

Эволюция операционных систем

За период своего существования операционные системы прошли сложный путь, насыщенный многими важными событиями. Огромное влияние на развитие операционных систем оказали успехи в совершенствовании элементной базы и вычислительной аппаратуры, поэтому многие этапы развития ОС тесно связаны с появлением новых типов аппаратных платформ, таких как мини-компьютеры или персональные компьютеры. Серьезную эволюцию операционные системы претерпели в связи с новой ролью компьютеров в локальных и глобальных сетях. Важнейшим фактором развития ОС стал Интернет.

Появление первых операционных систем

В середине 40-х были созданы первые ламповые вычислительные устройства. В то время одна и та же группа людей участвовала и в проектировании, и в эксплуатации, и в программировании вычислительной машины. Это была скорее научно-исследовательская работа в области вычислительной техники, а не использование компьютеров в качестве инструмента решения каких-либо практических задач из других прикладных областей. Программирование осуществлялось исключительно на машинном языке. Не было никакого системного программного обеспечения, кроме библиотек математических и служебных подпрограмм, которые программист мог использовать для того, чтобы не писать каждый раз коды, вычисляющие значение какой-либо математической функции или управляющие стандартным устройством ввода-вывода. Операционные системы все еще не появились, все задачи организации вычислительного процесса решались вручную каждым программистом с пульта управления, который представлял собой примитивное устройство ввода-вывода, состоящее из кнопок, переключателей и индикаторов.

С середины 50-х годов начался новый период в развитии вычислительной техники, связанный с появлением новой технической базы – полупроводниковых элементов. Выросло быстродействие процессоров, увеличились объемы оперативной и внешней памяти. Компьютеры стали более надежными, теперь они могли непрерывно работать настолько долго, чтобы на них можно было возложить выполнение действительно практически важных задач.

Наряду с совершенствованием аппаратуры заметный прогресс наблюдался также в области автоматизации программирования и организации вычислительных работ. В эти годы появились первые алгоритмические языки, и таким образом к библиотекам математических и служебных подпрограмм добавился новый тип системного программного обеспечения – трансляторы.

Выполнение каждой программы стало включать большое количество вспомогательных работ: загрузка нужного транслятора (АЛГОЛ, ФОРТРАН, КОБОЛ и т. п.), запуск транслятора и получение результирующей программы в машинных кодах, связывание программы с библиотечными подпрограммами, загрузка программы в оперативную память, запуск программы, вывод результатов на периферийное устройство. Для организации эффективного совместного использования трансляторов, библиотечных программ и загрузчиков в штат многих вычислительных центров были введены должности

операторов, профессионально выполнявших работу по организации вычислительного процесса для всех пользователей этого центра.

Но как бы быстро и надежно ни работали операторы, они никак не могли состязаться в производительности с работой устройств компьютера. Большую часть времени процессор простаивал в ожидании, пока оператор запустит очередную задачу. А поскольку процессор представлял собой весьма дорогое устройство, то низкая эффективность его использования означала низкую эффективность использования компьютера в целом. Для решения этой проблемы были разработаны первые системы пакетной обработки, которые автоматизировали всю последовательность действий оператора по организации вычислительного процесса.

Ранние системы пакетной обработки явились прообразом современных операционных систем, они стали первыми системными программами, предназначенными не для обработки данных, а для управления вычислительным процессом. В ходе реализации систем пакетной обработки был разработан формализованный язык управления заданиями, с помощью которого программист сообщал системе и оператору, какие действия и в какой последовательности он хочет выполнить на вычислительной машине. Типовой набор директив обычно включал признак начала отдельной работы, вызов транслятора, вызов загрузчика, признаки начала и конца исходных данных.

Оператор составлял пакет заданий, которые в дальнейшем без его участия последовательно запускались на выполнение управляющей программой – монитором. Кроме того, монитор был способен самостоятельно обрабатывать наиболее часто встречающиеся при работе пользовательских программ аварийные ситуации, такие как отсутствие исходных данных, переполнение регистров, деление на ноль, обращение к несуществующей области памяти и т. д. Пакет обычно представлял собой набор перфокарт, но для ускорения работы он мог переноситься на более удобный и емкий носитель, например на магнитную ленту или магнитный диск. Сама программа-монитор в первых реализациях также хранилась на перфокартах или перфоленте, а в более поздних – на магнитной ленте и магнитных дисках.

Ранние системы пакетной обработки значительно сократили затраты времени на вспомогательные действия по организации вычислительного процесса, а значит, был сделан еще один шаг по повышению эффективности использования компьютеров. Однако при этом программисты-пользователи лишились непосредственного доступа к компьютеру, что снижало эффективность их работы – внесение любого исправления требовало значительно больше времени, чем при интерактивной работе за пультом машины.

Появление мультипрограммных операционных систем для мэйнфреймов

Следующий важный период развития операционных систем относится к 1965–1975 годам.

В это время в технической базе вычислительных машин произошел переход от отдельных полупроводниковых элементов типа транзисторов к интегральным микросхемам, что открыло путь к появлению следующего поколения компьютеров. Большие функциональные возможности интегральных схем сделали возможным реализацию на практике сложных компьютерных архитектур, таких, например, как IBM/360.

В этот период были реализованы практически все основные механизмы, присущие современным ОС: мультипрограммирование, мультипроцессирование, поддержка многотерминального многопользовательского режима, виртуальная память, файловые системы, разграничение доступа и сетевая работа. В эти годы начинается расцвет системного программирования. Из направления прикладной математики, представляющего интерес для узкого круга специалистов, системное программирование превращается в отрасль индустрии, оказывающую непосредственное влияние на практическую деятельность миллионов людей.

Революционным событием данного этапа явилась промышленная реализация мультипрограммирования. В условиях резко возросших возможностей компьютера по обработке и хранению данных выполнение только одной программы в каждый момент времени оказалось крайне неэффективным. Решением стало *мультипрограммирование* – способ организации вычислительного процесса, при котором в памяти компьютера находилось одновременно несколько программ, попеременно выполняющихся на одном процессоре. Эти усовершенствования значительно улучшили эффективность вычислительной системы: компьютер теперь мог использоваться почти постоянно, а не менее половины времени работы компьютера, как это было раньше.

Мультипрограммирование было реализовано в двух вариантах – в системах пакетной обработки и разделения времени.

Мультипрограммные системы пакетной обработки так же, как и их однопрограммные предшественники, имели своей целью обеспечение максимальной загрузки аппаратуры компьютера, однако решали эту задачу более эффективно. В мультипрограммном пакетном режиме процессор не простаивал, пока одна программа выполняла операцию ввода-вывода (как это происходило при последовательном выполнении программ в системах ранней пакетной обработки), а переключался на другую готовую к выполнению программу. В результате достигалась сбалансированная загрузка всех устройств компьютера, а, следовательно, увеличивалось число задач, решаемых в единицу времени.

В мультипрограммных системах пакетной обработки пользователь по-прежнему был лишен возможности интерактивно взаимодействовать со своими программами. Для того чтобы хотя бы частично вернуть пользователям ощущение непосредственного взаимодействия с компьютером, был разработан другой вариант мультипрограммных систем – системы разделения времени. Этот вариант рассчитан на многотерминальные системы, когда каждый пользователь работает за своим терминалом. Вариант мультипрограммирования, применяемый в системах разделения времени, был нацелен на создание для каждого отдельного пользователя иллюзии единоличного владения вычислительной машиной за счет периодического выделения каждой программе своей доли процессорного времени. В системах разделения времени эффективность использования оборудования ниже, чем в системах пакетной обработки, что явилось платой за удобства работы пользователя.

Многотерминальный режим использовался не только в системах разделения времени, но и в системах пакетной обработки. При этом не только оператор, но и все пользователи получали возможность формировать свои задания и управлять их выполнением со своего терминала. Такие операционные системы получили название систем удаленного ввода заданий. Терминальные комплексы могли располагаться на большом расстоянии от процессорных стоек, соединяясь с ними с помощью различных

глобальных связей – модемных соединений телефонных сетей или выделенных каналов. Для поддержания удаленной работы терминалов в операционных системах появились специальные программные модули, реализующие различные (в то время, как правило, нестандартные) протоколы связи. Такие вычислительные системы с удаленными терминалами, сохраняя централизованный характер обработки данных, в какой-то степени являлись прообразом современных сетей, а соответствующее системное программное обеспечение – прообразом сетевых операционных систем.

Реализация мультипрограммирования потребовала внесения очень важных изменений в аппаратуру компьютера, непосредственно направленных на поддержку нового способа организации вычислительного процесса. При разделении ресурсов компьютера между программами необходимо обеспечить быстрое переключение процессора с одной программы на другую, а также надежно защитить коды и данные одной программы от непреднамеренной или преднамеренной порчи другой программой. В процессорах появился привилегированный и пользовательский режимы работы, специальные регистры для быстрого переключения с одной программы на другую, средства защиты областей памяти, а также развитая система прерываний.

В привилегированном режиме, предназначенном для работы программных модулей операционной системы, процессор мог выполнять все команды, в том числе и те из них, которые позволяли осуществлять распределение и защиту ресурсов компьютера. Программам, работающим в пользовательском режиме, некоторые команды процессора были недоступны. Таким образом, только ОС могла управлять аппаратными средствами и исполнять роль монитора и арбитра для пользовательских программ, которые выполнялись в непривилегированном, пользовательском режиме.

Система прерываний позволяла синхронизировать работу различных устройств компьютера, работающих параллельно и асинхронно, таких как каналы ввода-вывода, диски, принтеры и т. п. Аппаратная поддержка операционных систем стала с тех пор неотъемлемым свойством практически любых компьютерных систем, включая персональные компьютеры.

Еще одной важной тенденцией этого периода является создание семейств программно-совместимых машин и операционных систем для них. Примерами семейств программно-совместимых машин, построенных на интегральных микросхемах, являются серии машин IBM/360 и IBM/370 (аналоги этих семейств советского производства – машины серии ЕС), PDP-11 (советские аналоги – СМ-3, СМ-4, СМ-1420). Вскоре идея программно-совместимых машин стала общепризнанной.

