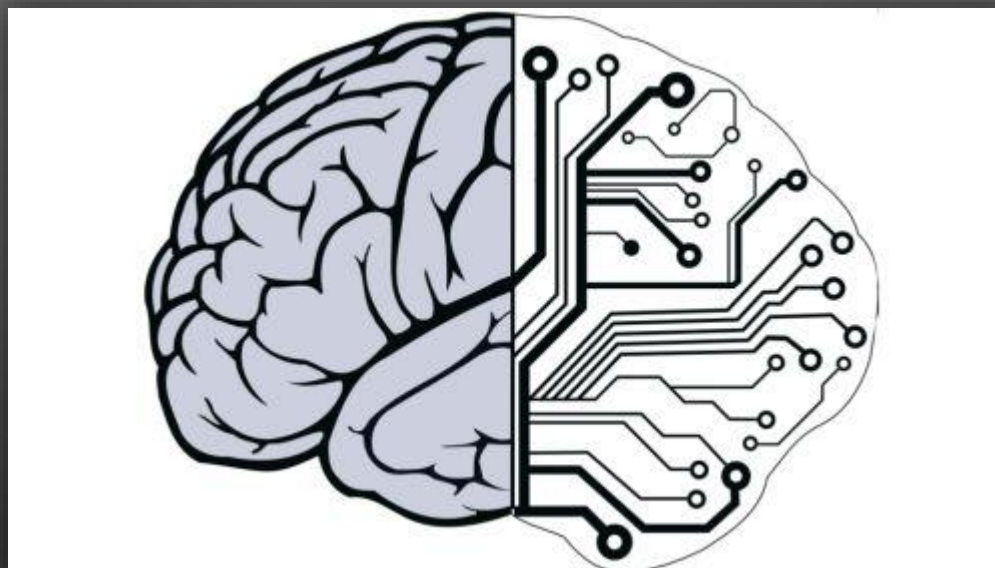


АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И ОПЕРАЦИОННЫЕ СИСТЕМЫ

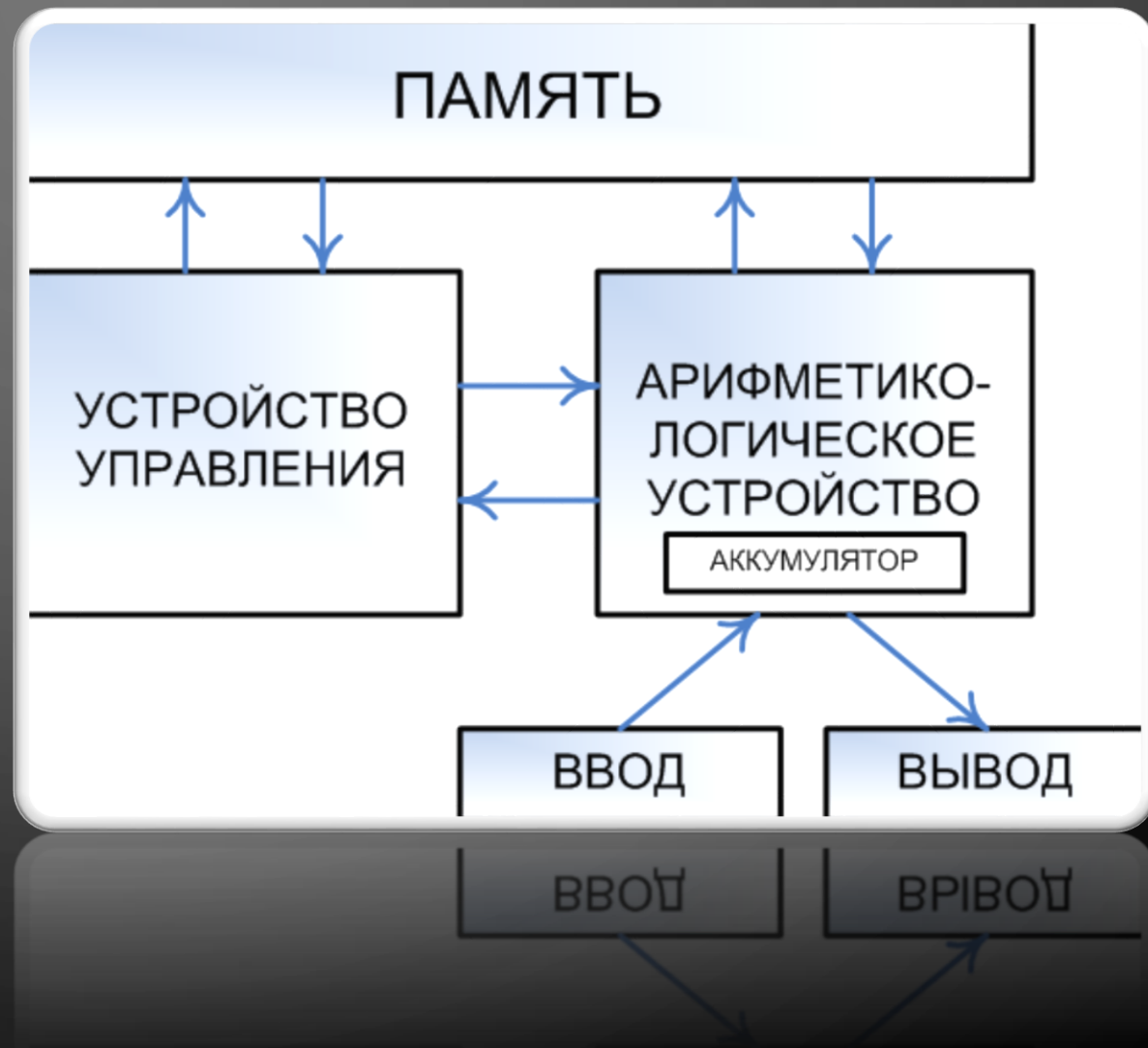


Выполнил студент:
Факультет ИТ
БПЗ1501
Тодика Арсений А.

АРХИТЕКТУРА ФОН-НЕЙМАНА

Архитектура Фон-Неймана – это широко известный принцип совместного хранения команд и данных в памяти компьютера.

Вычислительные системы такого рода часто обозначают термином «Машина Фон-Неймана», однако соответствие этих понятий не всегда однозначно. В общем случае, когда говорят об архитектуре Фон-Неймана, подразумевают принцип хранения данных и инструкций в одной памяти.



