

ПО сетевых устройств

Трещановский Павел Александрович, к.т.н.

28.02.20

Пример встроенной системы



Основные области применения встроенных систем

Согласно отчету www.marketresearchfuture.com за 2020 год:

- Automotive
- Telecommunication
- Healthcare
- Industrial
- Consumer Electronics
- Military & Aerospace

Промышленные и транспортные системы



- Двигатели
- Нефтепроводы, газопроводы
- Датчики (давления, температуры, влажности и др.)
- Промышленные контроллеры
- Мультимедийные системы
- Позиционирование

Телекоммуникационные системы



- Коммутаторы
- Маршрутизаторы
- Точки доступа
- Базовые станции
- Модемы

Требования к встроенным системам

- Работа в режиме реального времени
- Высокая производительность (в рамках отведенной задачи)
- Надежность
- Низкая мощность и тепловыделение
- Доверенность
- Низкая цена

Некоторые факты о встроенных системах

Christof Ebert, Embedded Software: Facts, Figures and Future:

- На каждого человека приходится 30 микропроцессоров
- 98 микропроцессоров находятся во встроенных системах
- В каждом автомобиле - от 20 до 70 управляющих устройств (100 миллионов машинных инструкций)

Прогноз www.marketresearchfuture.com:

- Объем мирового рынка встроенных систем к 2025 году - 291 миллиард долларов.

Ключевые тенденции: умный город

Проблемы:

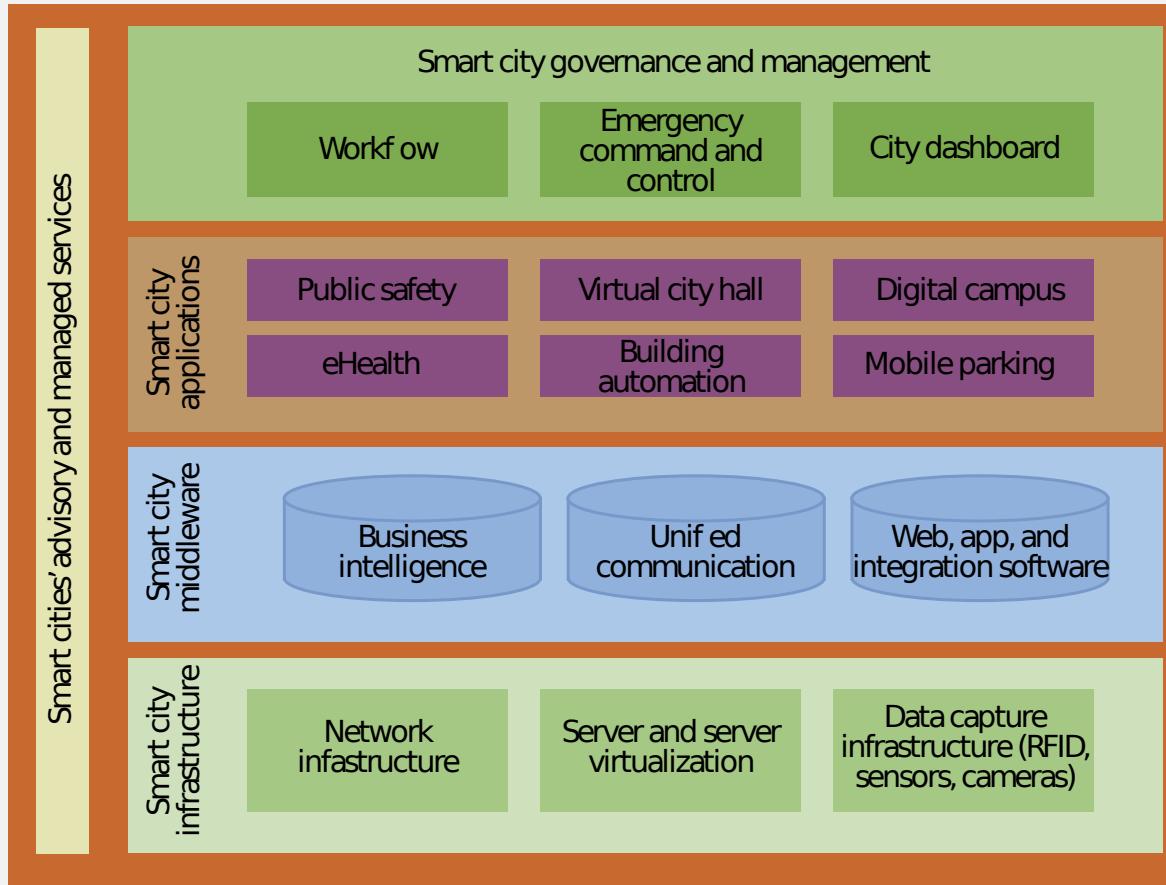
- Рост населения
- Рост урбанизации
- Исчерпание ресурсов
- Разрозненность и фрагментарность существующих информационных технологий

Определение умного города согласно аналитическому агентству Forrester:
A „city” that uses information and communications technologies to make the critical infrastructure components and services of a city — administration, education, healthcare, public safety, real estate, transportation, and utilities — more aware, interactive, and efficient.

Составляющие умного города

- Коммунальные службы: автоматический дистанционный учет ресурсов, гибкое распределение ресурсов.
- Транспорт: предсказание трафика, управление трафиком в зависимости от текущей нагрузки.
- Здравоохранение: электронная история болезни, дистанционное наблюдение за пациентами.
- Безопасность: видеонаблюдение.
- Умный дом: мониторинг и управление потреблением тепла, воды, света и др.