Программная совместимость требовала и совместимости операционных систем. Однако такая совместимость подразумевает возможность работы на больших и на малых вычислительных системах, с большим и с малым количеством разнообразной периферии, в коммерческой области и в области научных исследований. Операционные системы, построенные с намерением удовлетворить всем этим противоречивым требованиям, оказались чрезвычайно сложными. Они состояли из многих миллионов ассемблерных строк, написанных тысячами программистов, и содержали тысячи ошибок, вызывающих нескончаемый поток исправлений.

Однако, несмотря на необозримые размеры и множество проблем, операционные системы этого поколения действительно удовлетворяли большинству требований потребителей. За это десятилетие был сделан огромный шаг вперед и заложен прочный фундамент для создания современных операционных систем.

Операционные системы и сети передачи данных

В начале 70-х годов появились первые сетевые операционные системы, которые в отличие от многотерминальных ОС позволяли не только рассредоточить пользователей, но и организовать распределенное хранение и обработку данных между несколькими компьютерами, связанными электрическими связями. Любая сетевая операционная система, с одной стороны, выполняет все функции локальной операционной системы, а с другой стороны, обладает некоторыми дополнительными средствами, позволяющими ей взаимодействовать по сети с операционными системами других компьютеров. Программные модули, реализующие сетевые функции, появлялись в операционных системах постепенно, по мере развития сетевых технологий, аппаратной базы компьютеров и возникновения новых задач, требующих сетевой обработки.

Хотя теоретические работы по созданию концепций сетевого взаимодействия велись почти с самого появления вычислительных машин, значимые практические результаты по объединению компьютеров в сети были получены в конце 60-х, когда с помощью глобальных связей и техники коммутации пакетов удалось реализовать взаимодействие машин класса мэйнфреймов и суперкомпьютеров. Эти дорогостоящие компьютеры часто хранили уникальные данные и программы, доступ к которым необходимо было обеспечить широкому кругу пользователей, находившихся в различных городах на значительном расстоянии от вычислительных центров.

К середине 70-х годов наряду с мэйнфреймами широкое распространение получили мини-компьютеры, такие как PDP-11, Nova, HP. Мини-компьютеры первыми использовали преимущества больших интегральных схем, позволившие реализовать достаточно мощные функции при сравнительно невысокой стоимости компьютера.

Архитектура мини-компьютеров была значительно упрощена по сравнению с мэйнфреймами, что нашло отражение и в их операционных системах. Многие функции мультипрограммных многопользовательских ОС мэйнфреймов были усечены, учитывая ограниченность ресурсов мини-компьютеров. Операционные системы мини-компьютеров часто стали делать специализированными, например только для управления в реальном времени (ОС RT-11 для мини-компьютеров PDP-11) или только для поддержания режима разделения времени (RSX-11M для тех же компьютеров). Эти операционные системы не всегда были многопользовательскими, что во многих случаях оправдывалось невысокой стоимостью компьютеров.

Важной вехой в истории мини-компьютеров и вообще в истории операционных систем явилось создание ОС UNIX. С середины 70-х годов началось её массовое использование. К этому времени программный код для UNIX был на 90 % написан на языке высокого уровня C. Широкое распространение эффективных C-компиляторов сделало UNIX уникальной для того времени ОС, обладающей возможностью сравнительно легкого переноса на различные типы компьютеров. Поскольку эта ОС поставлялась вместе с исходными кодами, то она стала первой открытой ОС, которую могли совершенствовать простые пользователи-энтузиасты. Хотя UNIX была первоначально разработана для мини-компьютеров, гибкость, элегантность, мощные функциональные возможности и открытость позволили ей занять прочные позиции во всех классах компьютеров: суперкомпьютерах, мэйнфреймах, мини-компьютерах, серверах и рабочих станциях на базе RISC-процессоров, персональных компьютерах.

Доступность мини-компьютеров и вследствие этого их распространенность на предприятиях послужили мощным стимулом для создания локальных сетей. Предприятие могло себе позволить иметь несколько мини-компьютеров, находящихся в одном здании или даже в одной комнате. Естественно, возникала потребность в обмене информацией между ними и в совместном использовании дорогого периферийного оборудования.

Развитие операционных систем в 80-е годы

К наиболее важным событиям этого десятилетия можно отнести разработку стека TCP/IP, становление Интернета, стандартизацию технологий локальных сетей, появление персональных компьютеров и операционных систем для них.

Все десятилетие было отмечено постоянным появлением новых, все более совершенных версий ОС UNIX. Среди них были и фирменные версии UNIX: SunOS, HP-UX, Irix, AIX и многие другие, в которых производители компьютеров адаптировали код ядра и системных утилит для своей аппаратуры. Разнообразие версий породило проблему их совместимости, которую периодически пытались решить различные организации. В результате были приняты стандарты POSIX и XPG, определяющие интерфейсы ОС для приложений, а специальное подразделение компании AT&T выпустило несколько версий UNIX System III и UNIX System V, призванных консолидировать разработчиков на уровне кода ядра.

Начало 80-х годов связано с еще одним знаменательным для истории операционных систем событием – появлением персональных компьютеров. С точки зрения архитектуры персональные компьютеры ничем не отличались от класса мини-компьютеров типа PDP-11, но их стоимость была существенно ниже. Если мини-компьютер позволил иметь собственную вычислительную машину отделу предприятия или университету, то персональный компьютер дал такую возможность отдельному человеку. Компьютеры стали широко использоваться неспециалистами, что потребовало разработки «дружественного» программного обеспечения, и предоставление этих «дружественных» функций стало прямой обязанностью операционных систем. Персональные компьютеры послужили также мощным катализатором для бурного роста локальных сетей, создав для этого отличную материальную основу в виде десятков и сотен компьютеров, принадлежащих одному предприятию и расположенных в пределах одного здания. В результате поддержка сетевых функций стала для ОС персональных компьютеров необходимым условием.

Однако и дружественный интерфейс, и сетевые функции появились у операционных систем персональных компьютеров не сразу. Первая версия наиболее популярной операционной системы раннего этапа развития персональных компьютеров – MS-DOS компании Microsoft – была лишена этих возможностей. Это была однопрограммная однопользовательская ОС с интерфейсом командной строки, способная стартовать с дискеты. Основными задачами для нее были управление файлами, расположенными на гибких и жестких дисках в UNIX-подобной иерархической файловой системе, а также поочередный запуск программ. MS-DOS не была защищена от программ пользователя, так как процессор Intel 8088 не поддерживал привилегированного режима. Разработчики первых персональных компьютеров считали, что при индивидуальном использовании компьютера и ограниченных возможностях аппаратуры нет смысла в поддержке

мультипрограммирования, поэтому в процессоре не были предусмотрены привилегированный режим и другие механизмы поддержки мультипрограммных систем.

Недостающие функции для MS-DOS и подобных ей ОС компенсировались внешними программами, предоставлявшими пользователю удобный графический интерфейс (например, Norton Commander) или средства тонкого управления дисками (например, PC Tools). Наибольшее влияние на развитие программного обеспечения для персональных компьютеров оказала операционная среда Windows компании Microsoft, представлявшая собой надстройку над MS-DOS.

Сетевые функции также реализовывались в основном сетевыми оболочками, работавшими поверх ОС. При сетевой работе всегда необходимо поддерживать многопользовательский режим, при котором один пользователь – интерактивный, а остальные получают доступ к ресурсам компьютера по сети. В таком случае от операционной системы требуется хотя бы некоторый минимум функциональной поддержки многопользовательского режима. История сетевых средств MS-DOS началась с версии 3.1. Эта версия MS-DOS добавила к файловой системе необходимые средства блокировки файлов и записей, которые позволили более чем одному пользователю иметь доступ к файлу. Пользуясь этими функциями, сетевые оболочки могли обеспечить разделение файлов между сетевыми пользователями.

В 1987 году в результате совместных усилий Microsoft и IBM появилась первая многозадачная операционная система для персональных компьютеров с процессором Intel 80286, в полной мере использующая возможности защищенного режима – OS/2. Эта система была хорошо продуманна. Она поддерживала вытесняющую многозадачность, виртуальную память, графический пользовательский интерфейс (не с первой версии) и виртуальную машину для выполнения DOS-приложений. Фактически она выходила за пределы простой многозадачности с ее концепцией распараллеливания отдельных процессов, получившей название многопоточности.

Сетевые разработки компаний Microsoft и IBM привели к появлению NetBIOS – очень популярного транспортного протокола и одновременно интерфейса прикладного программирования для локальных сетей, получившего применение практически во всех сетевых операционных системах для персональных компьютеров. Этот протокол и сегодня применяется для создания небольших локальных сетей.

В 80-е годы были приняты основные стандарты на коммуникационные технологии для локальных сетей: в 1980 году – Ethernet, в 1985 – Token Ring, в конце 80-х – FDDI. Это позволило обеспечить совместимость сетевых операционных систем на нижних уровнях, а также стандартизировать интерфейс ОС с драйверами сетевых адаптеров.

Для персональных компьютеров применялись не только специально разработанные для них операционные системы, подобные MS-DOS и OS/2, но и адаптировались уже существующие ОС. Появление процессоров Intel 80286 и особенно 80386 с поддержкой мультипрограммирования позволило перенести на платформу персональных компьютеров ОС UNIX.

Особенности современного этапа развития операционных систем

В 90-е годы практически все операционные системы, занимающие заметное место на рынке, стали сетевыми. Сетевые функции сегодня встраиваются в ядро ОС, являясь ее неотъемлемой частью. Операционные системы получили средства для работы со всеми

основными технологиями локальных и глобальных сетей, а также средства для создания составных сетей. В операционных системах используются средства мультиплексирования нескольких стеков протоколов, за счет которого компьютеры могут поддерживать одновременную сетевую работу с разнородными клиентами и серверами. Появились специализированные ОС, которые предназначены исключительно для выполнения коммуникационных задач.

Особое внимание в течение последнего времени уделяется корпоративным сетевым операционным системам. Их дальнейшее развитие представляет одну из наиболее важных задач и в обозримом будущем. Корпоративная операционная система отличается способностью хорошо и устойчиво работать в крупных сетях, которые характерны для больших предприятий, имеющих отделения в десятках городов и, возможно, в разных странах. Таким сетям органически присуща высокая степень гетерогенности программных и аппаратных средств, поэтому корпоративная ОС должна беспрепятственно взаимодействовать с операционными системами разных типов и работать на различных аппаратных платформах.