ПРИНЦИПЫ ФОН-НЕЙМАНА



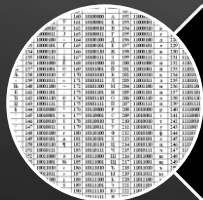
**Принцип однородности
памяти**



Принцип адресности



**Принцип программного
управления**



**Принцип двоичного
кодирования**

ПРИНЦИП ОДНОРОДНОСТИ ПАМЯТИ

Команды и данные хранятся в одной и той же памяти и внешне в памяти неразличимы. Распознать их можно только по способу использования; то есть одно и то же значение в ячейке памяти может использоваться и как данные, и как команда, и как адрес в зависимости лишь от способа обращения к нему. Это позволяет производить над командами те же операции, что и над числами, и, соответственно, открывает ряд возможностей. Так, циклически изменяя адресную часть команды, можно обеспечить обращение к последовательным элементам массива данных. Такой прием носит название модификации команд и с позиций современного программирования не приветствуется. Более полезным является другое следствие принципа однородности, когда команды одной программы могут быть получены как результат исполнения другой программы. Эта возможность лежит в основе трансляции — перевода текста программы с языка высокого уровня на язык конкретной вычислительной машины.

ПРИНЦИП АДРЕСНОСТИ

Структурно основная память состоит из пронумерованных ячеек, причём процессору в произвольный момент доступна любая ячейка. Двоичные коды команд и данных разделяются на единицы информации, называемые словами, и хранятся в ячейках памяти, а для доступа к ним используются номера соответствующих ячеек — адреса.

ПРИНЦИП ПРОГРАММНОГО УПРАВЛЕНИЯ

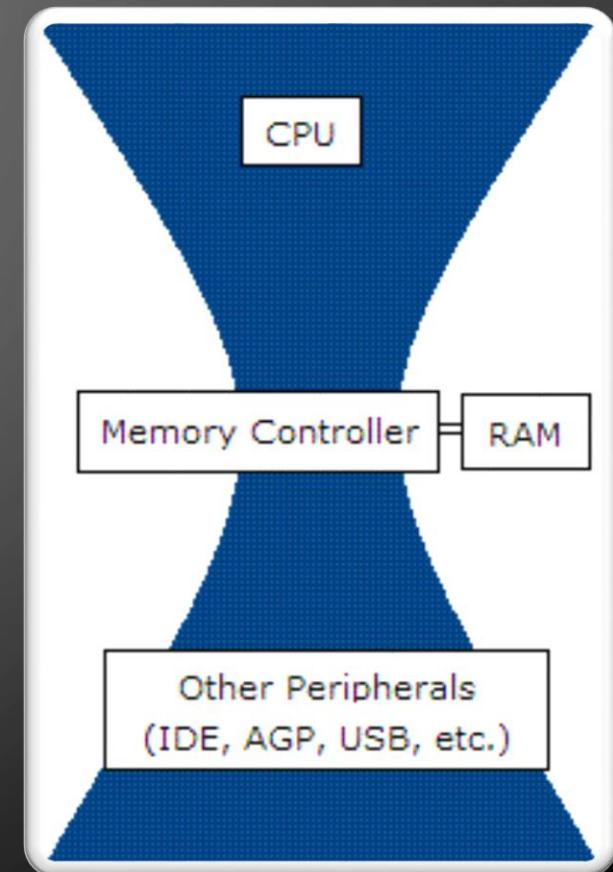
Все вычисления, предусмотренные алгоритмом решения задачи, должны быть представлены в виде программы, состоящей из последовательности управляющих слов — команд. Каждая команда предписывает некоторую операцию из набора операций, реализуемых вычислительной машиной. Команды программы хранятся в последовательных ячейках памяти вычислительной машины и выполняются в естественной последовательности, то есть в порядке их положения в программе. При необходимости, с помощью специальных команд, эта последовательность может быть изменена. Решение об изменении порядка выполнения команд программы принимается либо на основании анализа результатов предшествующих вычислений, либо безусловно.

ПРИНЦИП ДВОИЧНОГО КОДИРОВАНИЯ

Согласно этому принципу, вся информация, как данные, так и команды, кодируются двоичными цифрами 0 и 1. Каждый тип информации представляется двоичной последовательностью и имеет свой формат. Последовательность битов в формате, имеющая определённый смысл, называется полем. В числовой информации обычно выделяют поле знака и поле значащих разрядов. В формате команды в простейшем случае можно выделить два поля: поле кода операции и поле адресов.

УЗКОЕ МЕСТО АРХИТЕКТУРЫ ФОН-НЕЙМАНА

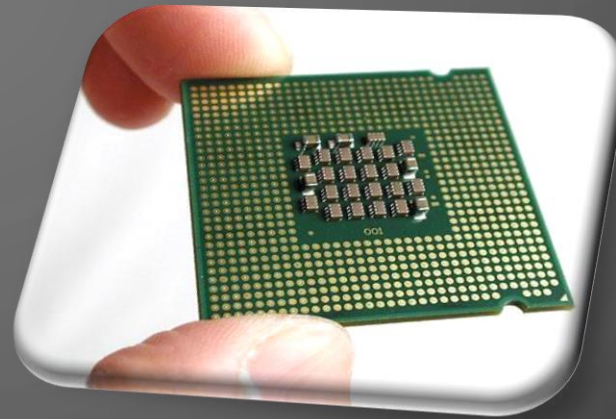
- Совместное использование шины для памяти программ и памяти данных приводит к узкому месту архитектуры фон Неймана, а именно ограничению пропускной способности между процессором и памятью по сравнению с объёмом памяти. Из-за того, что память программ и память данных не могут быть доступны в одно и то же время, пропускная способность канала «процессор-память» и скорость работы памяти существенно ограничивают скорость работы процессора — гораздо сильнее, чем если бы программы и данные хранились в разных местах. Так как скорость процессора и объём памяти увеличивались гораздо быстрее, чем пропускная способность между ними, узкое место стало большой проблемой, серьёзность которой возрастает с каждым новым поколением процессоров.
- Данная проблема решается совершенствованием систем кэширования, что в свою очередь усложняет архитектуру систем и увеличивает риск возникновения побочных ошибок (например, в 2017 году были обнаружены уязвимости Meltdown (уязвимость) и Spectre (уязвимость), присутствовавшие в современных процессорах в течение десятилетий, но не обнаруженные ранее из-за сложности современных вычислительных систем и, в частности, их взаимодействия с кэш-памятью).
- Термин «узкое место архитектуры фон Неймана» ввёл Джон Бэкус в 1977 в своей лекции «Можно ли освободить программирование от стиля фон Неймана?», которую он прочитал при вручении ему Премии Тьюринга.
- Учёные из США и Италии в 2015 заявили о создании прототипа мем-процессора (англ. memprocessor) с отличной от фон-неймановской архитектурой и возможности его использования для решения NP-полных задач.



