СПИСОК ОСНОВНОЙ ЛИТЕРАТУРЫ

1.Таненбаум Э. Современные операционные системы

2.Таненбаум Э., Вудхалл Операционные системы. Разработка и реализация.

3.Операционные системы. Конспект лекций. Intuit.ru

4.Современные операционные системы. Лекции + лабораторный практикум. Intuit.ru

5.Дейтел Г. Введение в операционные системы, в 2-х томах.

6.Макаров С.М. Конспект лекций по дисциплине «Операционные системы».

7.Копичко С.М., Макаров С.М. Системне програмне забезпечення

8.Dr.Tong Yu, April,2005 CS460 Operating Systems. Lecture Notes

1. Компоненты компьютерной системы. Архитектура компьютерной системы и общая картина ее функционирования.

1. **Аппаратура (hardware)** компьютера, основные части которой:

* **центральный процессор (CPU**), выполняющий **команды (инструкции)** компьютера;
* **память**, хранящая данные и программы;
* **устройства ввода-вывода, или внешние устройства (input-output devices**), обеспечивающие ввод информации в компьютер и вывод результатов работы программ в форме, воспринимаемой пользователем-человеком или другими программами.

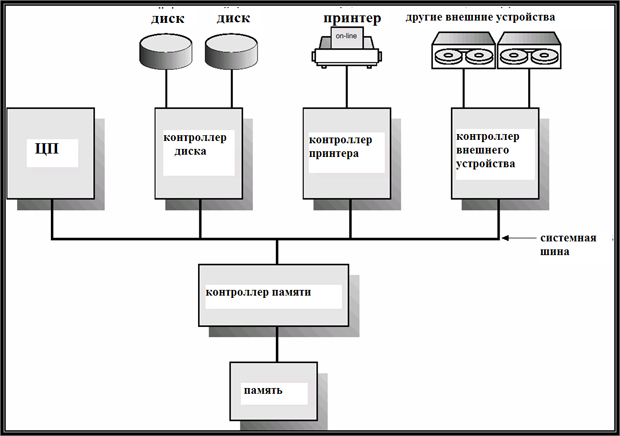
1. **ОС** – системное программное обеспечение, управляющее использованием аппаратуры компьютера различными программами и пользователями.
2. **Прикладное программное обеспечение (applications software) –** программы, предназначенные для решения различных классов задач.

* **компиляторы**, обеспечивающие трансляцию программ с языков программирования, в машинный **код** (команды);
* **СУБД**;
* **графические библиотеки**, **игровые программы**, **офисные программы**.

Прикладное программное обеспечение образует более высокий уровень, по сравнению с ОС, и позволяет решать на компьютере различные прикладные и повседневные задачи.

1. **Пользователи (users)** – люди и другие компьютеры. Другие компьютеры в сети также могут играть роль пользователей (**клиентов**) по отношению к данному компьютеру, выступающему в роли **сервера.**
2. **Центральный процессор** – устройство, выполняющее **команды (instructions)** компьютерной системы. В современных компьютерах, как правило, он является **многоядерным.** Производительность – **тактовая частота процессора (ядра)** – *время выполнения* им одной самой простой машинной команды.
3. **Оперативная (основная) память**, или просто **память** – устройство, хранящее обрабатываемые данные.
4. **Системная шина** – устройство, к которому подсоединены все модули компьютера и через которое они обмениваются сигналами, например, о прерываниях.
5. **Порты** – устройства с разъемами для подключения к компьютеру *внешних устройств*.

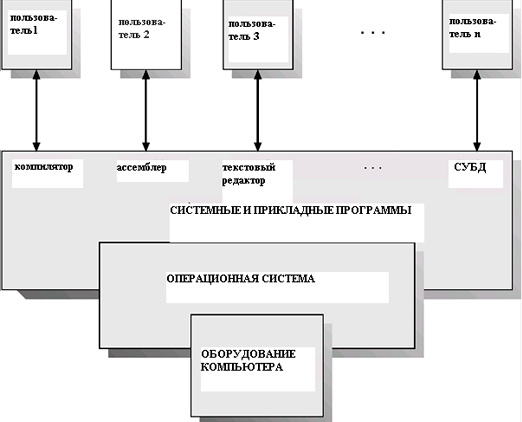
**Архитектура КС**

[](http://www.intuit.ru/EDI/18_03_15_2/1426630694-20867/tutorial/901/objects/4/files/4_1.png)

Компьютерная система имеет **модульную** структуру. Для каждого устройства в системе имеется специальное *устройство управления* – **контроллером устройства**.