Инфраструктура умного города



Ключевые тенденции: Интернет вещей

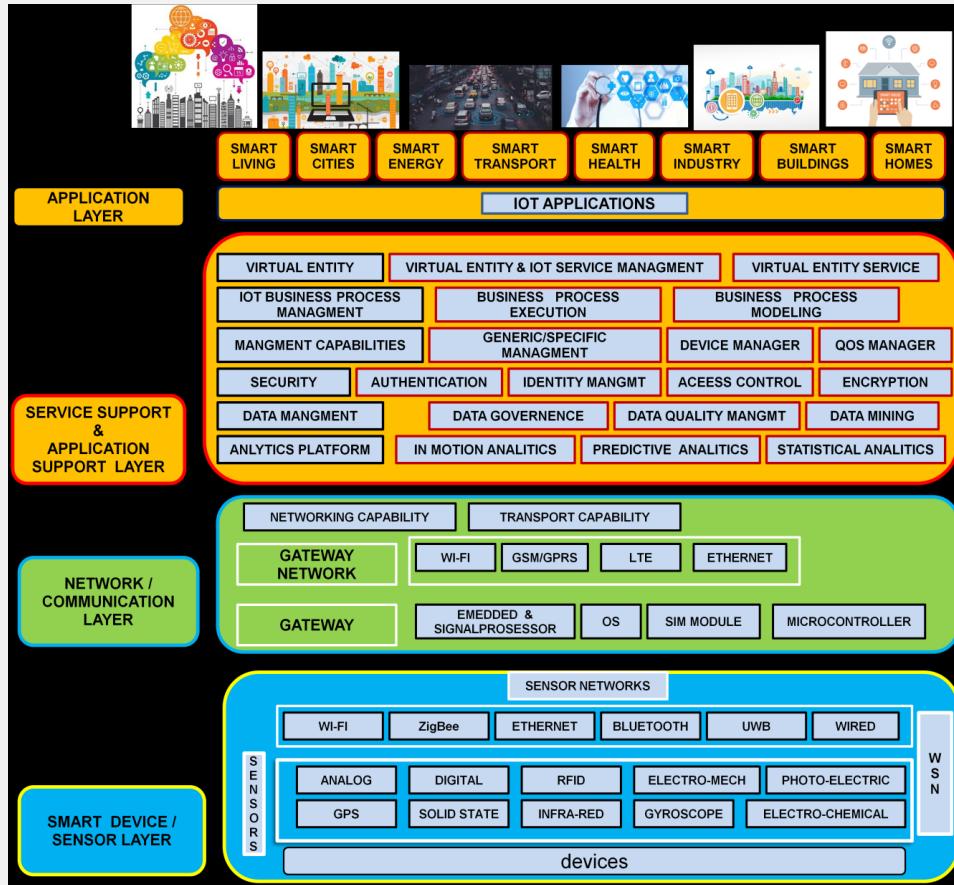
"An open and comprehensive network of intelligent objects that have the capacity to auto-organize, share information, data and resources, reacting and acting in face of situations and changes in the environment"

"The Internet [of things] is not only a network of computers, but it has evolved into a network of devices of all types and sizes, vehicles, smartphones, home appliances, toys, cameras, medical instruments and industrial systems, all connected, all communicating and sharing information all the time"

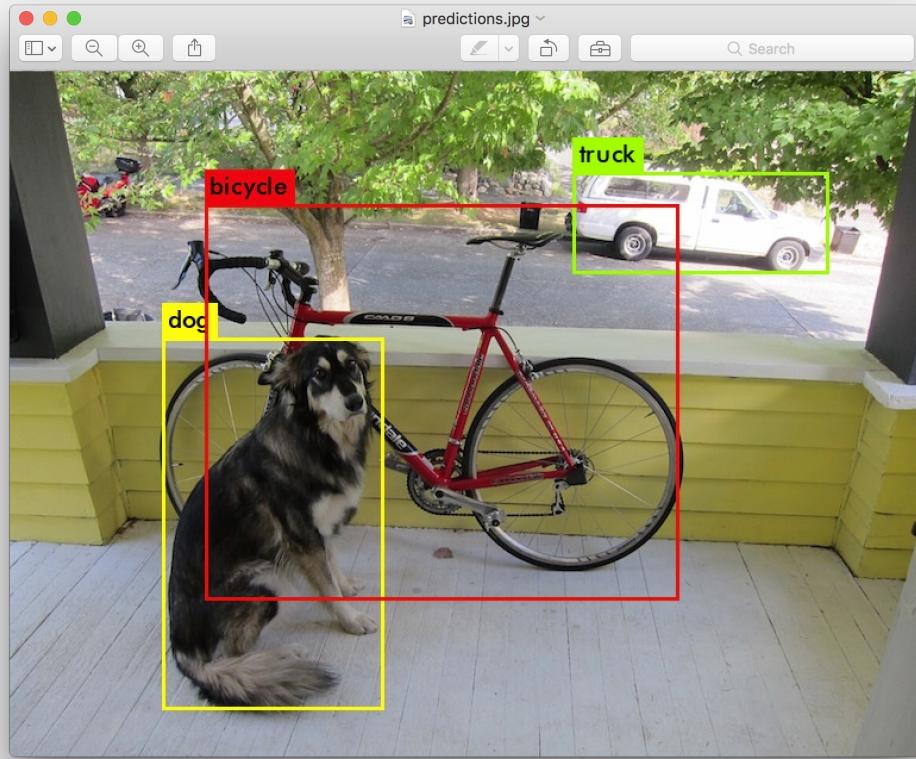
Интернет вещей, свойства

- Связность: все подключено к глобальной инфраструктуре.
- Гетерогенность: разнообразие аппаратных платформ и сетевых технологий.
- Масштаб: на порядок больше устройств, чем в традиционном Интернете.
- Динамичность: изменение количества и местоположения устройств, изменение состояния устройств (включение, выключение, засыпание, пробуждение).
- Защищенность: безопасная передача личных данных через глобальную сеть.