(IDE, AGP, USB, etc.)
Other Peripherals

УСТРОЙСТВО ЦЕНТРАЛЬНОГО ПРОЦЕССОРА

Центральный процессор — это мозг компьютера. Его задача — выполнять программы, находящиеся в основной памяти. Для этого он вызывает команды из памяти, определяет их тип, а затем выполняет одну за другой.



ПРОЦЕССОР СОСТОИТ
ИЗ НЕСКОЛЬКИХ
ЧАСТЕЙ:

Блок управления

Арифметико-логическое устройство

быстрая память небольшого объема

Регистровая память

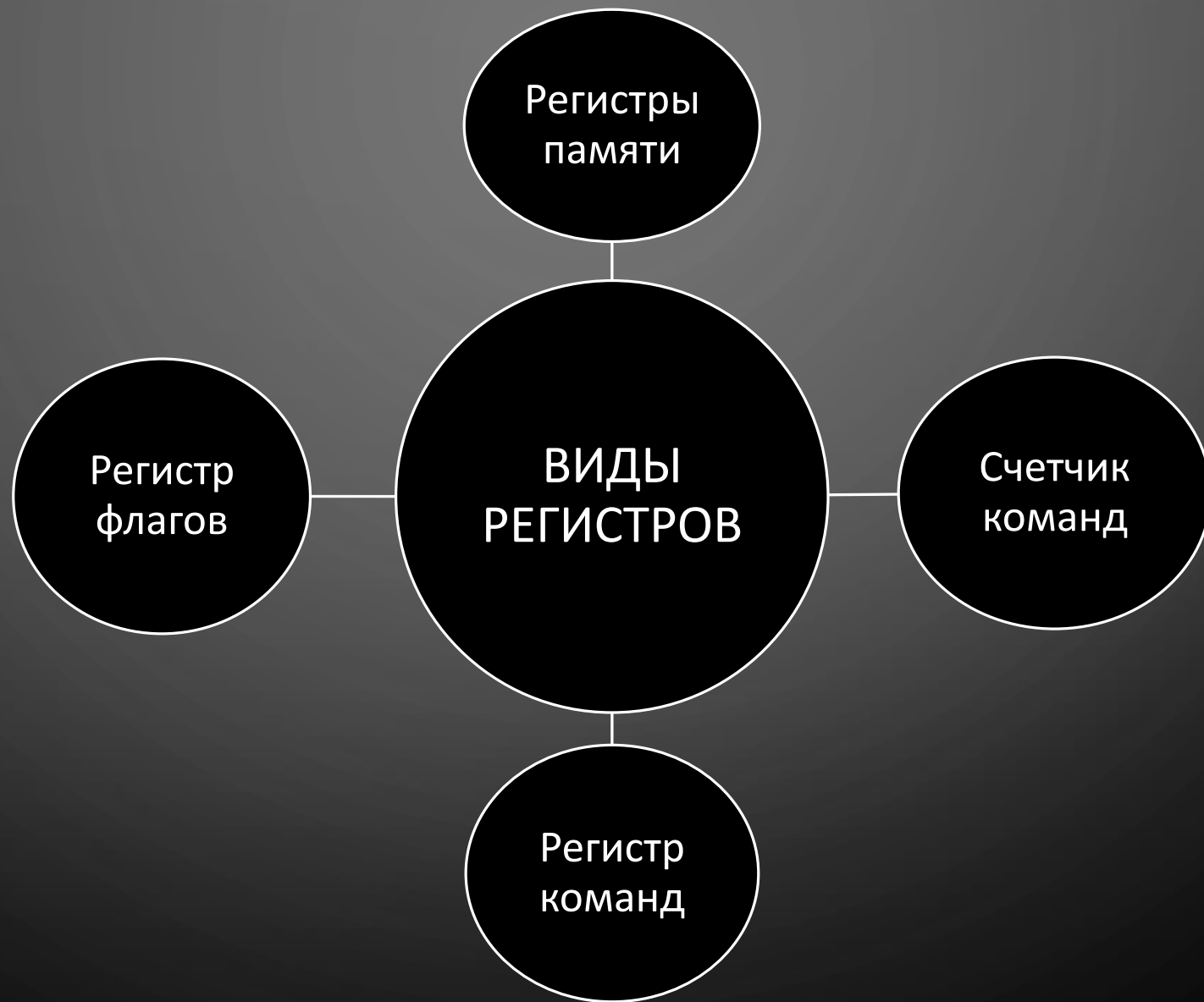
УСТРОЙСТВО ЦЕНТРАЛЬНОГО ПРОЦЕССОРА

Процессор состоит из нескольких частей:

- Блок управления отвечает за вызов команд из памяти и определение их типа.
- Арифметико-логическое устройство выполняет арифметические операции (например, сложение) и логические операции (например, логическое И).
- Внутри центрального процессора находится быстрая память небольшого объема для хранения промежуточных результатов и некоторых команд управления. Эта память состоит из нескольких регистров, каждый из которых выполняет определенную функцию.
- Регистровая память - вид компьютерной оперативной памяти, модули которой содержат регистр между микросхемами памяти и системным контроллером памяти.