Для корпоративной ОС очень важно наличие средств централизованного администрирования и управления, позволяющих в единой базе данных хранить учетные записи о десятках тысяч пользователей, компьютеров, коммуникационных устройств и модулей программного обеспечения, имеющихся в корпоративной сети. В современных операционных системах средства централизованного администрирования обычно базируются на единой справочной службе. Роль централизованной справочной службы настолько велика, что именно по качеству справочной службы оценивают пригодность операционной системы для работы в корпоративном масштабе. Длительная задержка выпуска Windows NT 2000 во многом была связана с созданием для этой ОС масштабируемой справочной службы Active Directory, без которой этому семейству ОС трудно было претендовать на звание истинно корпоративной ОС.

Создание многофункциональной масштабируемой справочной службы является стратегическим направлением эволюции ОС. От успехов этого направления во многом зависит и дальнейшее развитие Интернета. Такая служба нужна для превращения Интернета в предсказуемую и управляемую систему, например для обеспечения требуемого качества обслуживания трафика пользователей, поддержки крупных распределенных приложений, построения эффективной почтовой системы и т. п.

На современном этапе развития операционных систем на передний план вышли средства обеспечения безопасности. Это связано с возросшей ценностью информации, обрабатываемой компьютерами, а также с повышенным уровнем угроз, существующих при передаче данных по сетям, особенно по публичным, таким как Интернет. Многие операционные системы обладают сегодня развитыми средствами защиты информации, основанными на шифровании данных, аутентификации и авторизации.

Современным операционным системам присуща многоплатформенность, то есть способность работать на совершенно различных типах компьютеров. Многие операционные системы имеют специальные версии для поддержки кластерных архитектур, обеспечивающих высокую производительность и отказоустойчивость.

В последние годы получила дальнейшее развитие долговременная тенденция повышения удобства работы человека с компьютером. Эффективность работы человека становится основным фактором, определяющим эффективность вычислительной системы

в целом. Усилия человека не должны тратиться на настройку параметров вычислительного процесса, как это происходило в ОС предыдущих поколений.

Современная операционная система берет на себя выполнение задачи выбора параметров операционной среды, используя для этой цели различные адаптивные алгоритмы. Например, тайм-ауты в коммуникационных протоколах часто определяются в зависимости от условий работы сети. Распределение оперативной памяти между процессами осуществляется автоматически с помощью механизмов виртуальной памяти в зависимости от активности этих процессов и информации о частоте использования ими той или иной страницы. Мгновенные приоритеты процессов определяются динамически в зависимости от предыстории, включающей, например, время нахождения процесса в очереди, процент использования выделенного кванта времени, интенсивность ввода-вывода и т. п. Даже в процессе установки большинство ОС предлагают режим выбора параметров по умолчанию, который гарантирует пусть не оптимальное, но всегда приемлемое качество работы систем.

Постоянно повышается удобство интерактивной работы с компьютером путем включения в операционную систему развитых графических интерфейсов, использующих наряду с графикой звук и видеоизображение. Это особенно важно для превращения компьютера в терминал новой публичной сети, которой постепенно становится Интернет, так как для массового пользователя терминал должен быть почти таким же понятным и удобным, как телефонный аппарат. Пользовательский интерфейс операционной системы становится все более интеллектуальным, направляя действия человека в типовых ситуациях и принимая за него рутинные решения.

Назначение и функции операционной системы

Сегодня существует большое количество разных типов операционных систем, отличающихся областями применения, аппаратными платформами и методами реализации. Естественно, это обуславливает и значительные функциональные различия этих ОС. Даже у конкретной операционной системы набор выполняемых функций зачастую определить не так просто – та функция, которая сегодня выполняется внешним по отношению к ОС компонентом, завтра может стать ее неотъемлемой частью и наоборот. Поэтому при изучении операционных систем очень важно из всего многообразия выделить те функции, которые присущи всем операционным системам как классу продуктов.

Операционные системы для автономного компьютера

Операционная система компьютера представляет собой комплекс взаимосвязанных программ, который действует как интерфейс между приложениями и пользователями с одной стороны, и аппаратурой компьютера с другой стороны.

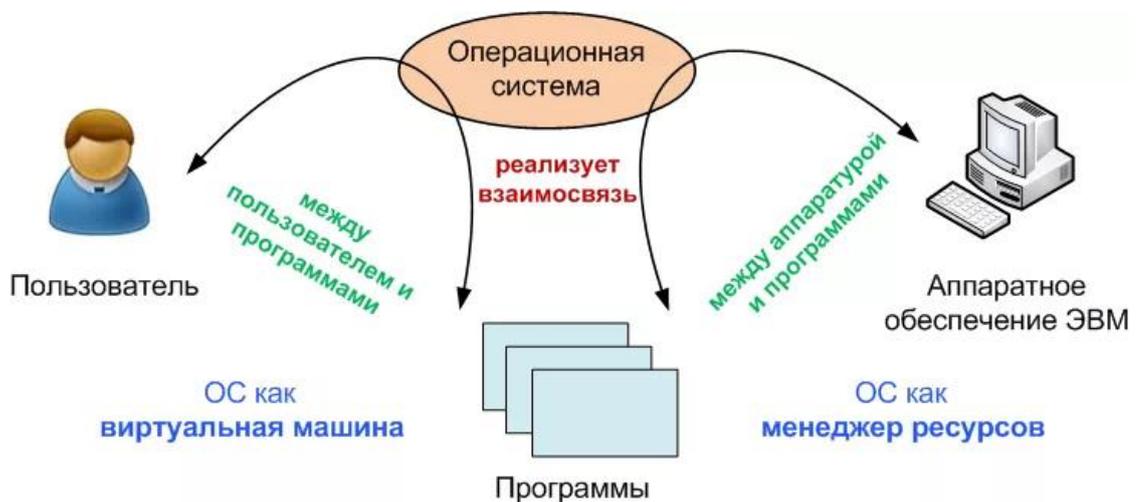


Рисунок 1 – Структура взаимодействия ОС с различными компонентами

В соответствии с этим определением ОС выполняет две группы функций:

- предоставление пользователю или программисту вместо реальной аппаратуры компьютера расширенной виртуальной машины, с которой удобней работать и которую легче программировать;
- повышение эффективности использования компьютера путем рационального управления его ресурсами в соответствии с некоторым критерием.

Операционная система – наиболее фундаментальная часть программного обеспечения, работающая в режиме ядра (этот режим называют еще режимом супервизора). В этом режиме она имеет полный доступ ко всему аппаратному обеспечению и может задействовать любую инструкцию, которую машина в состоянии выполнить.

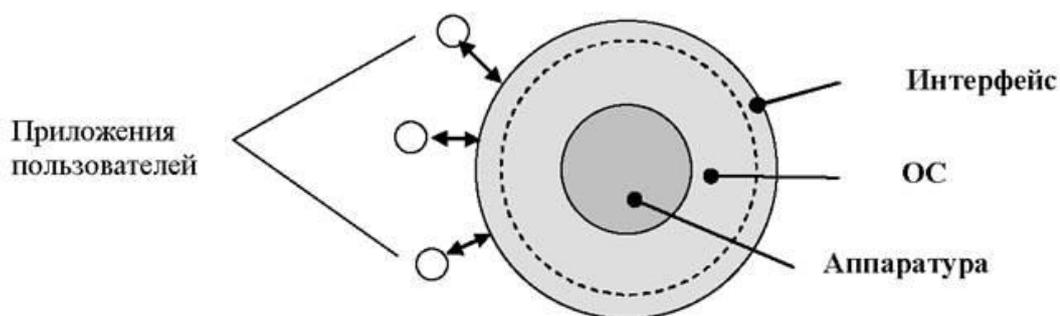


Рисунок 2 – Общая структура вычислительного устройства

Операционная система изолирует аппаратное обеспечение компьютера от прикладных программ пользователей (рисунок 2). И пользователь, и его программы взаимодействуют с компьютером через интерфейсы операционной системы. ОС работает непосредственно с аппаратным обеспечением и является основой всего остального программного обеспечения.

ОС как виртуальная машина

Для того чтобы успешно решать свои задачи, современный пользователь или даже прикладной программист может обойтись без досконального знания аппаратного устройства компьютера. Ему не обязательно быть в курсе того, как функционируют различные электронные блоки и электромеханические узлы компьютера. Более того, очень часто пользователь может не знать даже системы команд процессора. Пользователь-программист привык иметь дело с мощными высокоуровневыми функциями, которые ему предоставляет операционная система.

Так, например, при работе с диском программисту, пишущему приложение для работы под управлением ОС, или конечному пользователю ОС достаточно представлять его в виде некоторого набора файлов, каждый из которых имеет имя. Последовательность действий при работе с файлом заключается в его открытии, выполнении одной или нескольких операций чтения или записи, а затем в закрытии файла. Такие частности, как используемая при записи частотная модуляция или текущее состояние двигателя механизма перемещения магнитных головок чтения/записи, не должны волновать программиста. Именно операционная система скрывает от программиста большую часть особенностей аппаратуры и предоставляет возможность простой и удобной работы с требуемыми файлами.

Операционная система избавляет программистов не только от необходимости напрямую работать с аппаратурой дискового накопителя, предоставляя им простой файловый интерфейс, но и берет на себя все другие рутинные операции, связанные с управлением другими аппаратными устройствами компьютера: физической памятью, таймерами, принтерами и т. д.

В результате реальная машина, способная выполнять только небольшой набор элементарных действий, определяемых ее системой команд, превращается в виртуальную машину, выполняющую широкий набор гораздо более мощных функций. Виртуальная машина тоже управляется командами, но это уже команды другого, более высокого уровня: удалить файл с определенным именем, запустить на выполнение некоторую прикладную программу, повысить приоритет задачи, вывести текст из файла на печать. Таким образом, назначение ОС состоит в предоставлении пользователю/программисту некоторой

расширенной виртуальной машины, которую легче программировать и с которой легче работать, чем непосредственно с аппаратурой, составляющей реальный компьютер или реальную сеть.

ОС как система управления ресурсами

Операционная система не только предоставляет пользователям и программистам удобный интерфейс к аппаратным средствам компьютера, но и является механизмом, распределяющим ресурсы компьютера.