Все модули соединены между собой **системной шиной,** через которую они обмениваются сигналами. Работой каждого контроллера управляет **драйвер –** специализированная низкоуровневая *программа*, являющаяся частью ОС.

**Общая картина функционирования компьютерной системы**

[](http://www.intuit.ru/EDI/18_03_15_2/1426630694-20867/tutorial/901/objects/1/files/1_1.png)

2. Классификация компьютерных систем, их характеристики и особенности.

1. **Суперкомпьютеры –** мощные *многопроцессорные* компьютеры.
2. **Многоцелевые компьютеры**, или **компьютеры общего назначения (mainframes) –** громоздкие компьютеры с низкими характеристиками, которые были популярны в ХХ веке, но в них реализованы все основные методы и алгоритмы, которые были использованы в ОС для персональных, карманных компьютеров и **мобильных устройств**.
3. **Кластеры компьютеров** – группы компьютеров, физически расположенные рядом и соединенные друг с другом высокоскоростными шинами и линиями связи, используются для высокопроизводительных **параллельных вычислений**.
4. **Настольные компьютеры** – это наиболее распространенные в настоящее время компьютеры, состоят из монитора, системного блока, клавиатуры и мыши.
5. **Портативные компьютеры -** миниатюрные компьютеры, *по* своим параметрам не уступающие настольным, хорошо приспособлены для ввода, обработки и воспроизведения обработки мультимедийной информации.
6. **Карманные портативные компьютеры и органайзеры -** миниатюрные компьютеры, помещающиеся на ладони или в кармане, но *по* своему быстродействию иногда не уступающего ноутбуку.
7. **Мобильные устройства -** устройства для голосовой связи, реже – для записи или обработки информации или для выхода в *Интернет.* ОС для *мобильных устройств* отличаются большей компактностью, ввиду более жестких ограничений *по* памяти.
8. **Носимые компьютеры -** *память* и *быстродействие* значительно меньше, чем у настольных компьютеров, но критическим фактором является их сверхвысокая *надежность*, а для их ОС – минимальное возможное **время ответа** – *интервал*, в течение которого система обрабатывает информацию от датчиков, от пользователя или из сети. ОС для **носимых компьютеров** можно отнести к **системам реального времени**.
9. **Распределенные системы –** системы, состоящие из нескольких компьютеров, объединенных в проводную или беспроводную *сеть*.
10. **Системы реального времени -** вычислительные системы, предназначенные для управления различными техническими, военными и другими объектами в режиме реального времени. Характеризуются основным требованием к аппаратуре и программному обеспечению, в том числе к операционной системе: **недопустимость превышения времени ответа** системы.

3. Программное обеспечение компьютерной системы, его классификация и состав.

**ПО** – совокупность программ, стандартных программ (подпрограмм), языков программирования, правил и документации, которая необходима для использования и эксплуатации программных продуктов.

**Классификация:**

1. **Прикладное ПО** – совокупность программ для решения определенных целевых задач или класса таких задач. Например:

* базы данных;
* информационно-поисковые системы;
* электронные таблицы;
* системы программированного обучения;
* системы искусственного интеллекта;
* математические программы;
* программы для моделирования;
* системы автоматизированного программирования.

1. **Системное ПО** – комплекс программных средств для повышения эффективности использования мощностей ЭВМ, упрощения эксплуатации, снижения трудоемкости роботы с проектированием и выполнения программных продуктов, предоставление пользователю ЭВМ разнообразных услуг.

* ***ОС***— обязательная компонента ЭВМ, которая организует выполнение программ и взаимодействие пользователя с компьютером;
* **Сервисные** – расширение возможностей ОС;
* **Инструментальные** – объединение СПО для разработки программных продуктов и решения прикладных задач;
* **Тех. обслуживание** – упрощение тестирования оборудования и поиска неисправностей.

4. ОС как часть системного программного обеспечения компьютерной системы, ее место и роль (функции) в составе компьютерной системы. Подсистемы (компоненты) ОС.

ОС работает непосредственно с аппаратным обеспечением и является основой всего остального программного обеспечения.

***Роль ОС:***

* организация интерфейса между прикладной программой и ОС при помощи системных вызовов;
* организация очереди из заданий в памяти – выделение процессора одному из заданий потребовало планирования использования процессора;
* переключение с одного задания на другое;
* организация хранения информации на внешних носителях в виде файлов.

