**1.1.1. Определение ОС**

**Операционная система** — это комплект программ, которые служат интерфейсом между модулями вычислительных систем и прикладными программными приложениями, а также управляют компьютерным оборудованием и процессами вычислений, эффективным распределением вычислительных мощностей среди процессов вычислений.

**Операционная система** — это комплекс программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем.

**1.1.2. Назначение операционных систем**

ОС управляет всеми устройствами компьютерной системы и обеспечивает пользователя удобным интерфейсом для работы с аппаратурой.

**Две основные функции (назначение) ОС:**

1. предоставлять пользователю расширенную виртуальную машину, с которой легче работать, чем с аппаратурой реального компьютера или реальной сети;
2. управлять ресурсами вычислительной системы.

**1.1.3. Состав ОС**

**Наиболее важные Функции ОС:**

1. подсистема управления процессами,
2. подсистема управления памятью,
3. подсистема управления файлами, подсистема управления внешними устройствами,
4. подсистема пользовательского интерфейса,
5. подсистема защиты данных и администрирования.

**1.1.4. Функции ОС**

1. прием от пользователя заданий или команд, сформулированных на соответствующих языках, и их обработка;

2. загрузка в ОП программ и их исполнение;

3. инициация программы;

4. прием и исполнение программных запросов на запуск, приостановку, остановку других программ;

5. организация взаимодействия между задачами;

6. идентификация всех программ и данных;

7. обеспечение работы системы управления файлами и/или систем управления БД;

8. обеспечение режима мультипрограммирования (многозадачности);

9. планирование и диспетчеризация задач;

10. обеспечение функций по организации и управлению операциями ввода/вывода;

11. удовлетворение жестким ограничениям на время ответа в режиме реального времени (для соответствующих ОС);

12. управление памятью, организация виртуальной памяти;

13. организация механизмов обмена сообщениями и данными между

выполняющимися программами;

1. защита одной программы от влияния другой; обеспечение сохранности данных;
2. аутентификация, авторизация и другие средства обеспечения безопасности;
3. предоставление услуг на случай частичного сбоя системы;
4. обеспечение работы систем программирования;
5. параллельное исполнение нескольких задач.

**1.1.5. Классификация ОС**

1. В зависимости от **способа организации вычислений**:

**Системы пакетной обработки** – основной задачей является организация наибольшего количества вычислительных процессов за единицу времени.

**Системы разделения времени** – создание возможности единовременного взаимодействия с устройством сразу несколькими людьми.

**Системы реального времени** – организация работы каждой задачи за

определенный промежуток времени, присущий каждой конкретной задаче.

1. В зависимости от **типа ядра**:

OС с монолитным ядром;

OС с микроядром;

OС с гибридным ядром.

1. В зависимости от **количества единовременно решаемых задач**: однозадачные, многозадачные;
2. В зависимости от **количества пользователей**: однопользовательские, многопользовательские.
3. В зависимости от **количества поддерживаемых процессоров**: однопроцессорные, многопроцессорные;
4. В зависимости от **возможности работы в компьютерной сети**:

**локальные** – автономные ОС, которые не позволяют работать с компьютерными сетями;

**сетевые** – ОС с поддержкой компьютерных сетей.

1. В зависимости от **роли в сетевом взаимодействии**:

**серверные** – ОС, открывающие доступ к ресурсам сети и осуществляющие управление сетевой инфраструктурой;

**клиентские** – ОС, которые имеют возможность получения доступа к ресурсам сети.

1. В зависимости от **типа лицензии**:

**открытые** – ОС с открытым исходным кодом;

**проприетарные** – ОС, связанные с определенным правообладателем и, как правило, имеющие закрытый исходный код.

1. В зависимости от **сферы использования**:

ОС мэйнфреймов – больших компьютеров;

ОС серверов;

ОС персональных компьютеров;

OC мобильных устройств;

встроенные OC;

OC маршрутизаторов.

**1.1.6. Основные этапы развития ОС, поколения ОС**

Первый настоящий цифровой компьютер был изобретен английским математиком **Чарльзом Бэббиджем**, но он был механическим и так и не заработал должным образом.

Появление электронных приборов сделало возможным появление электронных вычислительных машин. В англоязычной среде использовалось слово компьютер (вычислитель).