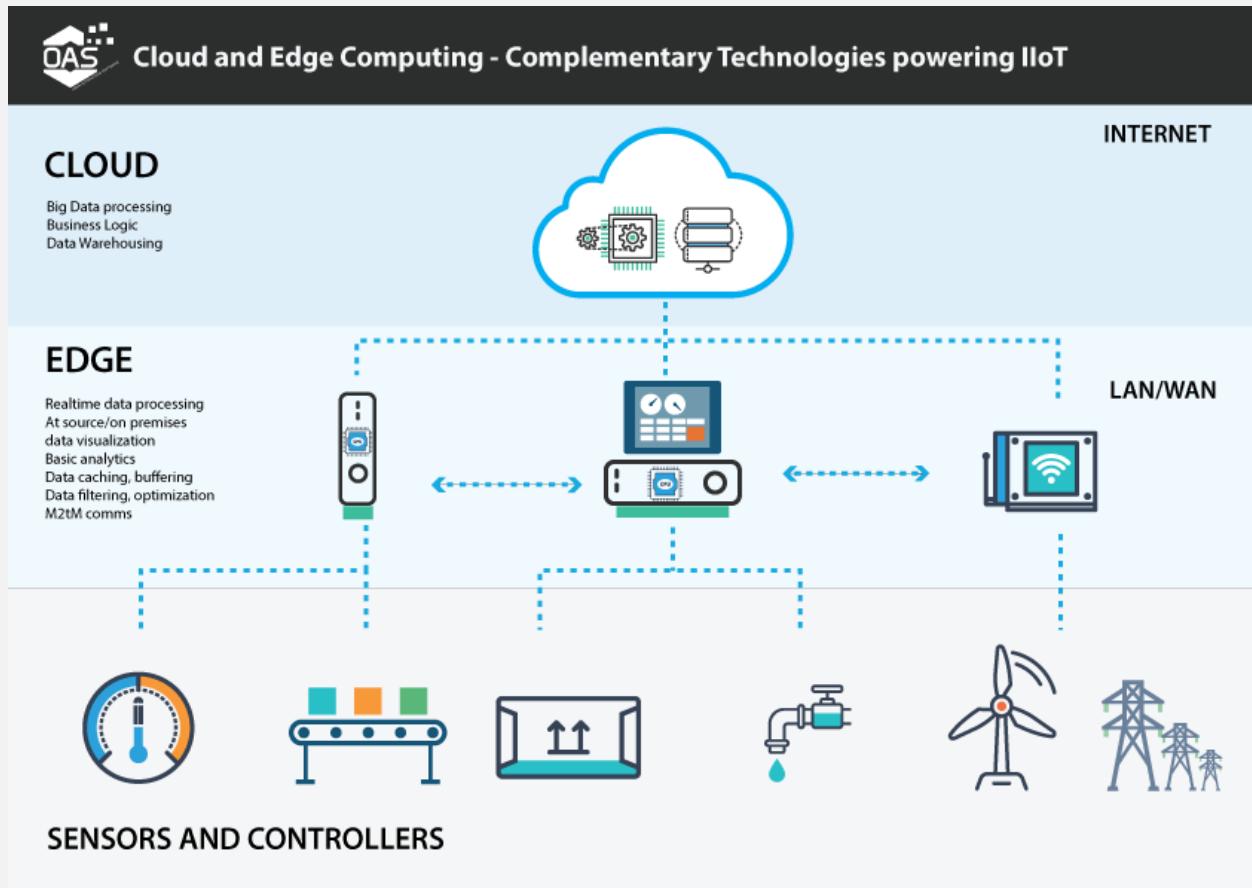
Интернет вещей, архитектура



Ключевые тенденции: искусственный интеллект



Границочные вычисления (Edge computing)



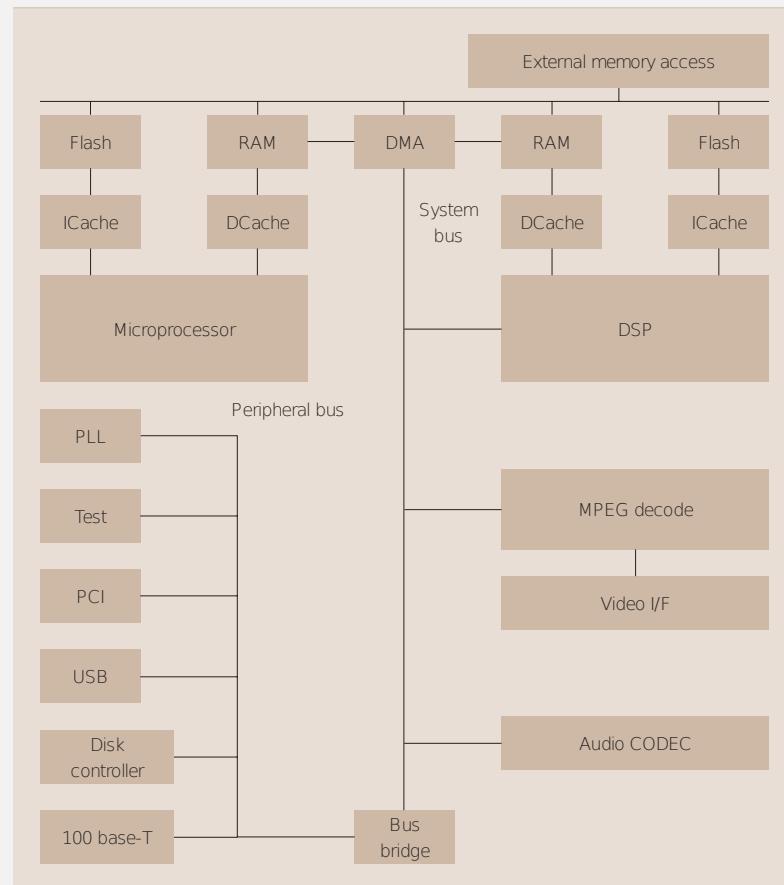
Преимущества граничных вычислений

- Высокая отзывчивость: меньше время передачи данных и больше пропускная способность.
- Масштабируемость: снижение нагрузки на облако за счет локальной предварительной обработки данных.
- Безопасность: не требуется передача всех данных через глобальную сеть.
- Надежность: защита от сбоя облака или сети передачи данных.