**Две основные функции ОС. Временное и пространственное распределение ресурсов .**

**Первичная задача ОС** – отслеживание какой программой какой ресурс используется, чтобы удовлетворять запросы на использование ресурсов, нести ответственность за их использование и принимать решение по конфликтующим запросам от различных программ и пользователей.

Управление ресурсами включает в себя *мультиплексирование* (распределение) ресурсов двумя способами:

* во времени;
* в пространстве.



5. Система программирования компьютерной системы как часть системного программного обеспечения компьютер ной системы, ее назначение, состав и место в структуре компьютерной системы.

**Система программирования** – совокупность языков программирования и комплексу программ, которые обеспечивают реализацию данного языка программирования средствами машинного языка конкретной ЭВМ.

* **Транслятор** – программа для перевода описаний алгоритмов из входящего языка в выходящий. Чаще всего используется машинный язык конкретной ЭВМ.
* **Компилятор** – программа, которая переводит текст программы на языке высокого уровня в эквивалентную программу на машинном языке.
* **Компоновщик –** программа, которая производит компоновку – принимает на вход один или несколько объектных модулей и собирает по ним осуществленный модуль.
* **Препроцессоры выходящих текстов –** программы, которые берут данные на входе и выдают данные, предназначенные для входа другой программы.
* **Отладчик –** модуль среды разработки или отдельное приложение для поиска ошибок в программе.
* **Текстовые редакторы –** программы для создания и изменения текстовых файлов, а также просмотра их на экране, вывода на печать, поиска фрагментов текста и т.д.
* **Специализированные редакторы выходящих текстов –** текстовые редакторы для создания и редактирования выходящего кода программ.
* **Библиотеки подпрограмм –** сборники подпрограмм или объектов, которые используются для разработки ПО.

6. Ресурсы компьютерной системы. Временное и пространственное распределения ресурсов.

Управление ресурсами включает в себя **мультиплексирование** (распределение) ресурсов двумя различными способами: **во времени и в пространстве.**

Когда ресурс разделяется **во времени**, различные программы или пользователи используют его **по очереди**: сначала ресурс получают в пользование одни, потом другие и т. д. Определение того, **как именно** ресурс будет разделяться во времени — кто будет следующим потребителем и как долго, — это **задача ОС**.

Другим видом разделения ресурсов является **пространственное разделение**. Вместо поочередной работы каждый клиент получает какую-то **часть разделяемого ресурса**. При этом возникают проблемы равной **доступности, обеспечения безопасности** и т. д., и их должна решать ОС.

**Распределение** дискового пространства и отслеживание того, кто какие дисковые блоки использует, — это типичная задача ОС по управлению ресурсами.

7. Режимы работы компьютеров. Поколения (эволюция, "зоопарк") ОС.

1. ОС 50-х годов – цель ускорения и упрощения перехода с задачи на задачу. Появились ***системы пакетной обработки***. Запущенная в решение задача получала в свое полное распоряжение все ресурсы машины.

2. Системы коллективного пользования с ***мультипрограммным***режимом работы (мультипроцессорный тип).

3. **Машины общего назначения**. ОС третьего поколения – многорежимные системы. Работают во всех режимах: пакетная обработка, разделение времени, режим реального времени и мультипроцессорный режим.

4. Появились **вычислительные сети**, средства оперативной обработки данных, **микропроцессоры**.

**Пирамида или "зоопарк" ОС**

1. **Мэйнфреймы** – большие машины, ориентированы на огромное количество операций ввода-вывода:

* пакетная обработка;
* обработка транзакций;
* разделение времени.

2. **Серверные ОС** – одновременно обслуживают много пользователей.

3**. Многопроцессорные ОС** – соединение процессоров в один:

* параллельные компьютеры;
* мультикомпьютеры;
* многопроцессорные системы.

4. ОС для **персональных компьютеров**: Windows, Macintosh, Linux.

5. ОС **реального времени** (главный параметр – время, собирают данные о промышленных процессе и управляют машинами на фабрике).

6. **Встроенные ОС** (в карманных компьютерах, небольшой набор функций).

7. ОС для **смарт-карт** (маленькая мощность и память, могут управлять только одной операцией).

8. Основные понятия ОС, их суть.

Ключевым понятием во всех ОС – **процесс** – программа во время ее выполнения. С каждым процессом связано его **адресное пространство** — список адресов ячеек памяти от нуля до некоторого максимума, откуда процесс может считывать данные и куда может записывать их.