**Первое поколение (1945-1955 гг.)**

**Первое поколение** компьютеров строилось преимущественно на **электронных лампах.** В начале программы писались в машинных кодах и загружались в память компьютера **с панели переключателей вручную.**

Позже появилась возможность загрузки с **перфокарт**. Вычислительная система выполняла одновременно только **одну операцию**. Так как электронные лампы часто выходили из строя, не было необходимости в постоянно работающей части программного обеспечения. В самом конце этого периода появляется первое системное программное обеспечение, например, компилятор **Fortran** и **ассемблер для**

**IBM-701.**

Первый период характеризуется крайне высокой стоимостью вычислительных систем, их малым количеством и низкой эффективностью использования.

**Второе поколение (1955-1965 гг.)**

**Второе поколение** компьютеров характеризуется использованием **транзисторов**, что повысило их надёжность и продлило время непрерывной работы. Подготовка компьютера к выполнению задания приводило к расходованию дорогостоящего машинного времени. Было принято решение использовать небольшие недорогие компьютеры для записи заданий на магнитную ленту и затем для настоящих вычислений использовали мощные дорогостоящие машины. Программа, которая считывала задания с ленты и запускала их на выполнение, является **прообразом современной операционной системы.**

Пакетное задание кроме текста программ содержало специальные перфокарты $JOB, $LOAD, $RUN, $END (прототип современных команд в командной оболочке CLI). Системное программное обеспечение обычно хранилось на магнитных лентах. Большие компьютеры второго поколения использовались главным образом для **научных и технических вычислений**. В основном программы для них составлялись на **языке Фортран и ассемблере,** а типичными операционными системами были FMS (Fortran Monitor System) и IBSYS (операционная система, созданная корпорацией IBM для компьютера IBM 7094). Появлялись новые языки программирования: LISP, COBOL, ALGOL-60, PL/1 и т. д.). В этот период появляются **первые настоящие компиляторы, редакторы связей, библиотеки математических и служебных подпрограмм.**

**Третье поколение (1965-1980 гг.)**

Компьютеры **третьего поколения** использовали **малые интегральные схемы,** что дало им преимущество в цене и качестве по сравнению с машинами второго поколения.

Самым важным достижением явилась **многозадачность**. В компьютерах предыдущего поколения вычисления останавливались на время ввода-вывода. Проблема была решена разбиением памяти на несколько частей. Пока одно задание ожидало завершения работы устройства ввода-вывода, другое могло использовать центральный процессор.

Другим важным плюсом операционных систем третьего поколения стала способность считывать задание с перфокарт на диск по мере того, как их приносили в машинный зал.

При окончании выполнения каждого текущего задания операционная система могла загружать новое задание с диска в освободившийся раздел памяти и запускать это задание. Этот технический прием называется **подкачкой данных**, или **спулингом** (spooling — английское слово, которое произошло от Simultaneous Peripheral Operation On Line, то есть совместная периферийная операция в интерактивном режиме).

Многозадачность требует как **аппаратной поддержки,** так и **поддержки со стороны операционной системы** .

В системах третьего поколения промежуток времени между передачей задания и возвращением результатов часто составлял **несколько часов**. Желание сократить время ожидания ответа привело к разработке **режима разделения времени** — варианту многозадачности, при котором у каждого пользователя есть свой диалоговый терминал.

В начале третьего периода появилась идея создания семейств программно совместимых машин, работающих под управлением одной и той же операционной системы. Это облегчило работу прикладных программистов, но широкие возможности этой концепции породили сложную и громоздкую операционную систему.

Примеры операционных систем третьего поколения — OS/360, CTSS, MULTICS. Упрощенная однопользовательская версия системы MULTICS позже переросла в операционную систему UNIX®, ставшую популярной в академических кругах, правительственных учреждениях и во многих компаниях.

**Четвертое поколение (1980-2005 гг.)**

Разработка БИС (большие интегральные схемы, LSI, Large Scale Integration — кремниевые микросхемы, содержащие тысячи транзисторов на одном квадратном сантиметре) привела к появлению **микрокомпьютеров**. Главная особенность таких компьютеров — весь **процессор помещался в одной микросхеме.** Размеры компьютера и его стоимость резко уменьшились, что предоставило возможность купить персональный компьютер каждому человеку.

Первой операционной системой для микропроцессора Intel 8080 стала **CP/M.** Для процессора Intel 8086 была разработана операционная система MS-DOS.

Первоначально персональные компьютеры предназначались для использования одним пользователем в **однопрограммном** режиме, что повлекло за собой деградацию архитектуры этих ЭВМ и их операционных систем (в частности, пропала необходимость защиты файлов и памяти, планирования заданий и т. п.).

Компьютеры стали использоваться не только специалистами, что потребовало разработки «дружественного» программного обеспечения.

