Архитектура ЭВМ и операционные системы

Венатовская Людмила Александровна
l.venatovskaya@spbu.ru

Введение

- **Программа –** последовательность команд, описывающих решение определенной задачи**.**

Все программы перед исполнением должны быть превращены в последовательность таких команд, которые обычно не сложнее, чем, например:

- сложить два числа;
- проверить, не является ли число нулем;
- скопировать блок данных из одной части памяти компьютера в другую.

Эти примитивные команды в совокупности составляют язык, на котором люди могут общаться с компьютером. Такой язык называется **машинным.**

Введение

Машинные команды стараются сделать как можно проще:

- ✔ чтобы избежать сложностей при разработке компьютера
- ✔ снизить затраты на необходимую электронику.

- Машинные языки крайне примитивны, из-за чего писать на них и трудно, и утомительно.

Многоуровневая компьютерная организация –

подход, который заключается в построении ряда уровней абстракций, каждая из которых надстраивается над абстракцией более низкого уровня.

Многоуровневая компьютерная организация

Существует огромная разница между тем, что удобно людям, и тем, что могут компьютеры.

Проблема

Люди хотят сделать **X**,

Но! компьютеры могут сделать только ***Y***.

Языки, уровни и виртуальные машины

Пусть

- **Язык Я0** – встроенные машинные команды.
- **Язык Я1** – язык, который состоит из команд, более удобных для человека, чем встроенные машинные команды.

Компьютер может исполнять только программы, написанные на его машинном языке Я0.

Каким образом компьютер будет исполнять программы, написанные на языке Я1 - ведь, в конечном итоге, компьютеру доступен только машинный язык Я0?

Трансляция и интерпретация

Первый способ **(трансляция)**:

подразумевает замену каждой команды на языке Я1 эквивалентным набором команд на языке Я0. В этом случае компьютер исполняет новую программу, написанную на языке Я0, вместо старой программы, написанной на Я1.

Трансляция и интерпретация

Второй способ (**интерпретация)**:

создаем на языке Я0 программу, получающей в качестве входных данных программы, написанные на языке Я1.

При этом каждая команда языка Я1 обрабатывается поочередно, после чего сразу исполняется эквивалентный ей набор команд языка Я0.

Эта технология не требует составления новой программы на Я0, а программа, которая осуществляет интерпретацию, называется **интерпретатором.**

Сходство:

в обоих случаях компьютер в конечном итоге исполняет набор команд на языке Я0, эквивалентных командам Я1.

Отличие:

при трансляции вся программа Я1 переделывается в программу Я0, программа Я1 отбрасывается, а новая программа на Я0 загружается в память компьютера и затем исполняется. Во время выполнения сгенерированная программа на Я0 управляет работой компьютера.

при интерпретации каждая команда программы на Я1 перекодируется в Я0 и сразу же исполняется. *Транслированная* программа при этом не создается. Работой компьютера управляет интерпретатор, для которого программа на Я1 есть не что иное, как «сырые» входные данные.

