**2 Вопрос:** Архитектура операционной системы и требования, предъявляемые к современным системам (переносимость, совместимость, расширяемость и т.д.)

. Операционная система состоит из четырех основных частей:

1. **Первая часть**— ядро, низкоуровневая основа любой ОС, выполняемая аппаратурой в особом **привилегированном режиме**. Ядро загружается в память один раз и находится в памяти **резидентно**– постоянно, по одним и тем же адресам. Ядро — командный интерпретатор, «переводчик» с программного языка на «железный», язык машинных кодов.
2. **Вторая часть**— **Подсистема управления ресурсами (resource allocator)** — управляет вычислительными ресурсами компьютера — оперативной и внешней памятью, процессором
3. **Третья часть — Управляющая программа (control program, supervisor) –**управляет исполнением других программ и функционированием устройств ввода-вывода.(используются специализированные программы для управления различными устройствами, входящими в состав компьютера. Драйвера «системные библиотеки»)
4. **Четвертая часть** — удобная оболочка, с которой общается пользователь — интерфейс. Своего рода красивая обертка, в которую упаковано скучное и не интересное для пользователя ядро.

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 на­ рушения 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 при­ влекательна 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/ 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.

**3 3 Вопрос:** Подходы к проектированию OС на примере структуры Windows\*

**4 Вопрос:** Характеристика основных компонент ОС (программы: управляющая, системные обрабатывающие, обеспечения теледоступа и интерактивной графики).

Программы ОС постоянно занимают в оперативной памяти объем, установленный при конфигурации системы. Остальные части ОС по мере необходимости вызываются из внешней памяти на МД.

ОС обеспечивает осуществление в вычислительной системе следующих процессов:

* обработка задач;
* работы системы в режиме диалога и квантования времени;
* работы в системе в реальном масштабе времени в составе многопроцессорных и многомашинных комплексов;
* связи оператора с системой;
* протоколирование хода выполнения вычислительных работ;
* обработки данных, поступающих по каналам связи;
* функционирование устройств ввода-вывода;
* использование широкого набора средств отладки и тестирование программ;
* планирование прохождения задач в соответствии с их приоритетами;
* ведение учета и контроля за использованием данных, программ и ресурсов ЭВМ.

Основные компоненты ОС – управляющие и обрабатывающие программы. Управляющие программы управляют работой вычислительной системы, обеспечивая в свою очередь автоматическую смену заданий для поддержания непрерывного режимы работы ЭВМ при переходе от одной программы к другой без вмешательства оператора.

Управляющая программа определяет порядок выполнения обрабатывающих программ и обеспечивает необходимым набором услуг для их выполнения. Основные функции: последовательное или приоритетное выполнение каждой работы (управление задачами); хранение, поиск и обслуживание данных независимо от их организации и способа хранения (управление данными).

Программы управления задачами считывают входные потоки задач, обрабатывают их в зависимости от приоритета, инициируют одновременное выполнение нескольких заданий; вызывают процедуры; ведут системный журнал.

Программы управления данными обеспечивают способы организации, идентификации, хранения, каталогизации и выборки обрабатываемых данных. Эти программы управляют вводом-выводом данных с различной организацией, объединением записей в блоки и разделением блоков на записи, обработки меток томов и наборов данных.

Программы управления восстановления после сбоя обрабатывают прерывания от системы контроля, регистрируют сбои в процессоре и внешних устройствах, формируют записи о сбое в журнале, анализируют возможность завершение сбоем задачи и переводят систему в состояние ожидания, если завершение задачи невозможно.

Конфигурация системы. Прикладная программа в ОС может получить от ОС в процессе своей работы характеристик конкретной реализации системы, в среде которой она функционирует: имя, версию и редакцию ОС, тип и технические характеристики комп-а. В ОС обычно имеются средства локализации, позволяющие настроить систему на конкретное национальное (местное) представление данных: представление десятичных дробей, денежных величин, даты и времени.