Ключевые тенденции, замечания

- Встроенные системы будут играть ключевую роль.
- Расширение сетевой инфраструктуры: сеть доходит до каждого физического объекта.
- В глобальной сети вещей все устройства будут сетевыми, не только коммутаторы, маршрутизаторы.
- В России развитие и внедрение Интернета вещей, искусственного интеллекта и концепции "Умный дом" будет происходить в рамках национального проекта "Цифровая экономика". Сквозные цифровые технологии "Новые производственные технологии", "Технологии беспроводной связи" и др.

Система на кристалле - основа встроенной системы



Процессорные ядра ARM и MIPS

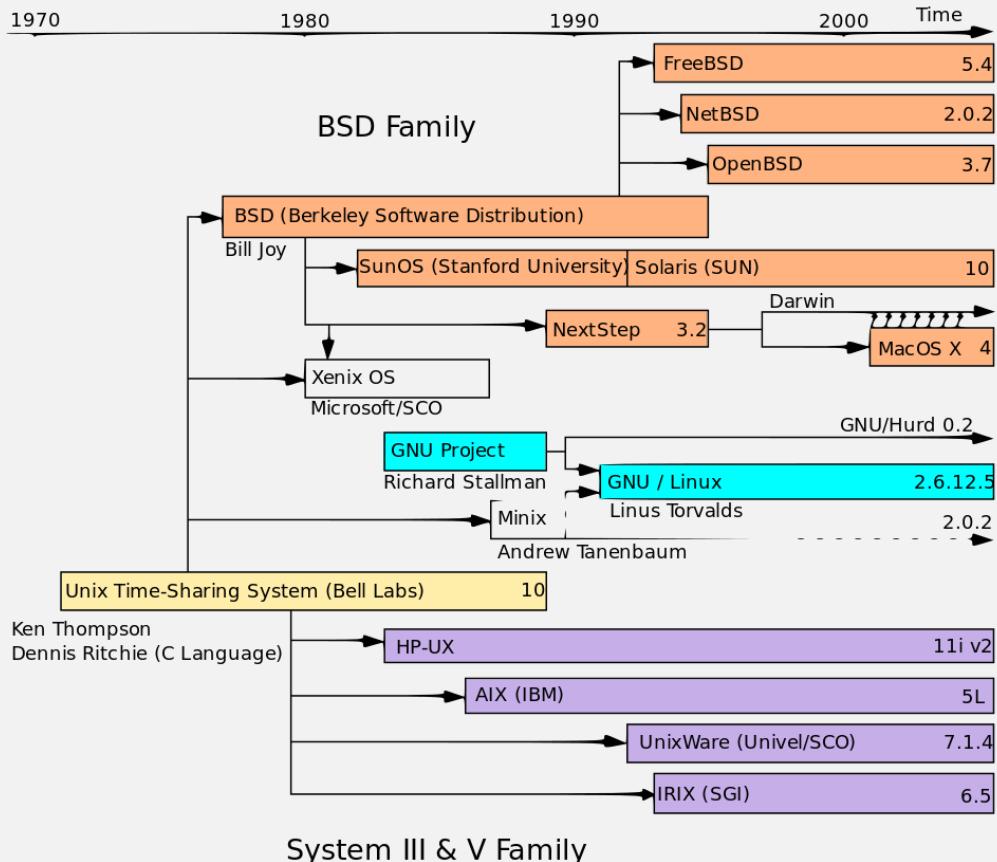
Ядро ARM

- Архитектура RISC.
- Существуют 32-битная и 64-битная версия.
- Старшая линейка ядер - Cortex-A. Типичная частота 1-1.5 ГГц. Количество ядер на кристалле - 1-16.
- Производители процессоров: NXP, Qualcomm, Broadcom.

Ядро MIPS

- Открытая архитектура.
- Являются частью российских процессоров "Байкал".