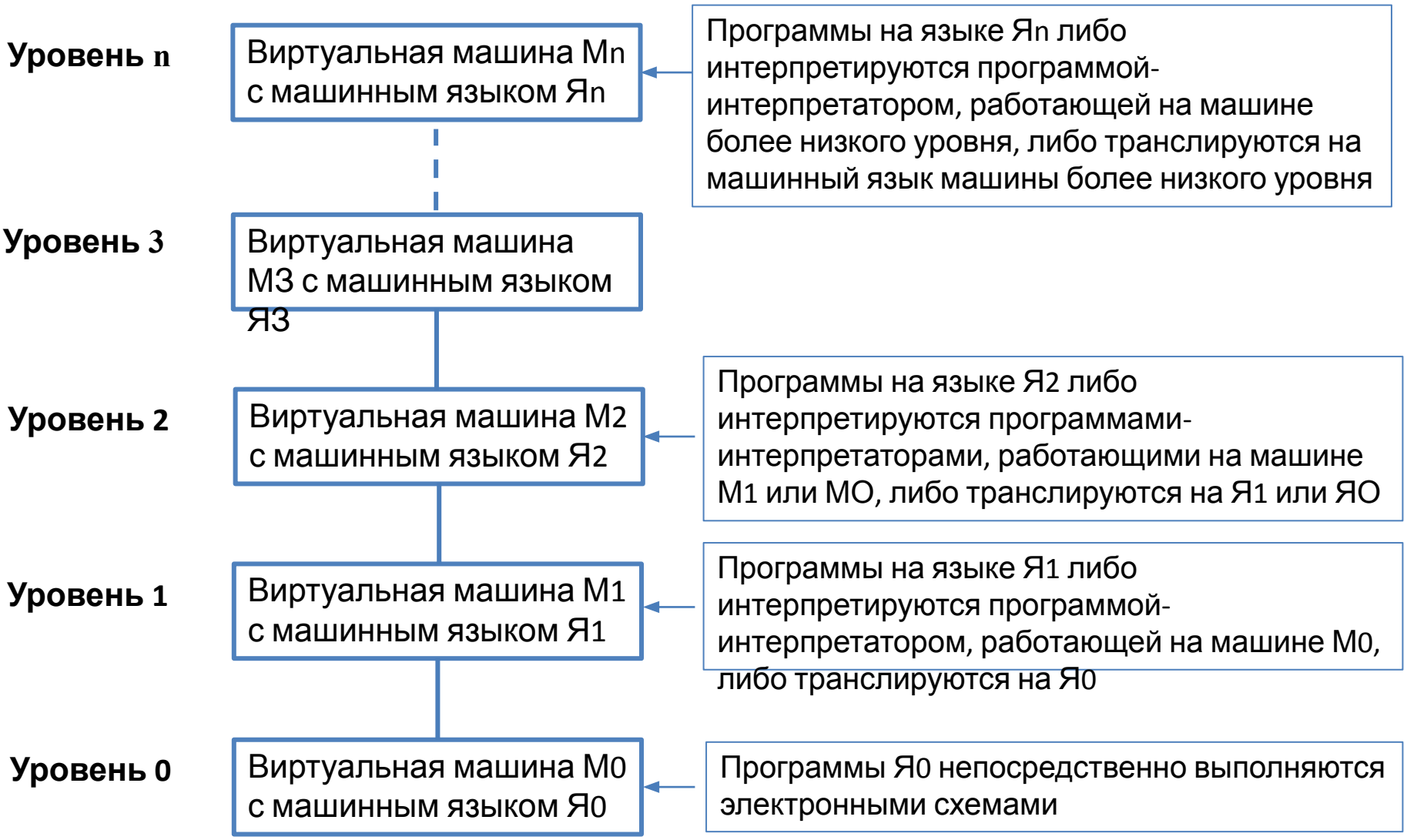
Оба подхода широко используются как вместе, так и по отдельности.

Многоуровневая машина

- Представим себе существование **виртуальной машины,** для которой машинным языком является язык Я1. Назовем такую виртуальную машину М1, а виртуальную машину для работы с языком Я0 — М0.
- Язык ориентированный на человека, будем называть Я2, а соответствующую виртуальную машину — М2.

Человек может писать программы на языке Я2, как будто виртуальная машина для работы с машинным языком Я2 действительно существует. Такие программы могут либо транслироваться на язык Я1, либо исполняться интерпретатором, написанным на языке Я1.