ВИДЫ РЕГИСТРОВ

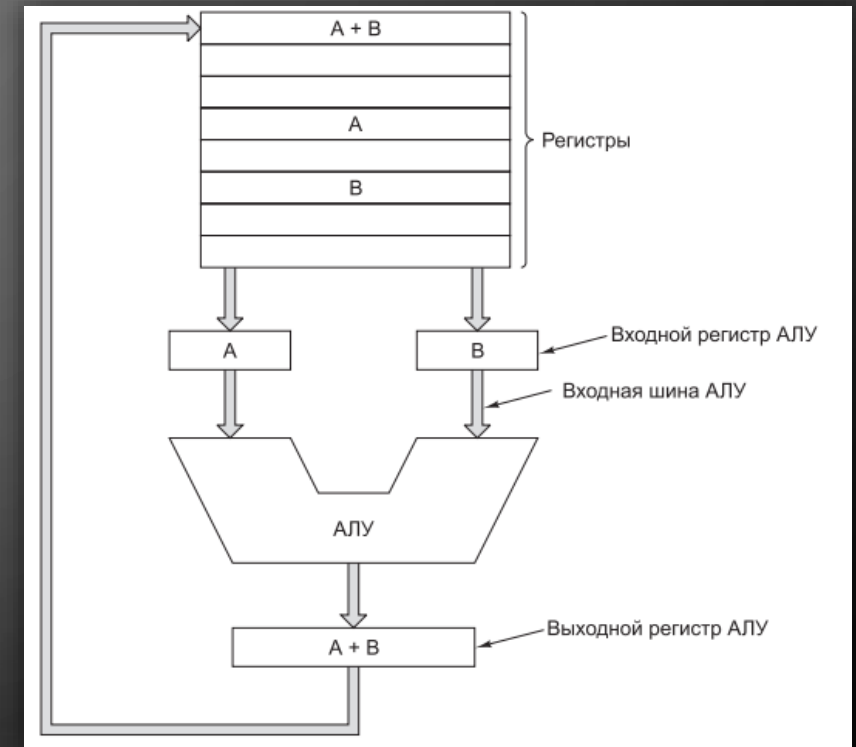


ВИДЫ РЕГИСТРОВ

- Регистры памяти. Оперативная память компьютера конструируется в виде набора регистров памяти, которые служат только для хранения информации. Один регистр образует одну ячейку памяти, которая имеет свой адрес. Если в регистр входит n триггеров, то можно запомнить n бит информации.
- Счетчик команд – регистр устройства управления (УУ) процессора, хранит адрес выполняемой в данный момент команды, по которому она находится в оперативной памяти.
- Регистр команд – регистр УУ, служит для вычисления адреса ячейки, где хранятся данные, требуемые выполняемой в данный момент программе.
- Регистр флагов – регистр УУ, хранит информацию о последней команде, выполненной процессором.

ТРАКТ ДАННЫХ ОБЫЧНОЙ ФОН-НЕЙМАНОВСКОЙ МАШИНЫ

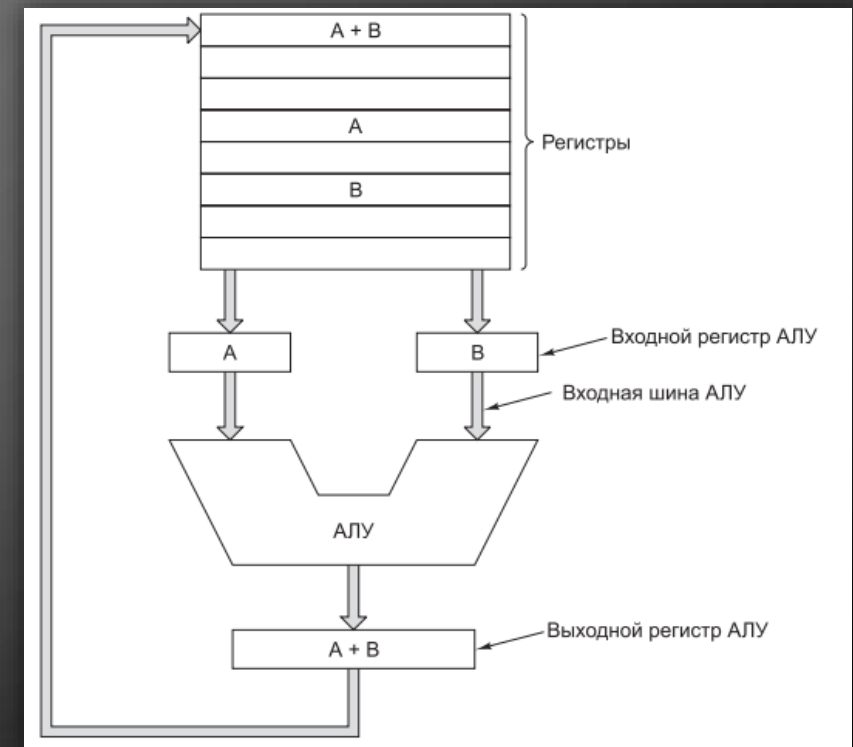
- Внутреннее устройство тракта данных типичного фон-неймановского процессора иллюстрирует данный рисунок.
- Тракт данных состоит из регистров (обычно от 1 до 32), арифметико-логического устройства (АЛУ) и нескольких соединительных шин. Содержимое регистров поступает во входные регистры АЛУ, которые на рисунке обозначены буквами А и В. В них находятся входные данные АЛУ, пока АЛУ производит вычисления. АЛУ выполняет сложение, вычитание и другие простые операции над входными данными и помещает результат в выходной регистр. Содержимое этого выходного регистра может записываться обратно в один из регистров или сохраняться в памяти, если это необходимо.
- Большинство команд можно разделить на две группы: команды типа регистр-память и типа регистр-регистр. Команды первого типа вызывают слова из памяти, помещают их в регистры, где они используются в качестве входных данных АЛУ. Другие команды этого типа помещают регистры обратно в память. Команды второго типа вызывают два операнда из регистров, помещают их во входные регистры АЛУ, выполняют над ними какую-нибудь арифметическую или логическую операцию и переносят результат обратно в один из регистров. Этот процесс называется циклом тракта данных.