**Вопрос:** Понятие Ядра ОС и различные подходы к проектированию ОС.

**1. Монолитное ядро**- с точки зрения программного кода все компоненты программного кода видят друг друга (полная прозрачность всех процедур и подпрограмм). Три слоя монолитного ядра: верхний слой содержит главную программу (точка входа), средний - системные вызовы, третий слой - утилиты для работы с аппаратным обеспечением. Программное обеспечение перед тем как выполнять системный вызов помещает системный данные в качестве параметров в памяти (код данных и операций) > главная программа считывает, что требуется ПО > выбирается сервисная процедура (сколько нужно выполнить столько и будет сервизных процедур) > сервизная процедура может вызвать несколько соответствующих утилит > утилиты взаимодействует с аппаратурой > главная программа возвращает ПО ответ либо можно делать / либо сообщение об ошибке. **Преимущества:** надежность (никакое ПО не может вмешаться в работу ядра ОС), минимальные накладные расходы (все прозрачно, без дополнительных интерфейсов). **Недостатки:**очень плохая изменяемость - изменение одной функции ведет к изменению всей ОС, исходный код ядра менялся под новые версии драйверов и тд, перекомпиляция ядра). Применяются во встраиваемых программах.

Монолитное ядро (МнЯ) предоставляет широкий набор абстракций оборудования. Все части ядра работают в одном адресном пространстве. МнЯ требуют перекомпиляции при изменении состава оборудования. Компоненты операционной системы являются не самостоятельными модулями, а составными частями одной программы. МнЯ более производительно, чем микроядро, поскольку работает как один большой процесс. МнЯ является большинство Unix-систем и Linux. Монолитность ядер усложняет отладку, понимание кода ядра, добавление новых функций и возможностей, удаление ненужного, унаследованного от предыдущих версий кода. "Разбухание" кода монолитных ядер также повышает требования к объёму оперативной памяти.

**2. Многослойная монолитная архитектура.**Схема данной архитектуры в центре находится железо. **Первый слой - аппаратные средства поддержки ядра** (микропрограммы записанные в чипсете материнской платы, например, для перехода в привилегированном режим ОС, защита памяти, системный таймер, механизм прерываний, смена контекста ( )). Именно из-за первого слоя не на все железо можно поставить определенные ОС (если на первом слое это не реализовано, то невозможно). **Второй слой (первый программным слой) - машинно-зависимые модули (слой абстракции железа HAL)** - фактически драйвера для железа (чтобы ОС не зависима от конкретной аппаратной платформы). **3 слой - базовые механизмы ядра**- на нем реализуется все основные функции ОС (открытие/закрытие файлов, подкачка и тд). Однако данный уровень является только исполнительным, принятие решений здесь не происходит. **4. Менеджеры ресурсов (слой принятия решения)**- реализованы все алгоритмы принятия решений (какой процесс будет выполняться следующим, какие файлы открыть). **Последний слой - системные вызовы (интерфейс API)**- слой отвечает за взаимодействие ПО и ОС (ПО знает о существовании только этого слоя). В таком режиме все слои резиденты в оперативной памяти. Преимущества: реорганизация ядра становится легче (изменение требуется только на одном слое), процессорная производительность довольно высока (на переключение между режимами ядра и приложения время минимально 2дельтаТ(туда-обратно)). Недостатки: большие накладные расходы по оперативной памяти, ПО не может выполняться на нескольких аппаратных средствах (ядра не взаимодействует друг с другом).

**3. Микроядерная архитектура. Идея микро ядерной архитектуры** (4 и 5 слои вынесем в отдельные программы). В ядре будет содержатся два слоя: машинно-зависимые модули и базовые механизмы ядра. В пользовательское режимы будут находиться сервера (памяти, вычислений), ПО, и все они равноправно могут вызывать ядро. Идея заключается в том, что есть вещи, которые обязательно нужно выполнять в ядре (выделение памяти), а принятие решений будут существовать отдельно как пользовательское ПО (в Microsoft это называется службами). Отсюда сервера не находятся в оперативной памяти, и появляются там только по требованию. В такой ситуации накладные расходы 4дельтаТ. Недостатки: большие процессорные накладные расходы, резкое снижение надежности. Преимущество: удобство работы с несколькими аппаратными системами, занимаем мало оперативной памяти.

