМ третьего поколения.

Машины третьего поколения стали выпускать семействами, которые включали в себя малые, средние и большие ЭВМ, причем все они были программно-совместимыми. Самыми известными семействами стали IBM 360 и IBM 370. Они имели унифицированную периферию, были совместимы между собой программно, снизу-вверх, т.е. программы, которые «шли» на более младших моделях семейства, были понятны и выполнимы на более старших моделях.



В нашей стране и странах Восточной Европы аналогами IBM 360 и IBM 370 стали семейства «Ряд-1» и, соответственно, «Ряд-2». Например, малую машину ЕС-1010 делали в Венгрии, ЕС-1020 – в Минске, ЕС-1030 – в Казани, ЕС-1040 – в Восточной Германии, ЕС-1050 – в Пензе, ЕС-1060 – в Минске. Аналогичные семейства ЭВМ производились и в других странах – в ФРГ, Японии, Великобритании, Франции.

Важной особенностью ЭВМ третьего поколения явилось появление в них операционных систем. Это, конечно, не значит, что подобные попытки не производились ранее: на ЭВМ типа М-20 была так называемая ИС-22 – интерпретирующая система, которая, интерпретируя псевдокод, значительно упрощала вызов стандартных программ на ЭВМ «Минск-32», была программа «Диспетчер», которая, в частности, обеспечивала диалог с пользователем через двухцветную печатающую машинку типа Consul, и т.п. На семействах IBM использовали сначала DOS, а потом OS (Operation System), а также некоторые их разновидности





Именно начиная с ЭВМ третьего поколения, благодаря наличию у них ОС, был реализован принцип, как тогда говорили, мультипрограммности (многозадачности), например в ОС могло одновременно существовать до 52 задач, как пользовательских, так и служебных (внутренних). Именно с появлением этих ЭВМ программист был освобожден от исполнения функции управления ресурсами компьютерной системы, это стало главной заботой ОС. По мере развития микроэлектроники стали появляться ЭВМ четвертого поколения, которые строились на основе больших интегральных схем (БИС). Процесс микроминиатюризации элементной базы ускорился и позволил в конце 1980-х гг. приступить к серийному выпуску персональных компьютеров. Появились многопроцессорные компьютерные системы, режимы разделения времени, средства телеобработки, телекоммуникации, локальные, а впоследствии и глобальные компьютерные сети, мультимедийные возможности и все то, что так хорошо известно нам сегодня.



Может возникнуть вопрос – зачем нужно знать сегодня о возможностях ЭВМ прошлого? С помощью этих ЭВМ были совершены выдающиеся научные открытия, многие из которых были удостоены Нобелевской премии, осуществлен прорыв в космос, раскрыты секреты атомного ядра, созданы лазерные системы и многое другое. И если вы хотите действительно хорошо разбираться в информатике и в том инструментарии, который она использует, в частности в компьютерной технике, вы должны знать ее историю, поскольку без прошлого нет будущего.