ТРАКТ ДАННЫХ ОБЫЧНОЙ ФОН- НЕЙМАНОВСКОЙ МАШИНЫ

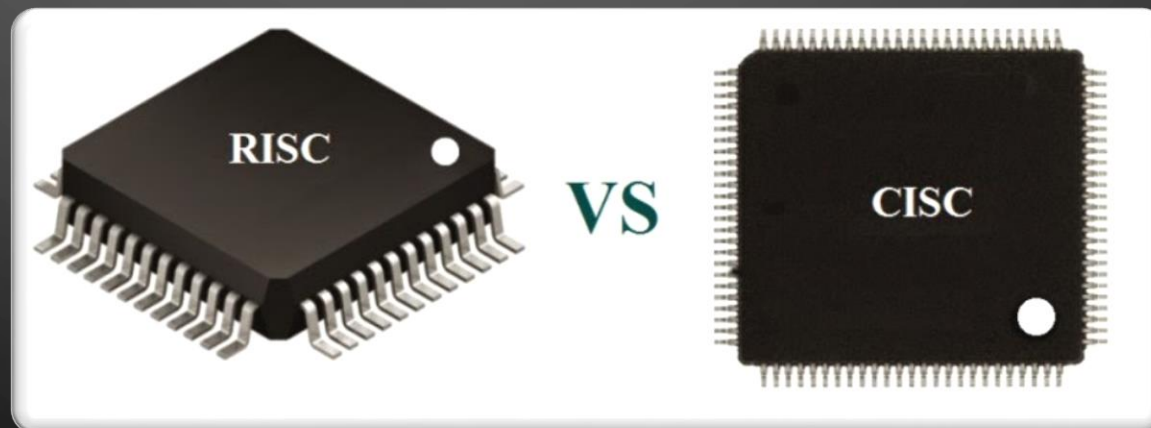
Центральный процессор выполняет каждую команду за несколько шагов:

- Вызывает следующую команду из памяти и переносит ее в регистр команд.
- Меняет положение счетчика команд, который после этого указывает на следующую команду.
- Определяет тип вызванной команды.
- Если команда использует слово из памяти, определяет, где находится это слово.
- Переносит слово, если это необходимо, в регистр центрального процессора.
- Выполняет команду.
- Переходит к шагу 1, чтобы начать выполнение следующей команды.
- Последовательность шагов (выборка — декодирование — исполнение) является основой работы всех компьютеров.



RISC, CISC

- Компьютер RISC (Reduced Instruction Set Computer — компьютер с сокращенным набором команд) противопоставлялся системе CISC (Complex Instruction Set Computer — компьютер с полным набором команд). В качестве примера машины типа CISC можно привести компьютер VAX, который доминировал в то время в университетской среде.
- В 1980 году группа разработчиков в университете Беркли начала разработку не ориентированных на интерпретацию процессоров VLSI. Для обозначения этого понятия они придумали термин RISC, а новый процессор назвали RISC I, вслед за которым вскоре был выпущен RISC II.
- В то время, когда разрабатывались эти простые процессоры, всеобщее внимание привлекало относительно небольшое количество команд (обычно около 50). Для сравнения: число команд в компьютерах VAX производства DEC и больших компьютерах производства IBM в то время составляло от 200 до 300.



ОПЕРАЦИОННЫЕ СИСТЕМЫ: УПРАВЛЕНИЕ ПРОЦЕССАМИ

- Процесс — программа, которая выполняется в текущий момент. Стандарт ISO 9000:2000 определяет процесс как совокупность взаимосвязанных и взаимодействующих действий, преобразующих входящие данные в исходящие.
- В вычислительной технике, процесс является экземпляром компьютерной программы, которая выполняется. Он содержит программный код и его текущей деятельности. В зависимости от используемой операционной системы (ОС), процесс может быть составлен из нескольких потоков выполнения, которые выполняют инструкции одновременно.
- Компьютерная программа сама по себе — это только пассивная совокупность инструкций, в то время как процесс — это непосредственное выполнение этих инструкций.
- Также, процессом называют выполняющуюся программу и все её элементы: адресное пространство, глобальные переменные, регистры, стек, открытые файлы и т. д.
- Компьютерная программа представляет собой пассивный сбор инструкций, в то время как процесс является фактическим выполнением этих инструкций. Несколько процессов могут быть связаны с той же самой программой; например, открывая несколько экземпляров одной и той же программы часто означает нечто большее, чем выполнение одного процесса.



УПРАВЛЕНИЕ ПРОЦЕССАМИ

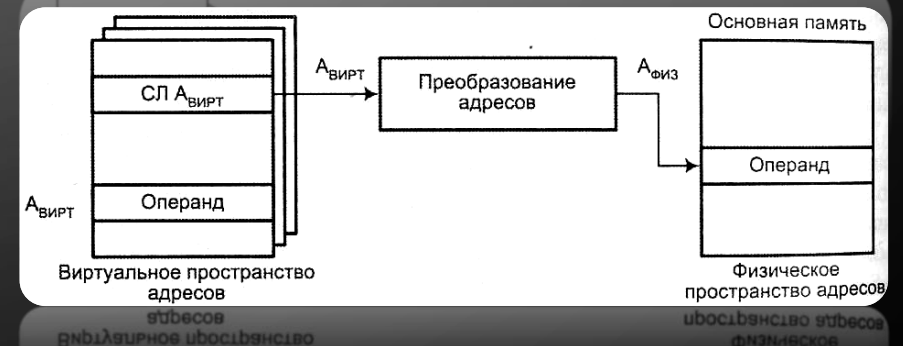
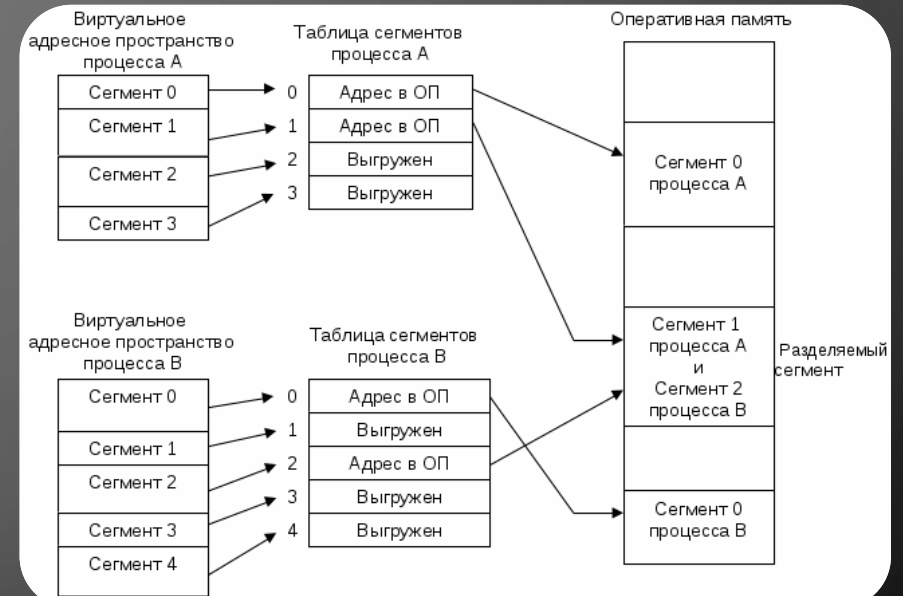
- Метод многозадачности позволяет нескольким процессам совместно использовать процессоры (CPU) и другие системные ресурсы. Каждый процессор выполняет одну задачу за один раз. Тем не менее, многозадачность позволяет каждому процессору переключаться между задачами, которые выполняются без необходимости ждать каждой задачи, чтобы закончить. В зависимости от реализации операционной системы, коммутаторы могут быть выполнены, когда задачи выполнения операций ввода/вывода, когда задача указывает на то, что он может быть включен, или на аппаратных прерываниях. Распространенной формой многозадачности с разделением времени. Время обмена является способ для обеспечения быстрого отклика для интерактивных пользовательских приложений. В разделения времени систем, переключение контекста выполняются быстро, что создает впечатление, что несколько процессов выполняется одновременно на одном процессоре. Эта кажущаяся выполнение нескольких процессов одновременно называется параллелизмом.
- В целях обеспечения безопасности и надежности, большинство современных операционных систем предотвращения прямой связи между независимыми процессами, обеспечивают строго опосредованные и контролируемые функциональные возможности межпроцессного взаимодействия.