К числу основных ресурсов современных вычислительных систем могут быть отнесены такие ресурсы, как процессоры, основная память, таймеры, наборы данных, диски, накопители на магнитных лентах, принтеры, сетевые устройства и некоторые другие. Ресурсы распределяются между процессами. Процесс (задача) представляет собой базовое понятие большинства современных ОС и часто кратко определяется как программа в стадии выполнения. Программа – это статический объект, представляющий собой файл с кодами и данными. Процесс – это динамический объект, который возникает в операционной системе после того, как пользователь или сама операционная система решает «запустить программу на выполнение», то есть создать новую единицу вычислительной работы.

Управление ресурсами вычислительной системы с целью наиболее эффективного их использования является назначением операционной системы. Например, мультипрограммная операционная система организует одновременное выполнение сразу нескольких процессов на одном компьютере, поочередно переключая процессор с одного процесса на другой, исключая простои процессора, вызываемые обращениями процессов к вводу-выводу. ОС также отслеживает и разрешает конфликты, возникающие при обращении нескольких процессов к одному и тому же устройству ввода-вывода или к одним и тем же данным.

Критерий эффективности, в соответствии с которым ОС организует управление ресурсами компьютера, может быть различным. Например, в одних системах важен такой критерий, как пропускная способность вычислительной системы, в других – время ее реакции. Соответственно выбранному критерию эффективности операционные системы по-разному организуют вычислительный процесс.

Управление ресурсами включает решение следующих общих, не зависящих от типа ресурса задач:

- планирование ресурса – то есть определение, какому процессу, когда и в каком количестве (если ресурс может выделяться частями) следует выделить данный ресурс;
- удовлетворение запросов на ресурсы;
- отслеживание состояния и учет использования ресурса – то есть поддержание оперативной информации о том, занят или свободен ресурс и какая доля ресурса уже распределена;
- разрешение конфликтов между процессами.

Для решения этих общих задач управления ресурсами разные ОС используют различные алгоритмы, особенности которых в конечном счете и определяют облик ОС в целом, включая характеристики производительности, область применения и даже пользовательский интерфейс. Например, применяемый алгоритм управления процессором

в значительной степени определяет, может ли ОС использоваться как система разделения времени, система пакетной обработки или система реального времени.

Задача организации эффективного совместного использования ресурсов несколькими процессами является весьма сложной, и сложность эта порождается в основном случайным характером возникновения запросов на потребление ресурсов. В мультипрограммной системе образуются очереди заявок от одновременно выполняемых программ к разделяемым ресурсам компьютера: процессору, странице памяти, к принтеру, к диску. Операционная система организует обслуживание этих очередей по разным алгоритмам: в порядке поступления, на основе приоритетов, кругового обслуживания и т. д.

Таким образом, управление ресурсами составляет важную часть функций любой операционной системы, в особенности мультипрограммной. В отличие от функций расширенной машины большинство функций управления ресурсами выполняются операционной системой автоматически и прикладному программисту недоступны.

Классификация операционных систем

Операционные системы могут различаться:

1. особенностями реализации внутренних алгоритмов управления основными ресурсами компьютера (процессорами, памятью, устройствами);
2. типами аппаратных платформ;
3. областями использования;
4. особенностями использованных методов проектирования ОС и т.д.

Особенности реализации внутренних алгоритмов управления основными ресурсами компьютера

От эффективности алгоритмов управления локальными ресурсами компьютера во многом зависит эффективность всей операционной системы в целом. Поэтому, характеризуя операционную систему, часто приводят важнейшие особенности реализации функций операционной системы по управлению процессорами, памятью, внешними устройствами компьютера.

Так, например, в зависимости от особенностей использованного алгоритма управления процессором, операционные системы делят:

1. На два класса по числу одновременно выполняемых задач:
 - **Однозадачные операционные системы** (например, MS-DOS, FreeDOS) в основном выполняют функцию предоставления пользователю виртуальной машины, делая более простым и удобным процесс взаимодействия пользователя с компьютером. Однозадачные ОС включают средства управления периферийными устройствами, средства управления файлами, средства общения с пользователем.
 - **Многозадачные операционные системы** (UNIX, Windows), кроме вышеперечисленных функций, управляют разделением совместно используемых ресурсов, таких как процессор, оперативная память, файлы и внешние устройства.
2. По числу одновременно работающих пользователей ОС на:
 - **однопользовательские** (MS-DOS, Windows 3.x, ранние версии OS/2);
 - **многопользовательские** (UNIX, Windows).

Главным отличием многопользовательских систем от однопользовательских является наличие средств защиты информации каждого пользователя от несанкционированного доступа других пользователей.

Следует заметить, что не всякая многозадачная система является многопользовательской, и не всякая однопользовательская ОС является однозадачной.

3. Важнейшим разделяемым ресурсом является процессорное время. Способ распределения процессорного времени между несколькими одновременно существующими в системе процессами (или нитями) во многом определяет специфику ОС. Среди множества существующих вариантов реализации многозадачности можно выделить две группы алгоритмов:

- **не вытесняющая многозадачность** (NetWare, Windows3.x). При **не вытесняющей** многозадачности активный процесс выполняется до тех пор, пока он сам, по собственной инициативе, не отдаст управление операционной системе для того, чтобы та выбрала из очереди другой готовый к выполнению процесс.
- **вытесняющая многозадачность** (Windows NT, Unix). При **вытесняющей** многозадачности решение о переключении процессора с одного процесса на другой принимается операционной системой, а не самим активным процессом.

Основным различием вытесняющего и не вытесняющего вариантов многозадачности является степень централизации механизма планирования процессов. В первом механизм планирования процессов целиком сосредоточен в операционной системе, а во втором – распределен между системой и прикладными программами.

4. По наличию поддержки многонитевости. Важным свойством операционных систем является возможность распараллеливания вычислений в рамках одной задачи. Многонитевая ОС разделяет время процессора не между задачами, а между их отдельными ветвями (нитями).

5. По отсутствию или наличию в ОС средств поддержки многопроцессорной обработки – мультипроцессированию. Мультипроцессирование неизбежно приводит к усложнению всех алгоритмов управления ресурсами. Многопроцессорные ОС могут классифицироваться по способу организации вычислительного процесса в системе с многопроцессорной архитектурой:

- **Асимметричная операционная система** целиком выполняется только на одном из процессоров системы, распределяя прикладные задачи по остальным процессорам.
- **Симметричная операционная система** полностью децентрализована и использует весь пул процессоров, разделяя их между системными и прикладными задачами.

Особенности аппаратных платформ

На свойства операционной системы непосредственное влияние оказывают аппаратные средства, на которые она ориентирована. По типу аппаратуры различают ОС:

- персональных компьютеров;
- мини-компьютеров;
- мейнфреймов;

- кластеров и сетей ЭВМ.

Среди перечисленных типов компьютеров могут встречаться как однопроцессорные варианты, так и многопроцессорные. В любом случае специфика аппаратных средств, как правило, отражается на специфике операционных систем.

Очевидно, что ОС большой машины является более сложной и функциональной, чем ОС персонального компьютера. Так в ОС больших машин функции по планированию потока выполняемых задач реализуются путем использования сложных приоритетных дисциплин и требуют большей вычислительной мощности, чем в ОС персональных компьютеров. Аналогично обстоит дело и с другими функциями

Особенности областей использования ОС

Многозадачные ОС подразделяются на три типа в соответствии с использованными при их разработке критериями эффективности:

- системы пакетной обработки (например, ОС ЕС),
- системы разделения времени (Unix, Windows),
- системы реального времени (QNX, FreeRTOS).

Системы пакетной обработки

Системы пакетной обработки предназначались для решения задач в основном вычислительного характера, не требующих быстрого получения результатов.

Главной целью и критерием эффективности систем пакетной обработки является максимальная пропускная способность, то есть решение максимального числа задач в единицу времени. Для достижения этой цели в системах пакетной обработки используются следующая схема функционирования:

- в начале работы формируется пакет заданий, каждое задание содержит требование к системным ресурсам;
- из этого пакета заданий формируется мультипрограммная смесь, то есть множество одновременно выполняемых задач.

Для одновременного выполнения выбираются задачи, предъявляющие отличающиеся требования к ресурсам, так, чтобы обеспечивалась сбалансированная загрузка всех устройств вычислительной машины; так, например, в мультипрограммной смеси желательно одновременное присутствие вычислительных задач и задач с интенсивным вводом-выводом. Таким образом, выбор нового задания из пакета заданий зависит от внутренней ситуации, складывающейся в системе, то есть выбирается «выгодное» задание. Следовательно, в таких ОС невозможно гарантировать выполнение того или иного задания в течение определенного периода времени.

В системах пакетной обработки переключение процессора с выполнения одной задачи на выполнение другой происходит только в случае, если активная задача сама отказывается от процессора, например, из-за необходимости выполнить операцию ввода-вывода. Поэтому одна задача может надолго занять процессор, что делает невозможным выполнение интерактивных задач. Таким образом, взаимодействие пользователя с вычислительной машиной, на которой установлена система пакетной обработки, сводится к тому, что он приносит задание, отдает его диспетчеру оператору, а в

конце дня после выполнения всего пакета заданий получает результат. Очевидно, что такой порядок снижает эффективность работы пользователя.

Системы разделения времени

Системы разделения времени призваны исправить основной недостаток систем пакетной обработки – изоляцию пользователя-программиста от процесса выполнения его задач. Каждому пользователю системы разделения времени предоставляется терминал, с которого он может вести диалог со своей программой.

Так как в системах разделения времени **каждой задаче выделяется только квант процессорного времени**, ни одна задача не занимает процессор надолго, и время ответа оказывается приемлемым. Если выбранный квант достаточно небольшой, то у всех пользователей, одновременно работающих на одной и той же машине, складывается впечатление, что каждый из них единолично использует машину. Ясно, что системы разделения времени обладают меньшей пропускной способностью, чем системы пакетной обработки, так как на выполнение принимается каждая запущенная пользователем задача, а не та, которая «выгодна» системе, и, кроме того, имеются накладные расходы вычислительной мощности на более частое переключение процессора с задачи на задачу.

Критерием эффективности систем разделения времени является не максимальная пропускная способность, а **удобство и эффективность работы пользователя**.