Linux, история

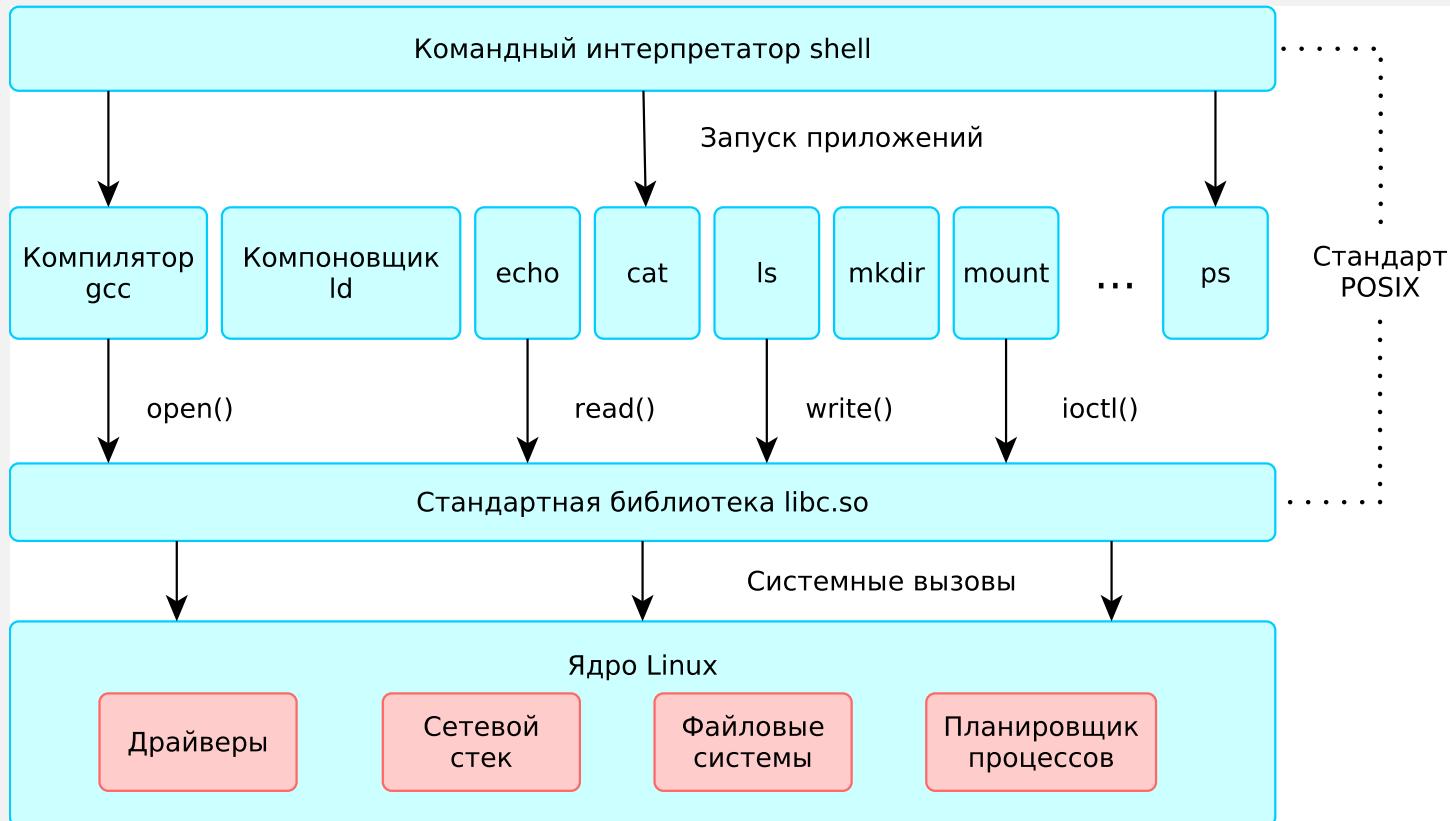


Linux, факты

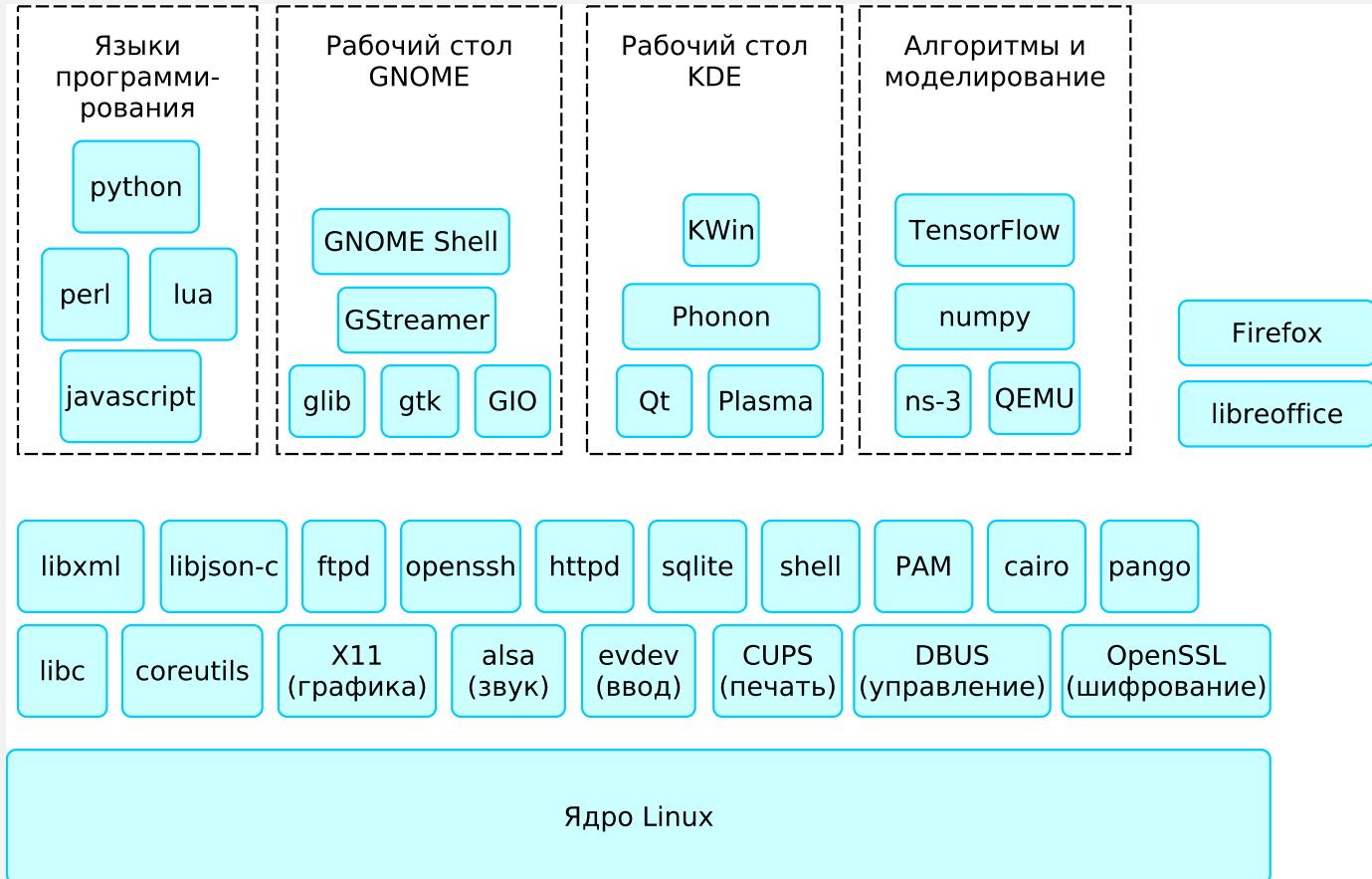


- Linux - ядро, а не полная ОС.
- Первая версия выпущена в 1991 году Линусом Торвальдсом.
- Открытый проект, распространяемый по лицензии GPLv2.
- Поддерживает 25 процессорных архитектур.
- Масштабируемость: от самых маленьких встроенных систем до суперкомпьютеров.
- Объем - около 2 миллионов строк кода.
- Оценочная стоимость разработки - 14 миллиардов долларов.

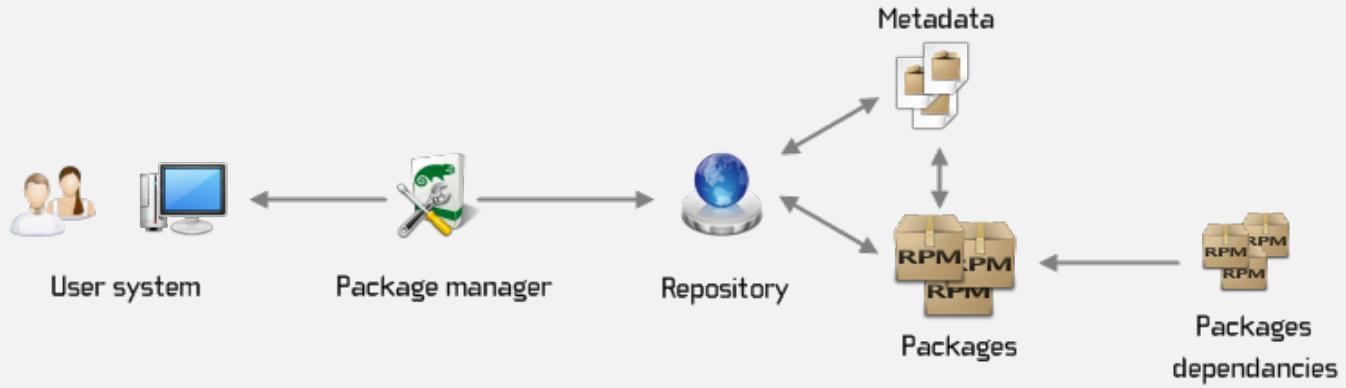
Минимальная программная среда Linux