Первоначально системы четвертого поколения работали в текстовом режиме, позже появился графический интерфейс. И с этого времени появляется множество операционных систем. Это Mac OS X от Apple, семейство Windows от Microsoft, а также большое количество UNIX-подобных систем.

В середине 1980-х годов стали бурно развиваться сети компьютеров, в том числе персональных, работающих под управлением сетевых или *распределенных операционных* систем. Распределённые системы могут объединять большое число компьютеров, поэтому наиболее важный аспект их дизайна — *масштабируемость* (способность увеличивать производительность пропорционально добавленным ресурсам).

**Пятое поколение (2005 г. - по н.в.)**

Первый настоящий смартфон появился только в середине 1990-х годов, когда Nokia выпустила свой N9000, представлявший собой комбинацию из двух отдельных устройств: **телефона и КПК.** В 1997 году в компании Ericsson для ее изделия GS88 «Penelope» был придуман термин «смартфон».

Но действительно радикальные изменения произошли **в 2005 году с появлением многоядерных процессоров. Популярные ОС Windows и Linux** были успешно адаптированы для работы на многопроцессорной (многоядерной) архитектуре.

На рынке мобильных устройств между различными операционными системами воцарилась жесткая конкуренци. Доминирующей считается операционная система Google Android, а на втором месте находилась Apple iOS.

В первое десятилетие после своего появления большинство смартфонов работало под управлением Symbian OS. Эту операционную систему выбрали такие популярные бренды, как Samsung, Sony Ericsson, Motorola и Nokia. Но долю рынка Symbian начали отбирать другие операционные системы, например, RIM Blackberry OS и Apple iOS.

Многие ожидали, что RIM будет доминировать на рынке бизнес-устройств, а iOS завоюет рынок потребительских устройств. Для рынка популярность Symbian упала. В 2011 году Nokia отказалась от Symbian и объявила о своем намерении в качестве основной платформы сосредоточиться на Windows Phone. Некоторое время операционные системы от Apple и RIM всех устраивали, но вскоре всех своих соперников обогнала основанная на ядре Linux операционная система Android, выпущенная компанией Google в 2008 году.

Для производителей телефонов Android обладала тем преимуществом, что имела **открытый исходный код** и была доступна по разрешительной лицензии. У этой операционной системы имеется огромное сообщество разработчиков, создающих приложения в основном на общеизвестном языке программирования **Java**. Широкомасштабное применение многоядерных процессоров привело к созданию многочисленных компьютеров большой мощности. Обычно такие системы собираются по кластерной технологии, состоят из множества автономных узлов с распределенной памятью, зачастую имеющих несколько процессоров и/или ядер, и отличаются способами соединения отдельных узлов.

Широкую популярность приобрела технология **виртуализации** — представление вычислительных ресурсов, абстрагированное от аппаратной реализации. Важным примером использования виртуализации являются об**лачные технологии**.

**Системы реального времени**

Системы реального времени характеризуются тем, что **время для них является ключевым параметром.** Например, в системах управления производственными процессами компьютеры, работающие в режиме реального времени, должны собирать сведения о процессе и использовать их для управления станками на предприятии. Например, когда автомобиль перемещается по сборочному конвейеру. Если операция должна быть проведена точно в срок (или в определенный период времени), то мы имеем дело с системой жесткого реального времени.

Другой разновидностью подобных систем является система мягкого реального времени, в которой хотя и нежелательно, но вполне допустимо несоблюдение срока какого-нибудь действия, что не наносит непоправимого вреда. К этой категории относятся цифровые аудио- или мультимедийные системы. Смартфоны также являются системами мягкого реального времени.

**1.1.7. POSIX**

Чтобы появилась возможность писать программы, работающие в любой UNIX-системе, Институт инженеров по электротехнике и электронике (IEEE) разработал стандарт системы UNIX, названный **POSIX**. Стандарт POSIX определяет минимальный интерфейс системных вызовов, который должны поддерживать совместимые с ним системы UNIX.

Фактически на данный момент POSIX-интерфейс поддерживается также рядом других операционных систем.

**POSIX** (англ. Portable Operating System Interface — переносимый интерфейс операционных систем) — набор стандартов, описывающих интерфейсы между операционной системой и прикладной программой (системный API), библиотеку языка C и набор приложений и их интерфейсов. Стандарт создан для обеспечения совместимости различных UNIX-подобных операционных систем и переносимости прикладных программ на уровне исходного кода, но может быть использован и для не-Unix систем.

Серия стандартов POSIX была разработана комитетом 1003 IEEE. Международная организация по стандартизации (ISO) совместно c Международной электротехнической комиссией (IEC) приняла стандарт POSIX под названием ISO/IEC 9945. Версии стандарта POSIX являются основой соответствующих версий стандарта Single UNIX Specification.