Системы реального времени

Системы реального времени применяются для управления различными техническими объектами (станок, спутник, научная экспериментальная установка) или технологическими процессами (гальваническая линия, доменный процесс и т.п.). Во всех этих случаях **существует предельно допустимое время, за которое должна быть выполнена та или иная программа, управляющая объектом, в противном случае может произойти авария**: спутник выйдет из зоны видимости, экспериментальные данные, поступающие с датчиков, будут потеряны, толщина гальванического покрытия не будет соответствовать норме.

Таким образом, критерием эффективности для систем реального времени является их способность выдерживать заранее заданные интервалы времени между запуском программы и получением результата (управляющего воздействия). Это время называется **временем реакции системы**, а соответствующее свойство системы – **реактивностью**. Для этих систем мультипрограммная смесь представляет собой фиксированный набор заранее разработанных программ, а выбор программы на выполнение осуществляется исходя из текущего состояния объекта или в соответствии с расписанием плановых работ.

Некоторые операционные системы могут совмещать в себе свойства систем разных типов, например, часть задач может выполняться в режиме пакетной обработки, а часть – в режиме реального времени или в режиме разделения времени.

Особенности методов проектирования ОС

В руководстве по работе с операционной системой часто указываются особенности ее структурной организации и основные концепции, положенные в ее основу.

К таким базовым концепциям относится способ построения ядра системы: **монолитное ядро** или **микроядро**.

Большинство ОС использует монолитное ядро, которое компонуется как одна программа, работающая в привилегированном режиме и использующая быстрые переходы с одной процедуры на другую, не требующие переключения из привилегированного режима в пользовательский режим и наоборот.



Рисунок 3 – ОС на основе монолитного ядра

Альтернативой является построение ОС на базе микроядра, работающего также в привилегированном режиме и выполняющего только минимум функций по управлению аппаратурой, в то время как функции ОС более высокого уровня выполняют специализированные компоненты ОС – серверы, работающие в пользовательском режиме. При такой реализации ОС работает более медленно, так как часто выполняются переходы между привилегированным режимом и пользовательским, зато система получается более гибкой – ее функции можно наращивать, модифицировать или сужать, добавляя, модифицируя или исключая серверы пользовательского режима. Кроме того, серверы хорошо защищены друг от друга, как и любые пользовательские процессы.

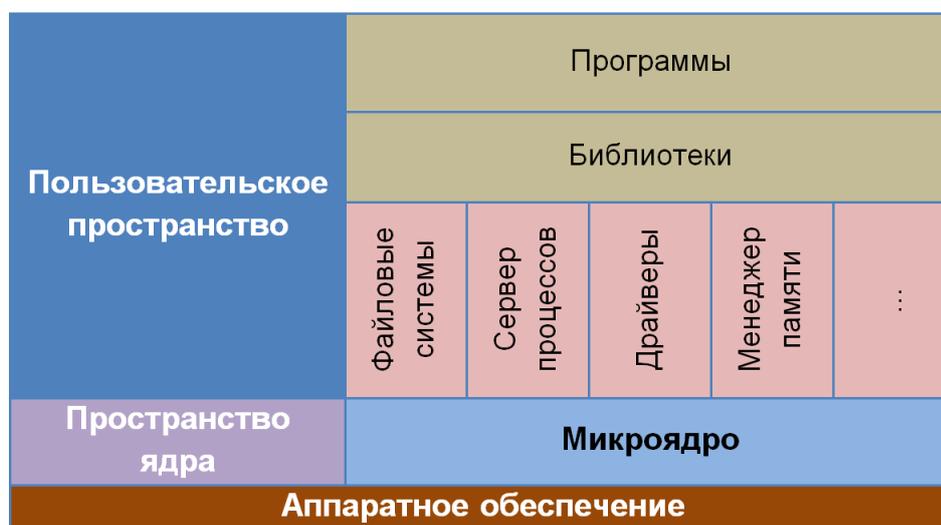


Рисунок 4 – ОС на основе микроядра

Функциональные компоненты операционной системы автономного компьютера

Функции операционной системы автономного компьютера обычно группируются либо в соответствии с типами локальных ресурсов, которыми управляет ОС, либо в соответствии со специфическими задачами, применимыми ко всем ресурсам. Иногда такие группы функций называют подсистемами. Наиболее важными подсистемами управления ресурсами являются подсистемы управления процессами, памятью, файлами и внешними устройствами, а подсистемами, общими для всех ресурсов, являются подсистемы пользовательского интерфейса, защиты данных и администрирования.

Управление процессами

Важнейшей частью операционной системы, непосредственно влияющей на функционирование вычислительной машины, является подсистема управления процессами.

Для каждого вновь создаваемого процесса ОС генерирует системные информационные структуры, которые содержат данные о потребностях процесса в ресурсах вычислительной системы, а также о фактически выделенных ему ресурсах. Таким образом, процесс можно также определить как некоторую заявку на потребление системных ресурсов.

Чтобы процесс мог быть выполнен, операционная система должна назначить ему область оперативной памяти, в которой будут размещены коды и данные процесса, а также предоставить ему необходимое количество процессорного времени. Кроме того, процессу может понадобиться доступ к таким ресурсам, как файлы и устройства ввода-вывода.

В информационные структуры процесса часто включаются вспомогательные данные, характеризующие историю пребывания процесса в системе (например, какую долю времени процесс потратил на операции ввода-вывода, а какую на вычисления), его текущее состояние (активное или заблокированное), степень привилегированности процесса (значение приоритета). Данные такого рода могут учитываться операционной системой при принятии решения о предоставлении ресурсов процессу.

В мультипрограммной операционной системе одновременно может существовать несколько процессов. Часть процессов порождается по инициативе пользователей и их приложений, такие процессы обычно называют пользовательскими. Другие процессы, называемые системными, инициализируются самой операционной системой для выполнения своих функций.

Поскольку процессы часто одновременно претендуют на одни и те же ресурсы, то в обязанности ОС входит поддержание очередей заявок процессов на ресурсы, например очереди к процессору, к принтеру, к последовательному порту.

Важной задачей операционной системы является защита ресурсов, выделенных данному процессу, от остальных процессов. Одним из наиболее тщательно защищаемых ресурсов процесса являются области оперативной памяти, в которой хранятся коды и данные процесса. Совокупность всех областей оперативной памяти, выделенных операционной системой процессу, называется его адресным пространством. Говорят, что каждый процесс работает в своем адресном пространстве, имея в виду защиту адресных

пространств, осуществляемую ОС. Защищаются и другие типы ресурсов, такие как файлы, внешние устройства и т. д. Операционная система может не только защищать ресурсы, выделенные одному процессу, но и организовывать их совместное использование, например разрешать доступ к некоторой области памяти нескольким процессам.

Имя	Состояние	2% ЦП	40% Память	0% Диск	0% Сеть	5% GPU	Ядро GPU
Приложения (6)							
> Firefox (12)		0,4%	8 989,0 МБ	0,1 МБ/с	0,1 Мбит/с	0%	Графический п
> GNU Image Manipulation Program		0%	128,1 МБ	0 МБ/с	0 Мбит/с	0%	
> WinDjView		0%	82,0 МБ	0 МБ/с	0 Мбит/с	0%	
> Блокнот		0%	2,0 МБ	0 МБ/с	0 Мбит/с	0%	
> Диспетчер задач		0,7%	31,6 МБ	0 МБ/с	0 Мбит/с	0%	
> Проводник (3)		0%	119,3 МБ	0 МБ/с	0 Мбит/с	0%	
Фоновые процессы (47)							
> Adobe Acrobat Update Service (32 бита)		0%	1,0 МБ	0 МБ/с	0 Мбит/с	0%	
> Antimalware Service Executable		0%	170,8 МБ	0 МБ/с	0 Мбит/с	0%	
Application Frame Host		0%	6,3 МБ	0 МБ/с	0 Мбит/с	0%	
COM Surrogate		0%	3,5 МБ	0 МБ/с	0 Мбит/с	0%	
COM Surrogate		0%	3,1 МБ	0 МБ/с	0 Мбит/с	0%	
Component Package Support Server		0%	2,3 МБ	0 МБ/с	0 Мбит/с	0%	
> Creative Audio Service (32 бита)		0%	1,2 МБ	0 МБ/с	0 Мбит/с	0%	
> Creative.AudPosService (32 бита)		0%	9,3 МБ	0 МБ/с	0 Мбит/с	0%	
CTF-загрузчик		0%	7,8 МБ	0 МБ/с	0 Мбит/с	0%	
Device Association Framework Provider Host		0%	7,1 МБ	0 МБ/с	0 Мбит/с	0%	

Рисунок 5 – Активные процессы в ОС

На протяжении периода существования процесса его выполнение может быть многократно прервано и продолжено. Для того чтобы возобновить выполнение процесса, необходимо восстановить состояние его операционной среды. Состояние операционной среды идентифицируется состоянием регистров и программного счетчика, режимом работы процессора, указателями на открытые файлы, информацией о незавершенных операциях ввода-вывода, кодами ошибок выполняемых данным процессом системных вызовов и т. д. Эта информация называется контекстом процесса. Говорят, что при смене процесса происходит переключение контекстов.

Операционная система берет на себя также функции синхронизации процессов, позволяющие процессу приостанавливать свое выполнение до наступления какого-либо события в системе, например завершения операции ввода-вывода, осуществляемой по его запросу операционной системой.

В операционной системе нет однозначного соответствия между процессами и программами. Один и тот же программный файл может породить несколько параллельно

выполняемых процессов, а процесс может в ходе своего выполнения сменить программный файл и начать выполнять другую программу.

Для реализации сложных программных комплексов полезно бывает организовать их работу в виде нескольких параллельных процессов, которые периодически взаимодействуют друг с другом и обмениваются некоторыми данными. Так как операционная система защищает ресурсы процессов и не позволяет одному процессу писать или читать из памяти другого процесса, то для оперативного взаимодействия процессов ОС должна предоставлять особые средства, которые называют средствами межпроцессного взаимодействия.

Таким образом, подсистема управления процессами планирует выполнение процессов, то есть распределяет процессорное время между несколькими одновременно существующими в системе процессами, занимается созданием и уничтожением процессов, обеспечивает процессы необходимыми системными ресурсами, поддерживает синхронизацию процессов, а также обеспечивает взаимодействие между процессами.