Микроядерная архитектура предоставляет только элементарные функции управления процессами и минимальный набор абстракций для работы с оборудованием. Большая часть работы осуществляется с помощью специальных пользовательских процессов, называемых сервисами. В микроядерной операционной системе можно, не прерывая ее работы, загружать и выгружать новые драйверы, файловые системы и т. д. Микроядерными являются ядра ОС *Minix* и GNU Hurd и ядро систем семейства BSD. Классическим примером микроядерной системы является *Symbian OS*. Это пример распространенной и отработанной микроядерной (a начиная c версии *Symbian OS* v8.1, и наноядерной) операционной системы.

**4. Наноядерная архитектура** - в ядро входят только машинно-зависимые модули. Крайне упрощённое и минимальное ядро, выполняет лишь одну задачу – обработку аппаратных прерываний, генерируемых устройствами компьютера. После обработки посылает информацию о результатах обработки вышележащему программному обеспечению. НЯ используются для виртуализации аппаратного обеспечения реальных компьютеров или для реализации механизма гипервизора.

**5. Экзоядерная архитектура** предоставляет лишь набор сервисов для взаимодействия между приложениями, а также необходимый минимум функций, связанных с защитой: выделение и высвобождение ресурсов, контроль прав доступа и т. д. ЭЯ не занимается предоставлением абстракций для *физических ресурсов* – эти функции выносятся в библиотеку пользовательского уровня (так называемую libOS). В отличие от микроядра ОС, базирующиеся на ЭЯ, обеспечивают большую эффективность за счет отсутствия необходимости в переключении между процессами при каждом обращении к оборудованию.

**6. Гибридные ядра** (модифицированные микроядра, позволяющие для ускорения работы запускать "несущественные" части в пространстве ядра. Имеют "гибридные" достоинства и недостатки. Примером смешанного подхода может служить возможность запуска операционной системы с монолитным ядром под управлением микроядра. Так устроены 4.4BSD и MkLinux, основанные на микроядре *Mach*. Микроядро обеспечивает управление виртуальной памятью и работу низкоуровневых драйверов. Все остальные функции, в том числе взаимодействие с прикладными программами, осуществляются монолитным ядром. Данный подход сформировался в результате попыток использовать преимущества *микроядерной архитектуры*, сохраняя по возможности хорошо отлаженный код *монолитного ядра*.)

Вопрос: Классы операционных систем для ПЭВМ, получившие наибольшее распространение: ОС семейства Microsoft, ОС семейства IBM, ОС семейства Apple, ОС семейства UNIX.

1) Операционные системы семейства DOS

Первый представитель этого семейства - система MS DOS (Microsoft Disc Operating System - дисковая операционная система Microsoft) была выпущена в 1981 году в связи с появлением IBM PC. Операционные системы семейства DOS являются однозадачными и обладают следующими характерными чертами и особенностями:

- интерфейс с ЭВМ осуществляется с помощью команд, вводимых пользователем;

- модульность структуры, упрощает перенос системы на другие типы ЭВМ;

- небольшой объем доступной оперативной памяти (640Кбайт).

Существенным недостатком операционных систем семейства DOS является отсутствие средств защиты от несанкционированного доступа к ресурсам ПК и ОС. В настоящее время широкое распространение получила ОС MS DOS 6.22.

2) Операционные системы семейства OS/2

Операционная система OS/2 (Operating System/2) была разработана фирмой IBM в 1987 году в связи с созданием нового семейства ПЭВМ.