Стандарт POSIX определяет интерфейс операционной системы, а соответствие стандарту Single UNIX Specification определяет реализацию интерфейса и позволяет операционным системам использовать торговую марку UNIX.

**1.1.8. Эксплуатационные требования к ОС**

Очевидно, что главным требованием, предъявляемым к операционной системе, является способность выполнения основных функций: **эффективного управления ресурсами и обеспечения удобного интерфейса для пользователя и прикладных програм.**

Кроме этих функциональных требований, к операционным системам предъявляются не менее важные эксплуатационные требования. К этим требованиям относятся:

• **расширяемость**. Код операционной системы должен быть написан таким образом, чтобы можно было легко внести дополнения и изменения;

• **переносимость**. Код операционной системы должен легко переноситься с процессора одного типа на процессор другого типа и с аппаратной платформы одного типа на аппаратную платформу другого типа;

• **надежность и отказоустойчивость**. Система должна быть защищена как от

внутренних, так и от внешних ошибок, сбоев и отказов;

• **совместимость**. Операционная система должна иметь средства для выполнения прикладных программ, написанных для других операционных систем;

• **безопасность**. Операционная система должна обладать средствами защиты ресурсов одних пользователей от других;

• **производительность**. Система должна обладать настолько хорошим быстродействием и временем реакции, насколько это позволяет аппаратная платформа.

**1.1.9. Особенности современного этапа и перспективы развития ОС**

На современном этапе развития OC 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.

Современным операционным системам присуща **многоплатформенность**, т.е. способность работать на совершенно различных типах компьютеров.

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.

Постоянно повышается удобство интерактивной работы с компьютером путем включения в операционную систему развитых графических интерфейсов, использующих наряду с графикой звук и видеоизображение.

Пользовательский интерфейс операционной системы становится все более интеллектуальным, направляя действия человека в типовых ситуациях и принимая за него рутинные решения.

Пока пользователи и администраторы сети тратят значительное время на попытки выяснить, где находится тот или иной ресурс, разработчики сетевых приложений прилагают много усилий для определения местоположения данных и программных модулей в сети. Операционные системы будущего должны обеспечить высокий уровень прозрачности сетевых ресурсов, взяв на себя задачу организации распределенных вычислений, превратив сеть в виртуальный компьютер.

**Квантовые вычисления**

Еще одним направлением и перспективой в развитии персональных компьютеров считается **квантовый компьютер.** Это устройство, работа которого базируется на явлениях квантовой механики). Для вычислительных действий квантовый компьютер оперирует совершенно другими единицами данных — не битами, в **кубитами** (Quantum Bits).

Отличительная особенность этих единиц измерения заключается в том, что биты могут принимать два значения: или 0, или 1. В то время как кубиты могут быть одновременно реализованы в виде 0 или 1.

Обобщением понятия кубит является кудит, способный хранить в одном разряде более двух значений, кутрит, куквадрит и т. д. По мнению ученых, это даст существенное преимущество — квантовое превосходство — новому поколению персональных компьютеров:

**Недостатки**

Квантовые кубиты настолько нестабильны, что чувствительны к тепловому шуму материи, поэтому вычислительные ячейки охлаждают жидким азотом.

Поскольку у кубита есть определенная вероятность нахождения в состоянии 1 или 0, всегда есть и вероятность ошибки. Чем больше кубитов в системе, тем больше суммарная вероятность, что система выдаст неправильный ответ, поэтому зачастую приходится проводить несколько расчетов одной и той же задачи или считать одну и ту же задачу на нескольких компьютерах. Кроме того, в силу квантовой природы вычислений, ответ всегда будет содержать в себе возможность ошибки, это неустранимый фактор, хотя его можно минимизировать.

**Перспективы**

В Японии в объединение Quantum Strategic Alliance for Revolution (Q-STAR) входят Toyota Motor, Hitachi и NTT и другие крупные организации. К середине следующего года страна планирует ввести в эксплуатацию свой первый квантовый компьютер. К 2030 году, по прогнозам японцев, квантовую технологию будут использовать 10 миллионов человек.

Alphabet, материнская компания Google, тратит на квантовые вычисления миллиарды долларов с целью создания к 2029 году коммерческого квантового компьютера, который сможет выполнять крупномасштабные вычисления. Также в квантовую технику вкладываются IBM, Intel, Microsoft и другие компании. Исследовательские группы, которые занимаются разработкой и исследованием квантовых компьютеров, есть чуть ли не в каждом большом американском институте.