**Взаимоблокировка** – оба процесса оказываются заблокированными навсегда.

Другим ключевым понятием, поддерживаемым практически всеми операционными системами, является **файловая система**. Чтобы предоставить место для хранения файлов, многие ОС ПК используют **каталог,** как способ объединения файлов в группы.

**Память** представляет собой очень важный ресурс, требующий четкого управления. Часть ОС, которая управляет иерархией памяти (или ее частью) – **менеджер** **памяти**. Он предназначен для действенного управления памятью и должен следить за тем, какие части памяти используются, выделять память процессам, которые в ней нуждаются, и освобождать память, когда процессы завершат свою работу.

У всех компьютеров имеются физические устройства для получения **входной и вывода** выходной информации. Поэтому у каждой ОС для управления такими устройствами существует своя подсистема ввода-вывода.

Управление **безопасностью** системы также возлагается на ОС: например, она должна обеспечить доступ к файлам только пользователям, имеющим на это право.

**Оболочка** - средство доступа ко многим функциям ОС и служит хорошим примером использования системных вызовов.

9. Структура ОС.

1. **МОНОЛИТНЫЕ СИСТЕМЫ.** Такая организация ОС является самой распространенной. Вся ОС работает как единая программа в режиме ядра. ОС написана в виде набора процедур, связанных вместе в одну большую исполняемую программу.

Такая организация предполагает следующую базовую структуру операционной системы:

* Основная программа, которая вызывает требуемую служебную процедуру.
* Набор служебных процедур, выполняющих системные вызовы.
* Набор вспомогательных процедур, содействующих работе служебных процедур.

1. **МНОГОУРОВНЕВЫЕ СИСТЕМЫ.** Организация ОС в виде иерархии уровней, каждый из которых является надстройкой над нижележащим уровнем.
2. **КЛИЕНТ-СЕРВЕРНАЯ МОДЕЛЬ.** Небольшая вариация идеи микроядер выражается в обособлении двух классов процессов: **серверов**, каждый из которых предоставляет какую-нибудь службу, и **клиентов**, которые пользуются этими службами.

Суть заключается в наличии клиентских процессов и серверных процессов.

1. **ВИРТУАЛЬНЫЕ МАШИНЫ.** Виртуализация рассматривается как способ запуска всех серверов на одной и той же машине с возможностью избежать при этом отказа всех серверов при отказе одного из них.
2. **ЭКЗОЯДРА.** Задача состоит в распределении ресурсов между виртуальными машинами и отслеживании попыток их использования, чтобы ни одна из машин не пыталась использовать чужие ресурсы.

10. Режимы функционирования программного обеспечения (режим ядра, пользовательский режим), их суть и использование. Привилегированные команды.

ОС работает в ***режиме ядра***(супервизора). В этом режиме она имеет полный доступ ко всему аппаратному обеспечению и может задействовать любую инструкцию, которую машина в состоянии выполнить.

Вся остальная часть ПО работает *в* ***режиме пользователя****,* в котором доступно лишь подмножество инструкций, управляющих машиной или осуществляющих операции ввода-вывода.

Можно предоставлять доступ по принципу **минимума привилегий**: каждому конкретному пользователю следует предоставлять минимальный возможный приоритет и право доступа только к тем ресурсам, которые ему действительно необходимы.

В процессе развития компьютерной архитектуры выявилась тенденция к увеличению количества привилегированных команд.

11. Системные вызовы, их назначение и типы.

ОС выполняют **две основные функции**: предоставляют абстракции пользовательским программам и управляют ресурсами компьютера.

В основном **взаимодействие пользовательских программ** и ОС касается первой функции — взять, к примеру, операции с файлами: создание, запись, чтение и удаление.

А **управление ресурсами** компьютера проходит большей частью незаметно для пользователей и осуществляется в автоматическом режиме. Так что интерфейс между пользовательскими программами и ОС строится в основном на абстракциях. Чтобы понять, что делает ОС, нужно более подробно рассмотреть этот интерфейс.

Имеющиеся в интерфейсе **системные вызовы** варьируются в зависимости от используемой операционной системы (хотя основные понятия практически ничем не различаются).

12. Прерывания, типы прерываний, механизм прерываний (последовательность событий).

**Прерывания** дают возможность одному устройству немедленно привлечь внимание другого устройства с тем, чтобы первое могло сообщить об изменении своего состояния.