Расширенная программная среда Linux



Пакеты и репозитории



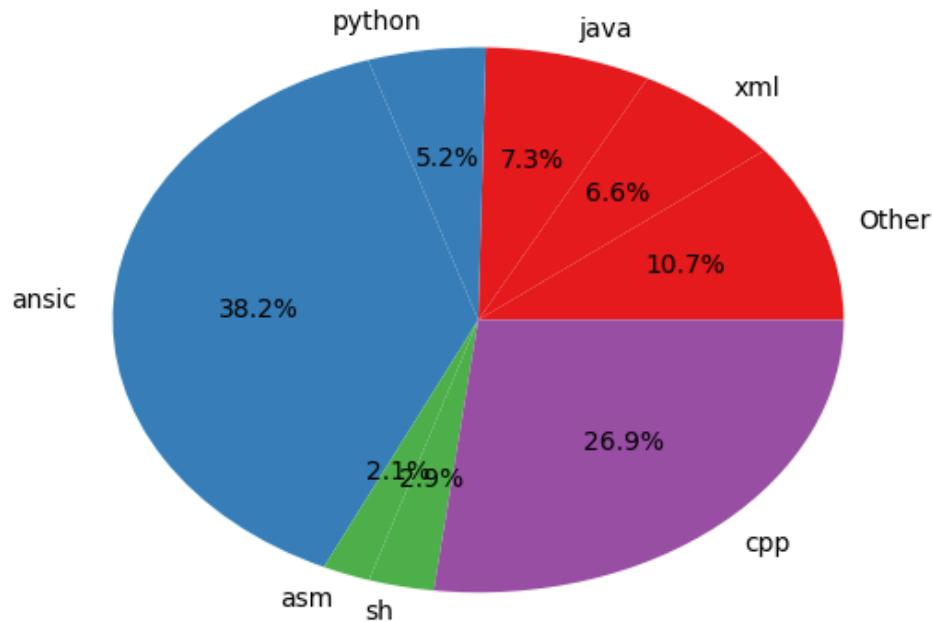
Дистрибутивы Linux

- Позволяют создать полноценную операционную систему на основе ядра Linux с помощью открытого ПО.
- Предоставляют коллекцию пакетов, оптимизированную под определенную область применения и процессорную архитектуру.
- Предоставляют средства создания и установки пакетов.
- Часто предоставляют средства конфигурирования ОС: настройка системных служб, сетевые настройки, firewall и др.
- Известные дистрибутивы: Debian, Ubuntu, Fedora, Gentoo, OpenSUSE.