Управление памятью

Память является для процесса таким же важным ресурсом, как и процессор, так как процесс может выполняться процессором только в том случае, если его коды и данные (не обязательно все) находятся в оперативной памяти.

Управление памятью включает распределение имеющейся физической памяти между всеми существующими в системе в данный момент процессами, загрузку кодов и данных процессов в отведенные им области памяти, настройку адреснозависимых частей кодов процесса на физические адреса выделенной области, а также защиту областей памяти каждого процесса.

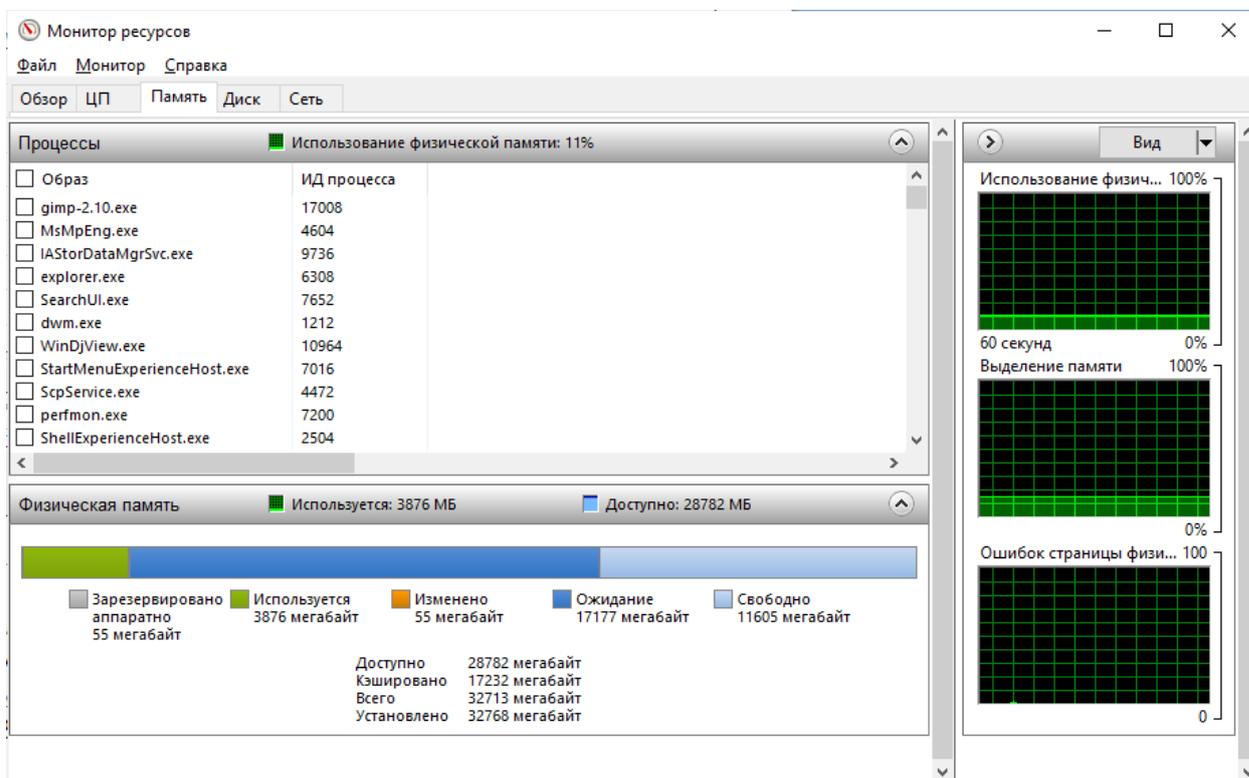


Рисунок 6 – Распределение памяти в ОС

Существует большое разнообразие алгоритмов распределения памяти. Они могут отличаться, например, количеством выделяемых процессу областей памяти (в одних случаях память выделяется процессу в виде одной непрерывной области, а в других – в виде нескольких несмежных областей), степенью свободы границы областей (она может быть жестко зафиксирована на все время существования процесса или же динамически перемещаться при выделении процессу дополнительных объемов памяти). В некоторых системах распределение памяти выполняется страницами фиксированного размера, а в других – сегментами переменной длины.

Одним из наиболее популярных способов управления памятью в современных операционных системах является так называемая виртуальная память. Наличие в ОС механизма виртуальной памяти позволяет программисту писать программу так, как будто в его распоряжении имеется однородная оперативная память большого объема, часто существенно превышающего объем имеющейся физической памяти. В действительности все данные, используемые программой, хранятся на диске и при необходимости частями (сегментами или страницами) отображаются в физическую память. При перемещении кодов и данных между оперативной памятью и диском подсистема виртуальной памяти выполняет трансляцию виртуальных адресов, полученных в результате компиляции и компоновки программы, в физические адреса ячеек оперативной памяти. Очень важно, что все операции по перемещению кодов и данных между оперативной памятью и дисками, а также трансляция адресов выполняются ОС прозрачно для программиста.

Защита памяти – это избирательная способность предохранять выполняемую задачу от записи или чтения памяти, назначенной другой задаче. Правильно написанные программы не пытаются обращаться к памяти, назначенной другим. Однако реальные программы часто содержат ошибки, в результате которых такие попытки иногда предпринимаются. Средства защиты памяти, реализованные в операционной системе, должны пресекать несанкционированный доступ процессов к чужим областям памяти.

Таким образом, функциями ОС по управлению памятью являются отслеживание свободной и занятой памяти; выделение памяти процессам и освобождение памяти при завершении процессов; защита памяти; вытеснение процессов из оперативной памяти на диск, когда размеры основной памяти недостаточны для размещения в ней всех процессов, и возвращение их в оперативную память, когда в ней освобождается место, а также настройка адресов программы на конкретную область физической памяти.

Управление файлами и внешними устройствами

Способность ОС к «экранированию» сложностей реальной аппаратуры очень ярко проявляется в одной из основных подсистем ОС – файловой системе. Операционная система виртуализирует отдельный набор данных, хранящихся на внешнем накопителе, в виде файла – простой неструктурированной последовательности байтов, имеющей символическое имя. Для удобства работы с данными файлы группируются в каталоги, которые, в свою очередь, образуют группы – каталоги более высокого уровня. Пользователь может с помощью ОС выполнять над файлами и каталогами такие действия, как поиск по имени, удаление, вывод содержимого на внешнее устройство (например, на дисплей), изменение и сохранение содержимого.

Чтобы представить большое количество наборов данных, разбросанных случайным образом по цилиндрам и поверхностям дисков различных типов, в виде хорошо всем

знакомой и удобной иерархической структуры файлов и каталогов, операционная система должна решить множество задач. Файловая система ОС выполняет преобразование символьных имен файлов, с которыми работает пользователь или прикладной программист, в физические адреса данных на диске, организует совместный доступ к файлам, защищает их от несанкционированного доступа.

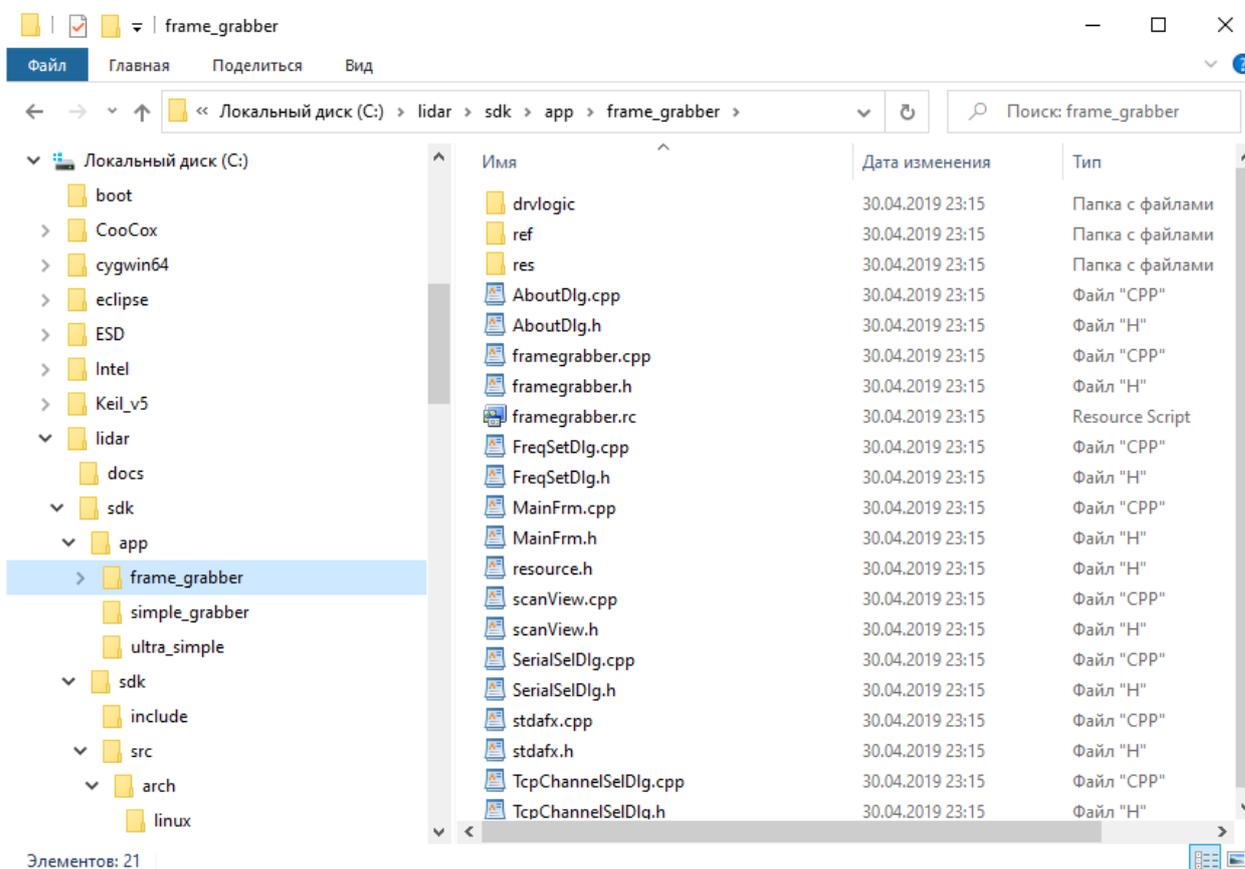


Рисунок 7 – Иерархическое представление данных в файловой системе

При выполнении своих функций файловая система тесно взаимодействует с подсистемой управления внешними устройствами, которая по запросам файловой системы осуществляет передачу данных между дисками и оперативной памятью.