Она является 32-разрядной графической многозадачной операционной системой для IBM PC - совместимых компьютеров, позволяет организовать параллельную работу нескольких прикладных программ, обеспечивая при этом защиту одной программы от другой и операционной системы от работающих в ней программ. Для написания программ под OS/2 можно использовать уже готовые программные модули, которые содержатся в так называемом интерфейсе прикладного программирования - API (Application Programming Interface).

Операционная система OS/2 обладает удобным графическим пользовательским интерфейсом и совместима с файловой системой DOS, что дает возможность использовать данные как в DOS, так и в OS/2 без каких либо преобразований.

Имеется несколько модификаций OS/2:

- OS/2 Warp 3.0 - усовершенствовано использование памяти и улучшен графический интерфейс;

- OS/2 Warp Connect - улучшена поддержка сетей;

- OS/2 Warp Server - предназначена для работы в качестве серверной ОС.

Главный недостаток OS/2 - малое число приложений для нее, что делает эту систему менее популярной, чем операционные систе­мы MS DOS и Windows.

3) Операционные системы семейства UNIX

Операционные системы семейства UNIX - это 32-разрядные многозадачные многопользовательские операционные системы. Сильная сторона UNIX состоит в том, что одна и та же система ис­пользуется в различных компьютерах - от суперкомпьютера до ПЭВМ, что дает возможность переноса системы с одной машинной архитектуры на другую с минимальными затратами.

UNIX объединяет в себе: доступ к распределенным базам дан­ных, локальные сети, удаленную дистанционную связь и возмож­ность выхода в глобальные сети, используя обычный модем. Почто­вая служба UNIX - одна из важнейших ее компонент. В настоящее время существует большое количество приложений для UNIX. Большинство популярных приложений для DOS и Windows могут использоваться в UNIX.

Имеется несколько ОС семейства UNIX. Различные версии это­го семейства имеют свои названия, но в общих чертах повторяют особенности базовой ОС UNIX. Файловая система ОС UNIX обес­печивает защиту файлов от несанкционированного доступа на уров­нях пользователя и групп пользователей. В настоящее время из сете­вых ОС семейства UNIX широкое распространение получила ОС для сетей предприятий UnixWare 2.0 - 32-разрядная многопользовательская многозадачная ОС, поддерживающая приложения реально­го времени.

4) Операционные системы семейства Windows

Операционные системы семейства Windows разработаны фир­мой Microsoft. Они являются многозадачными операционными сис­темами, предоставляющими удобный графический интерфейс. Ос­новными представителями данного семейства являются ОС Windows 95, ОС Windows 98 и ОС Windows NT. Windows 95 разработана на базе ОС MS DOS и операционных оболочек Windows 3.x.

Windows 95 является частично 32-разрядной, частично 16-разрядной операционной системой.

Операционная система Windows NT - одна из наиболее распро­страненных 32-разрядных сетевых ОС. Windows NT выпускается в двух модификациях: Windows NT Server и Windows NT Workstation. Windows NT Server в первую очередь предназначен для управления сетевыми ресурсами. Система обеспечивает высокую мобильность и безопасность без потери производительности. Windows NT Server содержит средства для организации быстрого поиска информации и просмотра ресурсов глобальных сетей, обеспечивает возможность использование любых каналов связи (включая обычные телефонные линии), поддерживает до 256 одновременных подключений к одно­му серверу, а несколько серверов могут быть использованы для ор­ганизации общедоступной сетевой службы.

Windows NT Workstation - это версия ОС Windows NT, предна­значенная для работы на локальных компьютерах и рабочих станци­ях. Она является полностью 32-разрядной операционной системой, наиболее защищенной и надежной.

Все приложения в Windows NT работают в режиме многозадач­ности. Вместе с тем под Windows NT работают не все приложения MS DOS и 16-разрядные Windows -программы.

Windows NT Workstation целесообразно использовать, когда не­обходима надежная защита конфиденциальных данных или про­грамм, а также при выполнении инженерных, научных, статистиче­ских и других работ, когда важна высокая производительность при анализе больших объемов данных.