ВИРТУАЛЬНАЯ ПАМЯТЬ

- Виртуальная память (англ. Virtual Memory) — метод управления памятью компьютера, позволяющий выполнять программы, требующие больше оперативной памяти, чем имеется в компьютере, путём автоматического перемещения частей программы между основной памятью и вторичным хранилищем (например, жёстким диском). Для выполняющейся программы данный метод полностью прозрачен и не требует дополнительных усилий со стороны программиста, однако реализация этого метода требует как аппаратной поддержки, так и поддержки со стороны операционной системы.
- В системе с виртуальной памятью используемые программами адреса, называемые виртуальными адресами, транслируются в физические адреса в памяти компьютера. Трансляцию виртуальных адресов в физические выполняет аппаратное обеспечение, называемое блоком управления памятью. Для программы основная память выглядит как доступное и непрерывное адресное пространство либо как набор непрерывных сегментов, вне зависимости от наличия у компьютера соответствующего объёма оперативной памяти. Управление виртуальными адресными пространствами, соотнесение физической и виртуальной памяти, а также перемещение фрагментов памяти между основным и вторичным хранилищами выполняет операционная система.

Виртуальное адресное пространство процесса

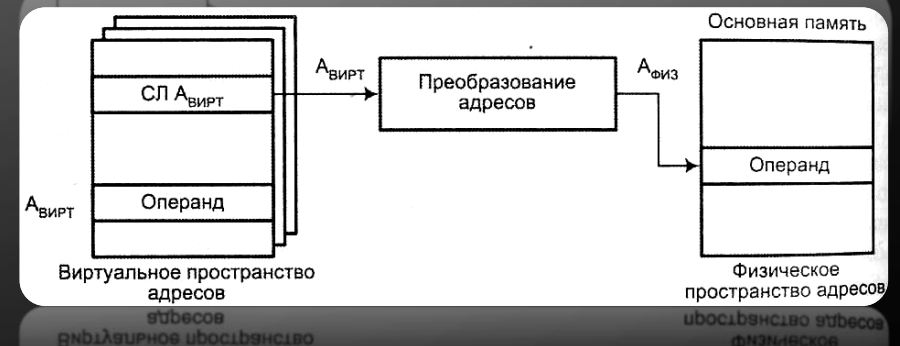
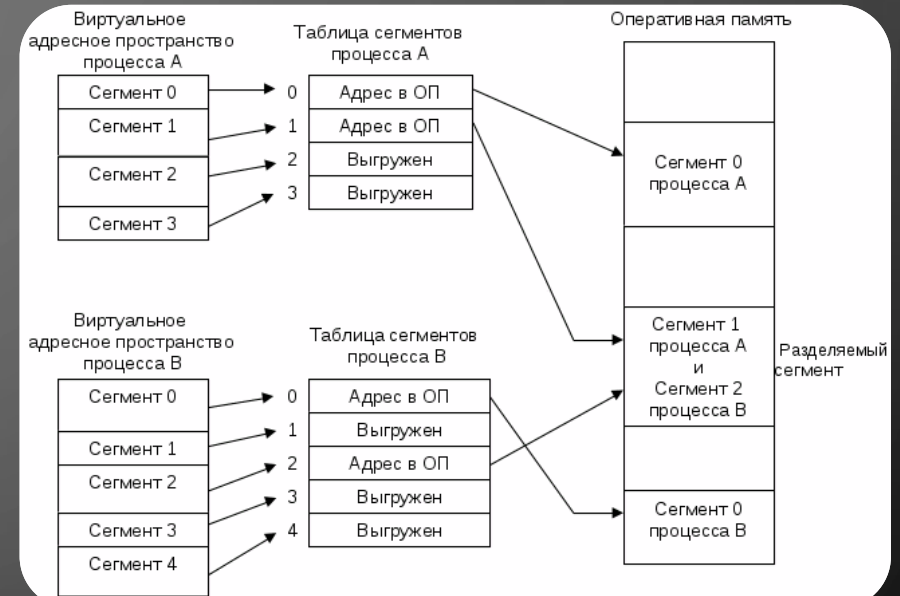


ВИРТУАЛЬНАЯ ПАМЯТЬ

Применение виртуальной памяти позволяет:

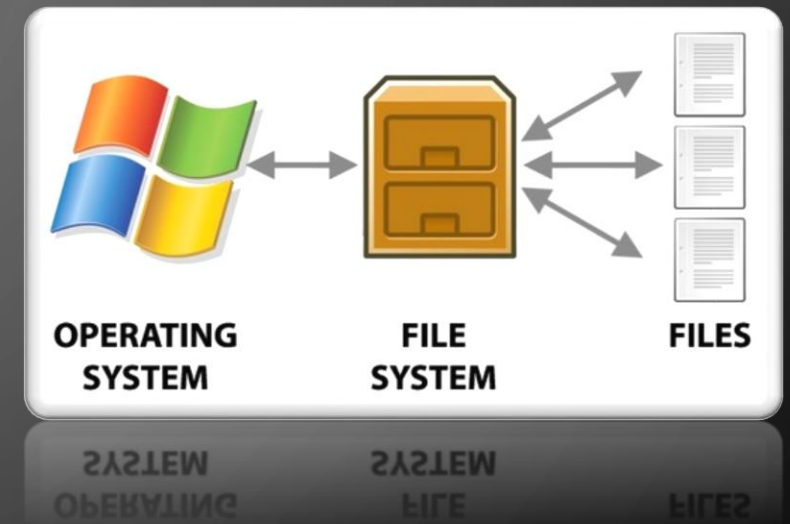
- освободить программиста от необходимости вручную управлять загрузкой частей программы в память и согласовывать использование памяти с другими программами
- предоставлять программам больше памяти, чем физически установлено в системе
- в многозадачных системах изолировать выполняющиеся программы друг от друга путём назначения им непересекающихся адресных пространств.
- В настоящее время виртуальная память аппаратно поддерживается в большинстве современных процессоров. В то же время в микроконтроллерах и в системах специального назначения, где требуется либо очень быстрая работа, либо есть ограничения на длительность отклика (системы реального времени), виртуальная память используется относительно редко. Также в таких системах реже встречается многозадачность и сложные иерархии памяти.

Виртуальное адресное пространство процесса



ФАЙЛОВАЯ СИСТЕМА

Файловая система — порядок, определяющий способ организации, хранения и именования данных на носителях информации в компьютерах. Файловая система определяет формат содержимого и способ физического хранения информации, которую принято группировать в виде файлов. Конкретная файловая система определяет размер имен файлов (и каталогов), максимальный возможный размер файла и раздела, набор атрибутов файла. Некоторые файловые системы предоставляют сервисные возможности, например, разграничение доступа или шифрование файлов.

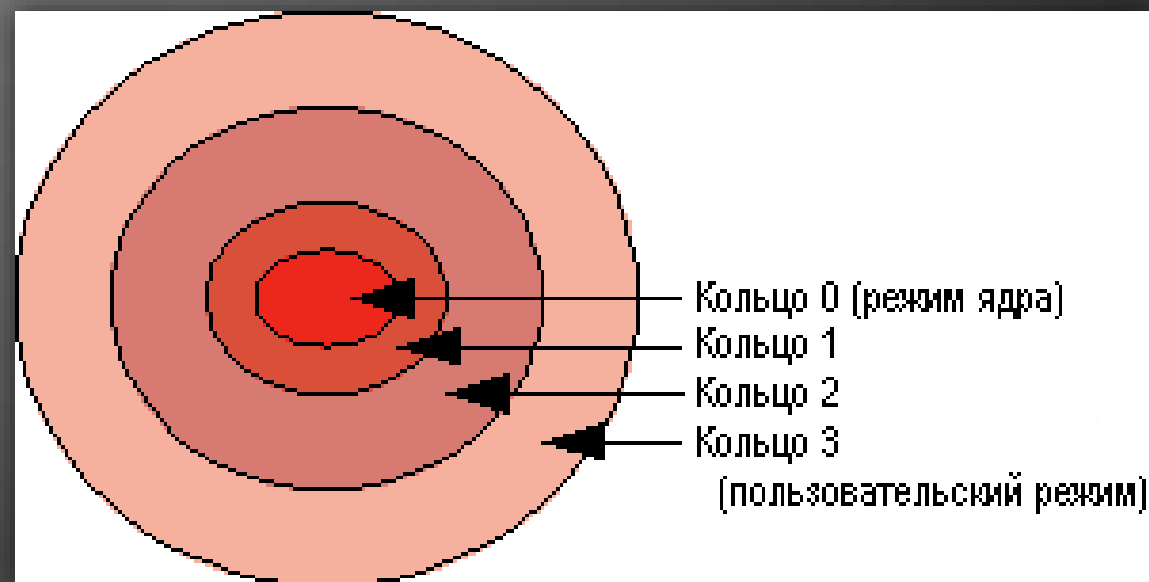


ФАЙЛОВАЯ СИСТЕМА

- Файловая система связывает носитель информации с одной стороны и API для доступа к файлам — с другой. Когда прикладная программа обращается к файлу, она не имеет никакого представления о том, каким образом расположена информация в конкретном файле, также, как и на каком физическом типе носителя (CD, жёстком диске, магнитной ленте, блоке флеш-памяти или другом) он записан. Всё, что знает программа — это имя файла, его размер и атрибуты. Эти данные она получает от драйвера файловой системы. Именно файловая система устанавливает, где и как будет записан файл на физическом носителе (например, жёстком диске).
- С точки зрения операционной системы (ОС), весь диск представляет собой набор кластеров (как правило, размером 512 байт и больше). Драйверы файловой системы организуют кластеры в файлы и каталоги (реально являющиеся файлами, содержащими список файлов в этом каталоге). Эти же драйверы отслеживают, какие из кластеров в настоящее время используются, какие свободны, какие помечены как неисправные.
- Однако файловая система не обязательно напрямую связана с физическим носителем информации. Существуют виртуальные файловые системы, а также сетевые файловые системы, которые являются лишь способом доступа к файлам, находящимся на удалённом компьютере.

РЕЖИМ ЯДРА

- Пользовательский режим - наименее привилегированный режим, поддерживаемый NT; он не имеет прямого доступа к оборудованию и у него ограниченный доступ к памяти.
- Режим супервизора, привилегированный режим, режим ядра (англ. Kernel mode) — привилегированный режим работы процессора, как правило, используемый для выполнения ядра операционной системы.