Debian, статистика

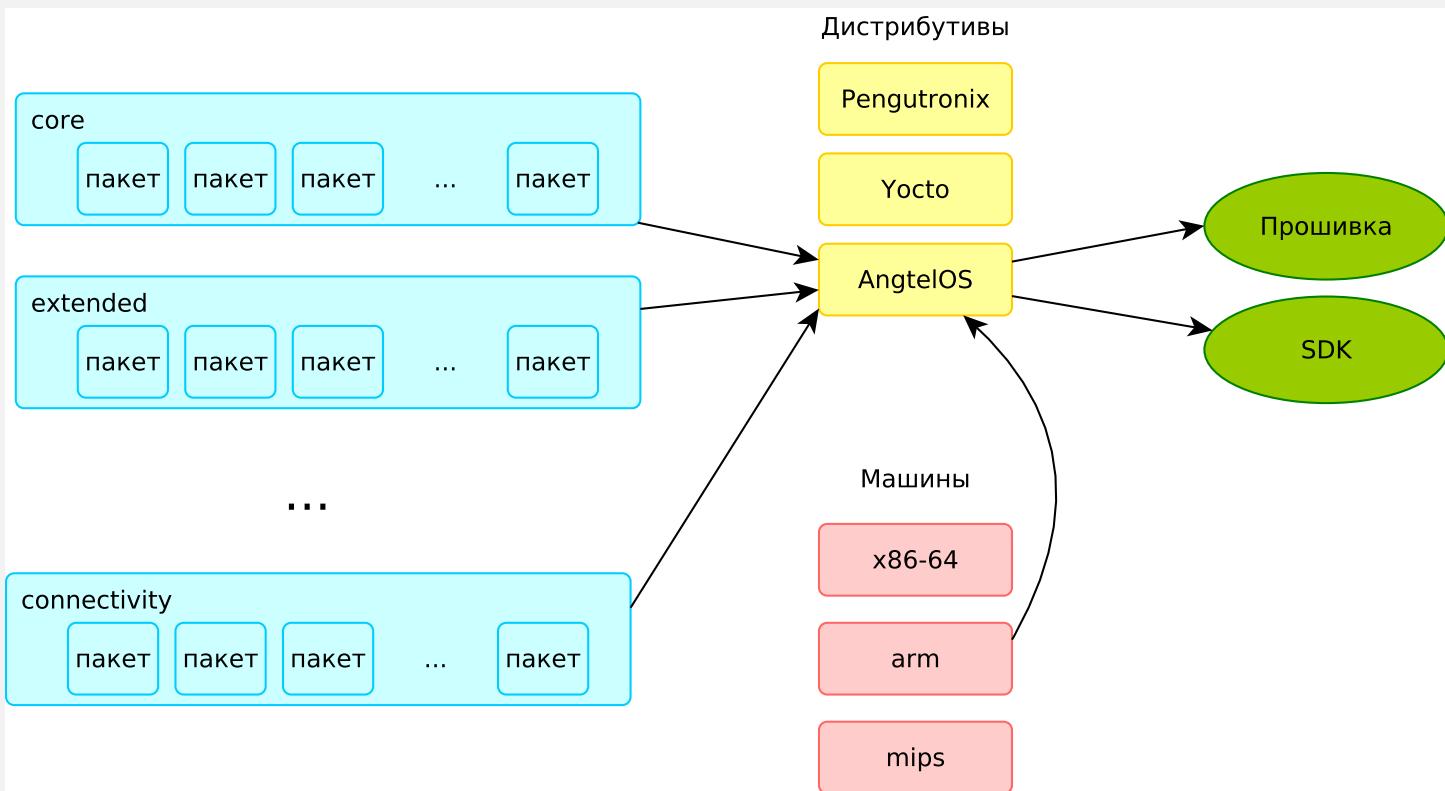
	lenny (2009)	sid (2020)
Место на диске (КБ)	63925088	332244092
Количество строк кода	351014627	1349464674
Количество пакетов	12517	31651
Количество исходных файлов	3713295	16430196

Debian, языки программирования



Other: vhdl / yacc / pascal / sed / cobol / cs / csh /
makefile / perl / fortran / ada / erlang / haskell /
f90 / awk / sql / lex / php / ruby / modula3 / lisp /
ml / objc / exp / jsp / tcl

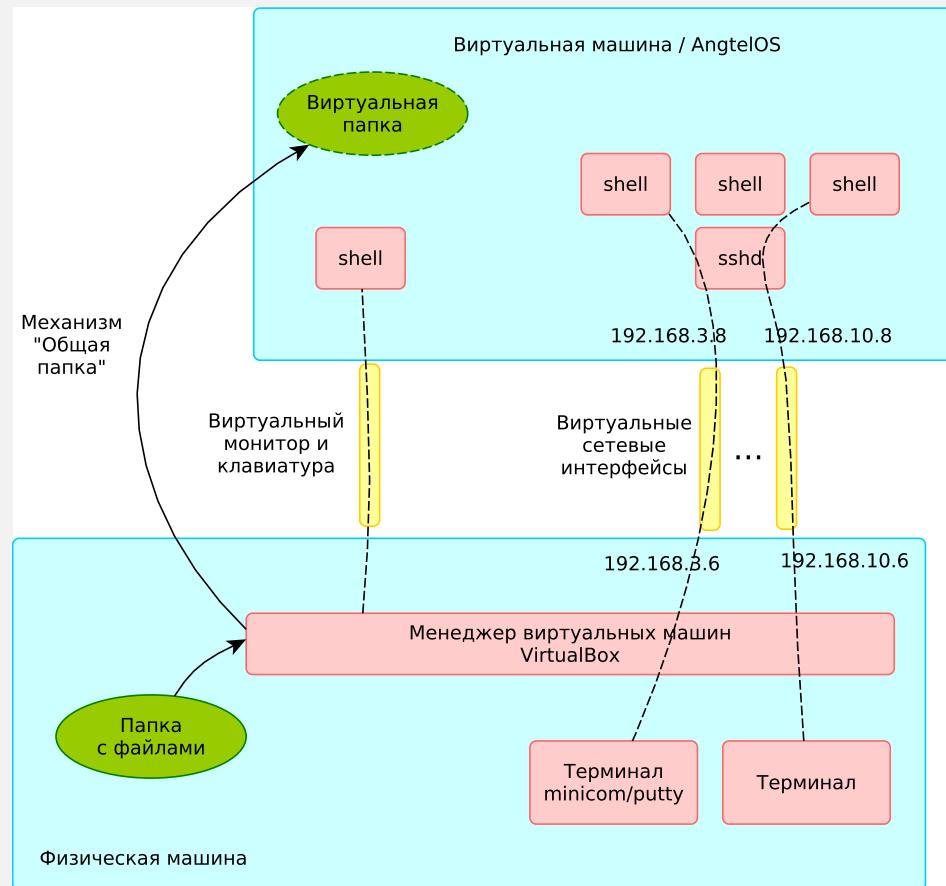
OpenEmbedded - конструктор дистрибутивов