Подсистема управления внешними устройствами, называемая также подсистемой ввода-вывода, исполняет роль интерфейса ко всем устройствам, подключенным к компьютеру. Спектр этих устройств очень обширен. Номенклатура выпускаемых накопителей на жестких, гибких и оптических дисках, принтеров, сканеров, мониторов, плоттеров, модемов, сетевых адаптеров и более специальных устройств ввода-вывода, таких как, например, аналого-цифровые преобразователи, может насчитывать сотни моделей. Эти модели могут существенно отличаться набором и последовательностью команд, с помощью которых осуществляется обмен информацией с процессором и памятью компьютера, скоростью работы, кодировкой передаваемых данных, возможностью совместного использования и множеством других деталей.

Программа, управляющая конкретной моделью внешнего устройства и учитывающая все его особенности, обычно называется **драйвером** этого устройства (от английского drive – управлять, вести). Драйвер может управлять единственной моделью устройства, или же группой устройств определенного типа. Для пользователя очень важно, чтобы операционная система включала как можно больше разнообразных драйверов, так

как это гарантирует возможность подключения к компьютеру большого числа внешних устройств различных производителей. От наличия подходящих драйверов во многом зависит успех операционной системы на рынке.

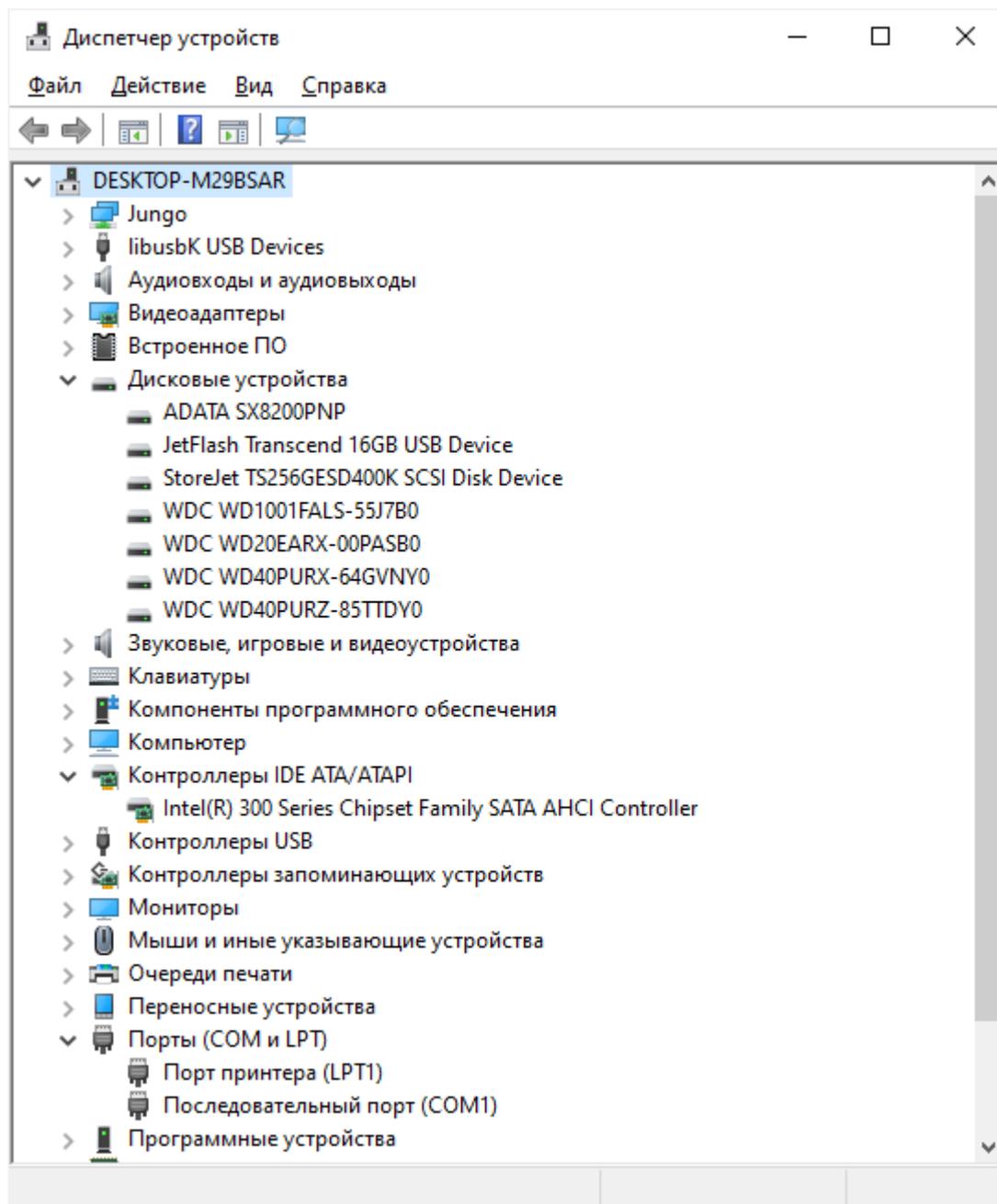


Рисунок 8 – Устройства внутри ПК

Созданием драйверов устройств занимаются как разработчики конкретной ОС, так и специалисты компаний, выпускающих внешние устройства. Операционная система должна поддерживать хорошо определенный интерфейс между драйверами и остальной частью ОС, чтобы разработчики из компаний-производителей устройств ввода-вывода могли поставлять вместе со своими устройствами драйверы для данной операционной системы.

Прикладные программисты могут пользоваться интерфейсом драйверов при разработке своих программ, но это не очень удобно – такой интерфейс обычно представляет собой низкоуровневые операции, обремененные большим количеством деталей.

Поддержание высокоуровневого унифицированного интерфейса прикладного

программирования к разнородным устройствам ввода-вывода является одной из наиболее важных задач ОС. Со времени появления ОС UNIX такой унифицированный интерфейс в большинстве операционных систем строится на основе концепции файлового доступа. Эта концепция заключается в том, что обмен с любым внешним устройством выглядит как обмен с файлом, имеющим имя и представляющим собой неструктурированную последовательность байтов. В качестве файла может выступать как реальный файл на диске, так и алфавитно-цифровой терминал, печатающее устройство или сетевой адаптер. Здесь мы опять имеем дело со свойством операционной системы подменять реальную аппаратуру удобными для пользователя и программиста абстракциями.

Защита данных и администрирование

Безопасность данных вычислительной системы обеспечивается средствами отказоустойчивости ОС, направленными на защиту от сбоев и отказов аппаратуры и ошибок программного обеспечения, а также средствами защиты от несанкционированного доступа. В последнем случае ОС защищает данные от ошибочного или злонамеренного поведения пользователей системы.

Первым рубежом обороны при защите данных от несанкционированного доступа является процедура логического входа. Операционная система должна убедиться, что в систему пытается войти пользователь, вход которого разрешен администратором. Функции защиты ОС вообще очень тесно связаны с функциями администрирования, так как именно администратор определяет права пользователей при их обращении к разным ресурсам системы – файлам, каталогам, принтерам, сканерам и т. п. Кроме того, администратор ограничивает возможности пользователей в выполнении тех или иных системных действий. Например, пользователю может быть запрещено выполнять процедуру завершения работы ОС, устанавливать системное время, завершать чужие процессы, создавать учетные записи пользователей, изменять права доступа к некоторым каталогам и файлам. Администратор может также урезать возможности пользовательского интерфейса, убрав, например, некоторые пункты из меню операционной системы, выводимого на дисплей пользователя.

Важным средством защиты данных являются функции аудита ОС, заключающиеся в фиксации всех событий, от которых зависит безопасность системы. Например, попытки удачного и неудачного логического входа в систему, операции доступа к некоторым каталогам и файлам, использование принтеров и т. п. Список событий, которые необходимо отслеживать, определяет администратор ОС. Поддержка отказоустойчивости реализуется операционной системой, как правило, на основе резервирования. Чаще всего в функции ОС входит поддержание нескольких копий данных на разных дисках или разных дисковых накопителях. Резервируются также принтеры и другие устройства ввода-вывода. При отказе одного из избыточных устройств операционная система должна быстро и прозрачным для пользователя образом произвести реконфигурацию системы и продолжить работу с резервным устройством. Особым случаем обеспечения отказоустойчивости является использование нескольких процессоров, то есть мультипроцессирование, когда система продолжает работу при отказе одного из процессоров, хотя и с меньшей производительностью. (Необходимо отметить, что многие ОС используют мультипроцессорную конфигурацию компьютера только для ускорения работы и при отказе одного из процессоров прекращают работу.) Поддержка отказоустойчивости также

входит в обязанности системного администратора. В состав ОС обычно входят утилиты, позволяющие администратору выполнять регулярные операции резервного копирования для обеспечения быстрого восстановления важных данных.

Интерфейс прикладного программирования

Прикладные программисты используют в своих приложениях обращения к ОС, когда для выполнения тех или иных действий им требуется особый статус, которым обладает только операционная система. Например, в большинстве современных ОС все действия, связанные с управлением аппаратными средствами компьютера, может выполнять только ОС. Помимо этих функций прикладной программист может воспользоваться набором сервисных функций ОС, которые упрощают написание приложений. Функции такого типа реализуют универсальные действия, часто требующиеся в различных приложениях, такие, например, как обработка текстовых строк. Эти функции могли бы быть выполнены и самим приложением, однако гораздо проще использовать уже готовые, отлаженные процедуры, включенные в состав операционной системы. В то же время даже при наличии в ОС соответствующей функции программист может реализовать ее самостоятельно в рамках приложения, если предложенный операционной системой вариант его не вполне устраивает.

Возможности операционной системы доступны прикладному программисту в виде набора функций, называющегося интерфейсом прикладного программирования (Application Programming Interface, API). От конечного пользователя эти функции скрыты за оболочкой алфавитно-цифрового или графического пользовательского интерфейса.