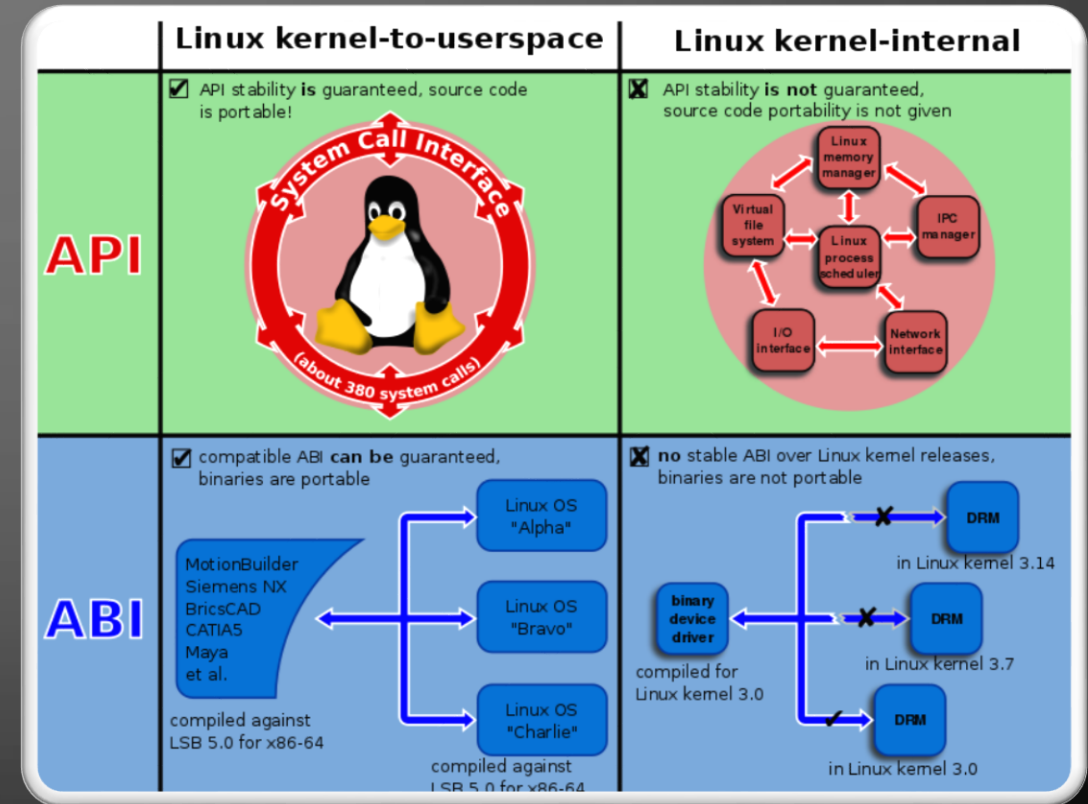


РЕЖИМ ЯДРА

- При обсуждении архитектуры ОС Windows NT постоянно используются понятия «режим пользователя» и «режим ядра», поэтому стоит определить, что это значит. Начнем с обсуждения разницы между пользовательским режимом и режимом ядра (User mode/Kernel mode).
- Повторюсь и скажу, что режим супервизора, привилегированный режим, режим ядра (англ. Kernel mode) — привилегированный режим работы процессора, как правило, используемый для выполнения ядра операционной системы.
- Различия в работе программ пользовательского режима и режима ядра поддерживаются аппаратными средствами компьютера (а именно - процессором).
- Большинство архитектур процессоров обеспечивают, по крайней мере, два аппаратных уровня привилегий. Аппаратный уровень привилегий процессора определяет возможное множество инструкций, которые может вызывать исполняемый в данный момент процессором код. Хотя понятия «режим пользователя» и «режим ядра» часто используются для описания кода, на самом деле это уровни привилегий, ассоциированные с процессором. Уровень привилегий накладывает три типа ограничений: 1) возможность выполнения привилегированных команд, 2) запрет обращения к данным с более высоким уровнем привилегий, 3) запрет передачи управления коду с уровнем привилегий, не равным уровню привилегий вызывающего кода.

СИСТЕМНЫЙ ВЫЗОВ

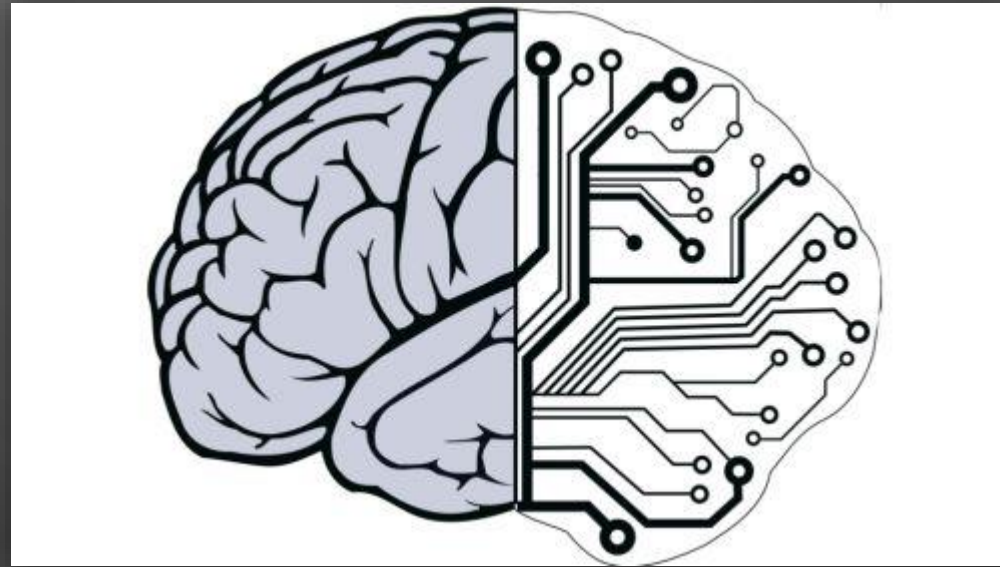
- Системный вызов (англ. system call) в программировании и вычислительной технике — обращение прикладной программы к ядру операционной системы для выполнения какой-либо операции.
- Интерфейс ядро — прикладные приложения ОС Linux: около 380 системных вызовов.



СИСТЕМНЫЙ ВЫЗОВ

- Современные операционные системы (ОС) предусматривают разделение времени между выполняющимися вычислительными процессами (многозадачность) и разделение полномочий, препятствующее обращению исполняемых программ к данным других программ и оборудованию. Ядро ОС выполняется в привилегированном режиме работы процессора. Для выполнения межпроцессной операции или операции, требующей доступа к оборудованию, программа обращается к ядру, которое, в зависимости от полномочий вызывающего процесса, исполняет либо отказывает в исполнении такого вызова.
- С точки зрения программиста, системный вызов обычно выглядит как вызов подпрограммы или функции из системной библиотеки. Однако системный вызов, как частный случай вызова такой функции или подпрограммы, следует отличать от более общего обращения к системной библиотеке, поскольку последнее может и не требовать выполнения привилегированных операций.

СПАСИБО ЗА ВНИМАНИЕ!



Презентацию подготовил студент:

Факультета ИТ

БПЗ1501

Тодика А. А.

Специально для ассистента кафедры ИБ:

Баркова В. В.

Москва, 2018 год.