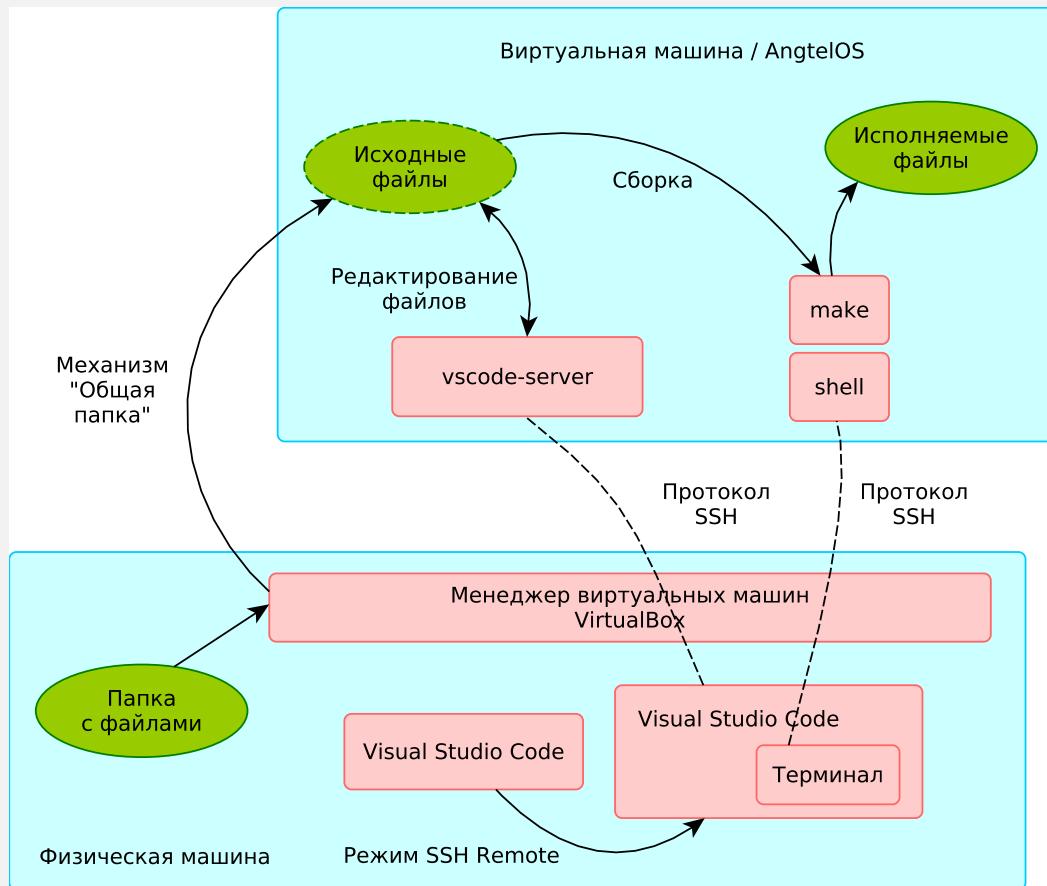
Системное программирование

- Системные утилиты
- Базы данных
- Сетевые серверы и клиенты
- Программы для потоковой обработки данных (в частности, сетевых пакетов)
- Программы управления датчиками и исполнительными устройствами
- Драйверы периферийных устройств

AngtelOS в виртуальной машине



Редактирование и сборка программ



Система сборки make

цель : зависимости
shell-команда для создания цели из зависимостей

```
lab_02: lab_02.o
        gcc -o lab_02 lab_02.o
```

```
% .o: %.c
        gcc -O2 -Wall -g -o $@ -c $<
```

```
clean:
        rm -f *.o lab_02
```

Призы

- Сертификат АО "Ангстрем-Телеком"
- Стажировка
- Отметка в дипломе (для некоторых групп)

Направления стажировки:

- разработка встроенного и управляемого ПО,
- цифровая и аналоговая схемотехника,
- разработка технических решений и взаимодействие с клиентами,
- тестирование сетевых устройств.

Краткая справка по командам shell

- `cd <directory>` - перейти в каталог `<directory>`.
- `cd ..` - перейти в родительский каталог.
- `ls` - вывести содержимое текущего каталога.
- `pwd` - вывести путь к текущему каталогу.
- `ps` - вывести список процессов.
- `make` - выполнить сборку приложения в соответствии с `Makefile`.
- `cat <file>` - вывести содержимое файла `<file>`.
- `echo <string>` - вывести строку `<string>`.

Нажатие `<Tab>` автоматически дополняет частично введенное имя команды или файла.

Повторное нажатие `<Tab>` предлагает все возможные варианты дополнения.