Для разработчиков приложений все особенности конкретной операционной системы представлены особенностями ее API. Поэтому операционные системы с различной внутренней организацией, но с одинаковым набором функций API кажутся им одной и той же ОС, что упрощает стандартизацию операционных систем и обеспечивает переносимость приложений между внутренне различными ОС, соответствующими определенному стандарту на API. Например, следование общим стандартам API UNIX, одним из которых является стандарт Posix, позволяет говорить о некоторой обобщенной операционной системе UNIX, хотя многочисленные версии этой ОС от разных производителей иногда существенно отличаются внутренней организацией.

Приложения выполняют обращения к функциям API с помощью системных вызовов. Способ, которым приложение получает услуги операционной системы, очень похож на вызов подпрограмм. Информация, нужная ОС и состоящая обычно из идентификатора команды и данных, помещается в определенное место памяти, в регистры и/или стек. Затем управление передается операционной системе, которая выполняет требуемую функцию и возвращает результаты через память, регистры или стеки. Если операция проведена неуспешно, то результат включает индикацию ошибки.

Способ реализации системных вызовов зависит от структурной организации ОС, которая, в свою очередь, тесно связана с особенностями аппаратной платформы. Кроме того, он зависит от языка программирования. При использовании ассемблера программист устанавливает значения регистров и/или областей памяти, а затем выполняет специальную инструкцию вызова сервиса или программного прерывания для обращения к некоторой функции ОС. При использовании языков высокого уровня функции ОС вызываются тем же способом, что и написанные пользователем подпрограммы, требуя задания определенных

аргументов в определенном порядке.

Пользовательский интерфейс

Операционная система должна обеспечивать удобный интерфейс не только для прикладных программ, но и для человека, работающего за терминалом. Этот человек может быть конечным пользователем, администратором ОС или программистом.

В ранних операционных системах пакетного режима функции пользовательского интерфейса были сведены к минимуму и не требовали наличия терминала. Команды языка управления заданиями набивались на перфокарты, а результаты выводились на печатающее устройство.

Современные ОС поддерживают развитые функции пользовательского интерфейса для интерактивной работы за терминалами двух типов: алфавитно-цифровыми и графическими.

При работе за алфавитно-цифровым терминалом пользователь имеет в своем распоряжении систему команд, мощность которой отражает функциональные возможности данной ОС. Обычно командный язык ОС позволяет запускать и останавливать приложения, выполнять различные операции с файлами и каталогами, получать информацию о состоянии ОС (количество работающих процессов, объем свободного пространства на дисках и т. п.), администрировать систему. Команды могут вводиться не только в интерактивном режиме с терминала, но и считываться из так называемого командного файла, содержащего некоторую последовательность команд.

Программный модуль ОС, ответственный за чтение отдельных команд или же последовательности команд из командного файла, иногда называют командным интерпретатором.

Ввод команды может быть упрощен, если операционная система поддерживает графический пользовательский интерфейс. В этом случае пользователь для выполнения нужного действия с помощью мыши выбирает на экране нужный пункт меню или графический символ.

Требования к современным операционным системам

Главным требованием, предъявляемым к операционной системе, является выполнение ею основных функций эффективного управления ресурсами и обеспечение удобного интерфейса для пользователя и прикладных программ. Современная ОС, как правило, должна поддерживать мультипрограммную обработку, виртуальную память, свопинг, многооконный графический интерфейс пользователя, а также выполнять многие другие необходимые функции и услуги. Кроме этих требований функциональной полноты к операционным системам предъявляются не менее важные эксплуатационные требования, которые перечислены ниже.

- **Расширяемость.** В то время как аппаратная часть компьютера устаревает за несколько лет, полезная жизнь операционных систем может измеряться десятилетиями. Примером может служить ОС UNIX. Поэтому операционные системы всегда изменяются со временем эволюционно, и эти изменения более значимы, чем изменения аппаратных средств. Изменения ОС обычно заключаются в приобретении ею новых свойств, например поддержке новых типов внешних

устройств или новых сетевых технологий. Если код ОС написан таким образом, что дополнения и изменения могут вноситься без нарушения целостности системы, то такую ОС называют расширяемой. Расширяемость достигается за счет модульной структуры ОС, при которой программы строятся из набора отдельных модулей, взаимодействующих только через функциональный интерфейс.

- **Переносимость.** В идеале код ОС должен легко переноситься с процессора одного типа на процессор другого типа и с аппаратной платформы (которые различаются не только типом процессора, но и способом организации всей аппаратуры компьютера) одного типа на аппаратную платформу другого типа. Переносимые ОС имеют несколько вариантов реализации для разных платформ, такое свойство ОС называют также многоплатформенностью.
- **Совместимость.** Существует несколько «долгоживущих» популярных операционных систем (разновидности UNIX, MS-DOS, Windows 3.x, Windows NT, OS/2), для которых наработана широкая номенклатура приложений. Некоторые из них пользуются широкой популярностью. Поэтому для пользователя, переходящего по тем или иным причинам с одной ОС на другую, очень привлекательна возможность запуска в новой операционной системе привычного приложения. Если ОС имеет средства для выполнения прикладных программ, написанных для других операционных систем, то про нее говорят, что она обладает совместимостью с этими ОС. Следует различать совместимость на уровне двоичных кодов и совместимость на уровне исходных текстов. Понятие совместимости включает также поддержку пользовательских интерфейсов других ОС.
- **Надежность и отказоустойчивость.** Система должна быть защищена как от внутренних, так и от внешних ошибок, сбоев и отказов. Ее действия должны быть всегда предсказуемыми, а приложения не должны иметь возможности наносить вред ОС. Надежность и отказоустойчивость ОС прежде всего определяются архитектурными решениями, положенными в ее основу, а также качеством ее реализации (отлаженностью кода). Кроме того, важно, включает ли ОС программную поддержку аппаратных средств обеспечения отказоустойчивости, таких, например, как дисковые массивы или источники бесперебойного питания.
- **Безопасность.** Современная ОС должна защищать данные и другие ресурсы вычислительной системы от несанкционированного доступа. Чтобы ОС обладала свойством безопасности, она должна как минимум иметь в своем составе средства аутентификации – определения легальности пользователей, авторизации – предоставления легальным пользователям дифференцированных прав доступа к ресурсам, аудита – фиксации всех «подозрительных» для безопасности системы событий. Свойство безопасности особенно важно для сетевых ОС. В таких ОС к задаче контроля доступа добавляется задача защиты данных, передаваемых по сети.
- **Производительность.** Операционная система должна обладать настолько хорошим быстродействием и временем реакции, насколько это позволяет аппаратная платформа. На производительность ОС влияет много факторов, среди которых основными являются архитектура ОС, многообразие функций, качество программирования кода, возможность исполнения ОС на высокопроизводительной (многопроцессорной) платформе.

Выводы

История ОС насчитывает примерно полвека. Она во многом определялась и определяется развитием элементной базы и вычислительной аппаратуры.

Первые цифровые вычислительные машины, появившиеся в начале 40-х годов, работали без операционных систем, все задачи организации вычислительного процесса решались вручную каждым программистом с пульта управления.

Прообразом современных операционных систем явились мониторные системы середины 50-х, которые автоматизировали действия оператора по выполнению пакета заданий.

В 1965-1975 годах переход к интегральным микросхемам открыл путь к появлению следующего поколения компьютеров, ярким представителем которых является IBM/360. В этот период были реализованы практически все основные концепции, присущие современным ОС: мультипрограммирование, мультипроцессирование, многотерминальный режим, виртуальная память, файловые системы, разграничение доступа и сетевая работа.

Реализация мультипрограммирования потребовала внесения очень важных изменений в аппаратуру компьютера. В процессорах появился привилегированный и пользовательский режимы работы, специальные регистры для быстрого переключения с одной задачи на другую, средства защиты областей памяти, а также развитая система прерываний.

К середине 70-х годов широкое распространение получили мини-компьютеры. Архитектура мини-компьютеров была значительно упрощена по сравнению с мэйнфреймами, что нашло отражение и в их ОС. Экономичность и доступность мини-компьютеров послужила мощным стимулом для создания локальных сетей.

С середины 70-х годов началось массовое использование UNIX, уникальной для того времени ОС, которая сравнительно легко переносилась на различные типы компьютеров. Хотя ОС UNIX была первоначально разработана для мини-компьютеров, ее гибкость, элегантность, мощные функциональные возможности и открытость позволили ей занять прочные позиции во всех классах компьютеров.

Начало 80-х годов связано с появлением персональных компьютеров, которые послужили мощным катализатором для бурного роста локальных сетей, создав для этого отличную материальную основу в виде десятков и сотен компьютеров, расположенных в пределах одного здания. В результате поддержка сетевых функций стала для ОС персональных компьютеров необходимым условием.

К началу 90-х практически все ОС стали сетевыми, способными поддерживать работу с разнородными клиентами и серверами. Появились специализированные сетевые ОС, предназначенные исключительно для выполнения коммуникационных задач.

Особое внимание в течение последнего времени уделяется корпоративным сетевым ОС, для которых характерны высокая степень масштабируемости, поддержка сетевой работы, развитые средства обеспечения безопасности, способность работать в гетерогенной среде, наличие средств централизованного администрирования и управления.

ОС – это комплекс взаимосвязанных программ, предназначенный для повышения эффективности аппаратуры компьютера путем рационального управления его ресурсами, а также для обеспечения удобств пользователю путем предоставления ему расширенной

виртуальной машины.

Операционные системы могут различаться особенностями реализации внутренних алгоритмов управления основными ресурсами компьютера (процессорами, памятью, устройствами), типами аппаратных платформ, областями использования, особенностями использованных методов проектирования ОС и т.д.

К числу основных ресурсов, управление которыми осуществляет ОС, относятся процессоры, основная память, таймеры, наборы данных, диски, накопители на магнитных лентах, принтеры, сетевые устройства и некоторые другие. Ресурсы распределяются между процессами. Для решения задач управления ресурсами разные ОС используют различные алгоритмы, особенности которых в конечном счете и определяют облик ОС.

Наиболее важными подсистемами ОС являются подсистемы управления процессами, памятью, файлами и внешними устройствами, а также подсистемы пользовательского интерфейса, защиты данных и администрирования.

Прикладному программисту возможности ОС доступны в виде набора функций, составляющих интерфейс прикладного программирования (API).