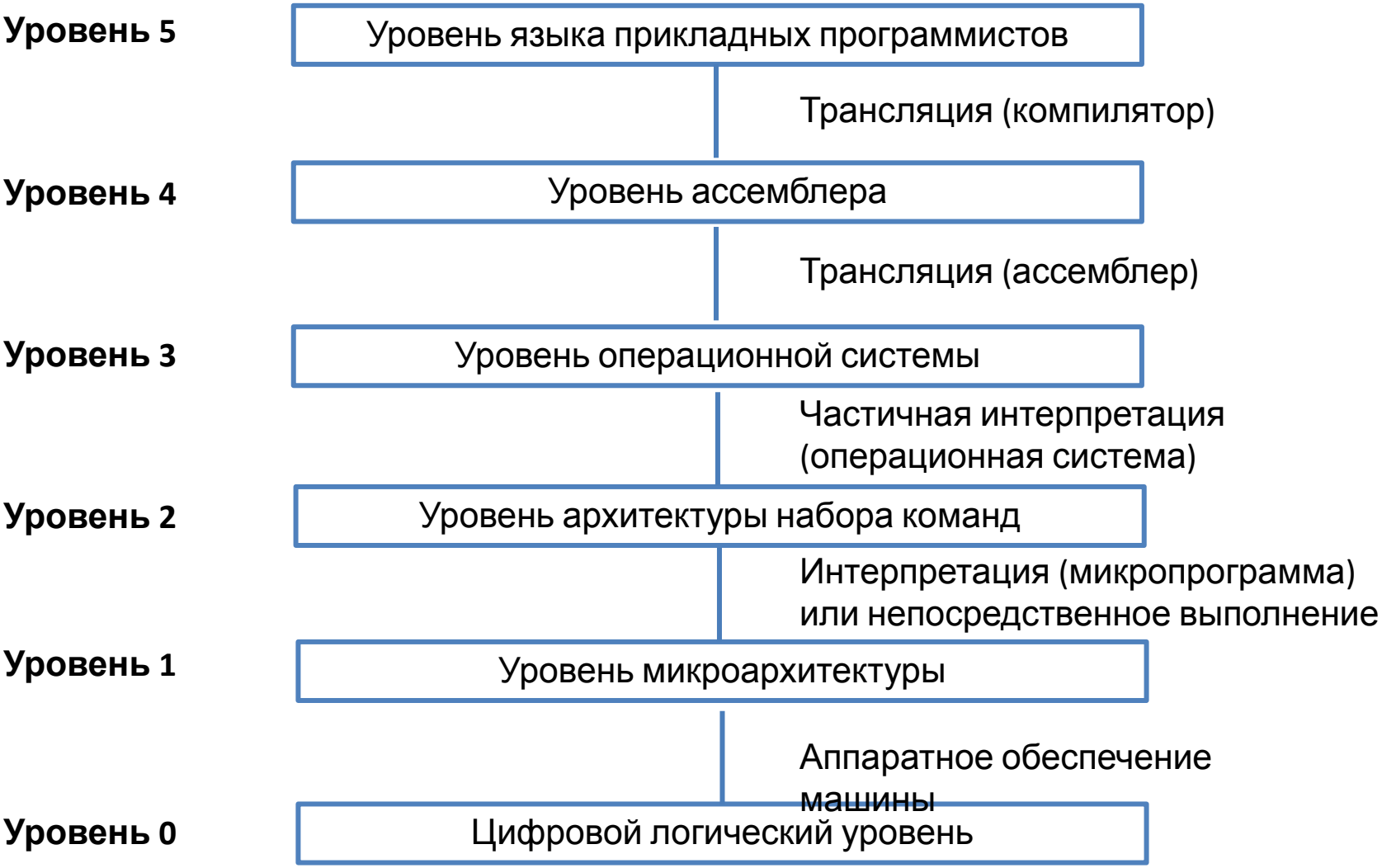
Многоуровневая машина



Многоуровневая машина

- Компьютер с *n* уровнями можно рассматривать как *n* разных виртуальных машин, у каждой из которых есть свой машинный язык.
- Только программы, написанные на Я0, могут исполняться компьютером без трансляции или интерпретации. Программы, написанные на Я1, Я2, Яn, должны проходить через интерпретатор более низкого уровня или транслироваться на язык, соответствующий более низкому уровню.
- Человеку, который пишет программы для виртуальной машины уровня n, не обязательно знать о трансляторах и интерпретаторах более низких уровней. Машина исполнит эти программы, и не важно, будут они поэтапно исполняться интерпретатором или же их обработает сама машина. В обоих случаях результат один и тот же — это исполнение программы.

Современные многоуровневые машины



Шестиуровневый компьютер.

Уровень 0. Цифровой логический уровень

Уровень 0 – это аппаратное обеспечение машины. Его электронные схемы исполняют машинно-зависимые программы уровня 1.

PS: *ниже уровня 0 -* ***уровень физических устройств.*** *На этом уровне находятся транзисторы, которые для разработчиков компьютеров являются примитивами.*

На **цифровом логическом уровне,** объекты называются **вентилями.**

- Хотя вентили строятся из аналоговых компонентов (таких как транзисторы), могут быть точно смоделированы как цифровые устройства.
- У каждого вентиля есть одно или несколько цифровых входных данных (сигналов, представляющих 0 или 1).
- Вентиль вычисляет простые функции этих сигналов, такие как И или ИЛИ.
- Каждый вентиль формируется из нескольких транзисторов.
- Несколько вентилей формируют 1 бит памяти, который может содержать 0 или 1.
- Биты памяти, объединенные в группы, например, по 16, 32 или 64, формируют **регистры.**
- Каждый регистр может содержать одно двоичное число в определенном диапазоне.
- Из вентилей также может строиться само ядро вычислительной системы.