Если произошло прерывание, то:

1) управление передается ОС;

2) ОС запоминает состояние прерванного процесса;

3) ОС анализирует тип прерывания и передает управление соответствующей

программе обработки этого прерывания.

**Инициатор** – выполняющийся процесс или некоторое событие, связанное или несвязанное с этим процессом.

**Типы прерываний:**

* *SVC-прерывания* (инициатор – работающий процесс, который выполняет команду SVC).

**Команда SVC** – это генерируемый программой пользователя запрос на

предоставление конкретной системы услуги. Механизм SVC защищает ОС от

пользователя;

* *прерывания ввода-вывода* (инициатор – аппаратура ввода-вывода; сигнализирует ЦП о том, что произошло изменение состояния канала или устройств);
* *внешние прерывания* (причина – различные события, в том числе истечение кванта времени, заданного на таймере прерываний, нажатие оператором клавиши прерываний на пульте прерывания или прием сигнала прерывания от другого процессора в мультипроцессорной системе);
* *прерывания по рестарту* (когда оператор нажимает на пульте управления кнопку рестарта или когда от другого процессора в мультипроцессорной системе поступает команда рестарта SIGP (сигнал процессору);
* *прерывания по контролю (ошибке) программы* (вызываются различными видами ошибок, обнаруженными в выполняющемся процессе).

13. Процессоры компьютерных систем, их параметры, архитектура.

**Центральный процессор** — это «мозг» компьютера. Он выбирает команды из памяти и выполняет их. Обычный цикл работы центрального процессора выглядит так: выборка из памяти первой команды, ее декодирование для определения ее типа и операндов, выполнение этой команды, а затем выборка, декодирование и выполнение последующих команд. Этот цикл повторяется до тех пор, пока не закончится программа. Таким образом программы выполняются.

Для каждого типа центрального процессора существует **определенный набор команд**, которые он может выполнять. Поэтому x86 не может выполнять программы, написанные для **ARM-процессоров**, а те, в свою очередь, не в состоянии выполнять программы, написанные для x86.

Поскольку **доступ к памяти** для получения команды или данных занимает намного больше времени, чем выполнение команды, у всех центральных процессоров есть несколько **собственных регистров** для хранения основных переменных и промежуточных результатов.

Соответственно набор команд содержит, как правило, **команды на загрузку слова** из памяти в регистр и на запоминание слова из регистра в память. Другие команды объединяют два операнда из регистров, памяти или обоих этих мест для получения результата — например, складывают два слова и сохраняют результат в регистре или в памяти.

14. Память, основные параметры, иерархия. Аппаратная защита памяти и процессора. Управление перемещением программ.

**Система памяти** создается в виде **иерархии уровней**. Верхние уровни обладают более высоким быстродействием, меньшим объемом и более высокой удельной стоимостью хранения одного бита информации, чем нижние уровни, иногда в миллиарды и более раз.

В целях совместного использования ***системных ресурсов*** несколькими программами, требуется, чтобы **аппаратура и** ***ОС*** обеспечили невозможность влияния некорректно исполняемой программы на другие пользовательские программы.

Для этого необходима **аппаратная*****поддержка***, как *минимум*, двух режимов исполнения программ:

* **пользовательский (непривилегированный) (user mode) –** выполнение программ пользователей;
* **системный (привилегированный**, **режим ядра system mode**, **monitor mode) –** для модулей ОС.

Идея двух режимов в том, чтобы выполняемые в привилегированном режиме модули ОС могли выполнять **распределение и выделение*****системных ресурсов***, в частности, формировать новые адреса, а пользовательские программы, в результате ошибок или преднамеренных атак, выполняясь в обычном режиме, не могли бы обратиться в область памяти ОС или другой задачи, изменять их и этим нарушать их ***целостность***.

Для **защиты памяти**необходимо обеспечить защиту, *по* крайней мере, для вектора прерываний и подпрограмм обслуживания прерываний.

15. Устройства ввода-вывода (контроллеры, драйверы и их взаимодействие с ядром ОС, способы ввода-вывода данных).

**Устройства ввода-вывода** обычно состоят из двух компонентов: **самого устройства и его контроллера**. Контроллер представляет собой микросхему или набор микросхем, которые управляют устройством на **физическом уровне**. Он принимает от ОС команды, а затем их выполняет.

Программа, предназначенная для общения с контроллером, выдачи ему команды и получения, поступающих от него ответов, называется **драйвером** устройства. Для использования драйвер нужно поместить в ОС, предоставив ему тем самым возможность работать в **режиме ядра.**

**Ввод и вывод данных** можно делать тремя различными способами. В простейшем из них пользовательская программа производит системный вызов, который транслируется ядром в процедуру вызова соответствующего драйвера. Этот способ называется **активным ожиданием** или **ожиданием готовности**, а его недостаток заключается в том, что он загружает процессор опросом устройства об окончании работы.

**Второй способ** заключается в том, что драйвер запускает устройство и просит его **выдать прерывание** по окончании выполнения команды (завершении ввода или вывода данных). Когда контроллер обнаруживает окончание передачи данных, он генерирует **прерывание**, чтобы просигнализировать о завершении операции.

**При третьем способе** ввода-вывода используется специальный контроллер **прямого доступа к памяти** (Direct Memory Access (**DMA**)), который может управлять потоком битов между оперативной памятью и некоторыми контроллерами без постоянного вмешательства центрального процессора.

16. Шины и порты, их назначение и особенности использования.

**Архитектура шин совместного использования** означает, что для передачи данных разными устройствами используются одни и те же проводники. Таким образом, когда данные для передачи имеются сразу у нескольких устройств, для определения устройства, которому будет позволено использовать шину, требуется **арбитр.**

**Архитектура параллельной шины**, подобная той, что используется в PCI, предполагает отправку каждого слова данных по нескольким проводникам.

**Шина USB** была разработана для подключения к компьютеру всех низкоскоростных устройств ввода-вывода вроде клавиатуры и мыши. USB является **централизованной шиной**, в которой главное (корневое) устройство опрашивает устройства ввода-вывода каждую миллисекунду, чтобы узнать, есть ли у них данные для передачи.

**SCSI** (Small Computer System Interface — интерфейс малых вычислительных систем) является высокоскоростной шиной, предназначенной **для высокопроизводительных дисков**, сканеров и других устройств, нуждающихся в значительной пропускной способности.

17. Классификация компьютерных архитектур.

Компьютерные системы отличаются между собой не только *по* своим параметрам и своему назначению, но и *по* своим внутренним архитектурным принципам. Наиболее известны следующие подходы к архитектуре компьютерных систем.

**CISC (Complicated Instruction Set Computers – компьютеры с усложненной системой команд) –** исторически первый подход к компьютерной архитектуре, суть которого в том, что в систему команд компьютера включаются сложные *по* семантике *операции*, реализующие типовые действия, часто используемые при программировании и при *реализации языков* – например, вызов рекурсивных процедур и автоматическое обновление *дисплей*-регистров, групповые *операции* пересылки строк и массивов и др.

**RISC (Reduced Instruction Set Computers – компьютеры с упрощенной системой команд) –** упрощенный подход к*архитектуре компьютеров.* **Принципы данного подхода**:

* упрощение семантики команд, отсутствие сложных групповых операций; одинаковая *длина* команд (32 бита – *архитектура* была разработана в расчете на 32-битовые процессоры);
* выполнение арифметических операций только в регистрах и использование специальных команд считывания из памяти в *регистр* и записи из регистра в *память*;
* отсутствие специализированных регистров (например, *дисплей*-регистров для адресации доступных областей локальных данных в стеке).

**VLIW (Very Long Instruction Word – компьютеры с широким командным словом) –** подход к *архитектуре компьютеров.* Основная идея данного подхода – **статическое планирование параллельных вычислений компилятором** на уровне отдельных последовательностей команд и подкоманд.

**EPIC (Explicit Parallelism Instruction Computers – компьютеры с явным распараллеливанием) –** *по* архитектуре аналогичны **VLIW**, но с добавлением ряда важных усовершенствований: например, **спекулятивных** вычислений – параллельного выполнения обеих веток условной конструкции с вычислением условия.

**Multi-core computers (многоядерные компьютеры) –** получившая наиболее широкую популярность в настоящее время *архитектура* компьютеров, при которой каждый *процессор* имеет несколько **ядер (cores)**, объединенных в одном кристалле и параллельно работающих на одной и той же общей памяти, что дает широкие возможности для *параллельных вычислений*.

**Hybrid processor computers (компьютеры с гибридными процессорами)** – новый, все шире распространяющийся подход к *архитектуре компьютеров*, при котором *процессор* имеет **гибридную** структуру – состоит из (**многоядерного**) **центрального процессора (CPU)** и (также **многоядерного**) **графического процессора (GPU – Graphical Processor Unit**).