Уровень 1. Уровень микроархитектуры

- На этом уровне находятся наборы из (обычно) 8 или 32 регистров, которые формируют локальную память и схему, называемую **АЛУ (арифметико-логическое устройство).**
- АЛУ исполняет простые арифметические операции.
- Регистры вместе с АЛУ формируют **тракт данных,** по которому поступают данные.

Базовая операция тракта: выбирается один или два ре гистра, АЛУ производит над ними какую-либо операцию (например сложение), после чего результат вновь помещается в какой-либо регистр.

Работа тракта данных может контролироваться:

- ✔ особой программой, которая называется **микропрограммой**
- ✔ напрямую **аппаратными средствами**.

- На машинах, где тракт данных контролируется программным обеспечением, микропрограмма – это интерпретатор для команд на уровне 2. Микропрограмма читает команды из памяти и исполняет их одну за другой, используя при этом тракт данных.

Например, при исполнении команды add она вызывается из памяти, ее операнды помещаются в регистры, АЛУ вычисляет сумму, а затем результат направляется туда, где он должен находиться.

- На компьютере с аппаратным управлением тракта данных происходит такая же процедура, но при этом нет программы, интерпретирующей команды уровня 2.

Уровень 2. Уровень архитектуры набора команд

- Набор машинных команд, которые исполняются микропрограммой-интерпретатором или аппаратным обеспечением.

Уровень 3. Уровень операционной системы

Дополнительные особенности:

- новый набор команд,
- другая организация памяти,
- способность исполнять две и более программ одновременно и т.д.

Гибридный уровень: одна часть команд уровня 3 интерпретируется операционной системой, а другая часть — микропрограммой.

Новые средства, появившиеся на уровне 3, исполняются интерпретатором (был назван операционной системой), который работает на втором уровне. Команды уровня 3, идентичные командам уровня 2, исполняются микропрограммой или аппаратным обеспечением, но не ОС.

Уровень 4. Уровень ассемблера

- Между уровнями 3 и 4 есть принципиальная разница. Нижние три уровня не предназначены для использования рядовыми программистами. Они изначально ориентированы на интерпретаторы и трансляторы, обеспечивающие работу на более высоких уровнях. Эти трансляторы и интерпретаторы создаются **системными программистами,** которые специализируются на разработке новых виртуальных машин. Уровни с четвертого и выше предназначены для прикладных программистов, решающих конкретные задачи.
- Еще одно изменение, появившееся на уровне 4, — механизм поддержки более высоких уровней. Уровни 2 и 3 всегда интерпретируются, а уровни 4, 5 и выше обычно (хотя и не всегда) транслируются.

Уровень 4. Уровень ассемблера

- Уровень 4 представляет собой символическую форму одного из языков более низкого уровня. На этом уровне человек может писать программы для уровней 1, 2 и 3 в форме не настолько неприятной, как язык виртуальных машин. Эти программы сначала транслируются на язык уровня 1, 2 или 3, а затем интерпретируются соответствующей виртуальной или реально существующей машиной. Программа, которая исполняет трансляцию, называется **ассемблером.**

Уровень 5. Языки высокого уровня

- Состоит из языков, разработанных для прикладных программистов. Существуют сотни языков высокого уровня. Наиболее известные среди них — C, C++, Java, Perl, Python и PHP. Программы, написанные на этих языках, обычно транслируются на уровень 3 или 4. Трансляторы, которые обрабатывают эти программы, называются **компиляторами,** хотя в некоторых случаях имеет место интерпретация. Например, программы на языке Java сначала транслируются на язык, напоминающий машинные команды и называемый байт-кодом Java, который затем интерпретируется.

Современные многоуровневые машины

- Компьютер проектируется как иерархическая структура уровней, которые надстраиваются друг над другом.
- Каждый уровень представляет собой абстракцию некоторых объектов и операций.
- Набор типов данных, операций и характеристик каждого отдельно взятого уровня называется **архитектурой.**

Архитектура связана с аспектами, видимыми пользователю этого уровня. Например, сведения о том, сколько памяти можно использовать при написании программы, — часть архитектуры. Аспекты реализации (например, технология, применяемая при реализации памяти) не являются частью архитектуры.