

# Архитектура компьютера. Операционная система

Андрей Тряпичников  
Senior Unix Engineer, ZFX



# Андрей Тряпичников

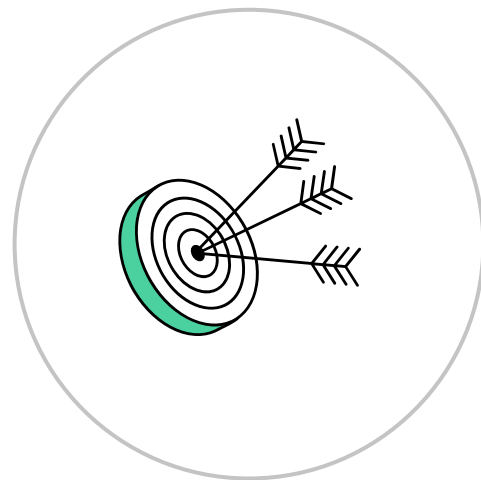
О спикере:

- Senior Unix Engineer в ZFX
- более 10 лет опыта работы системным администратором, из них 6 лет — в Одноклассниках



# Цели занятия

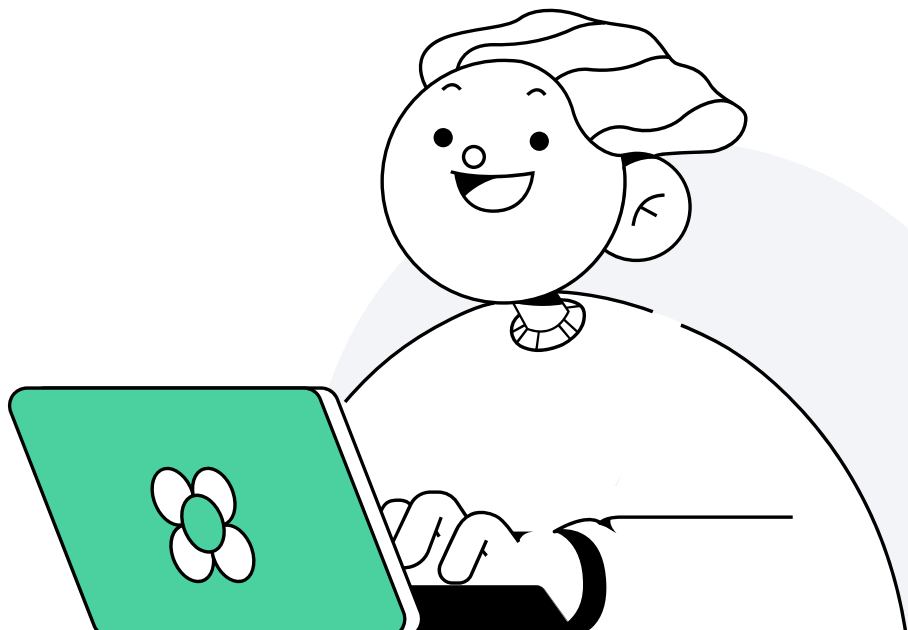
- Изучить устройство и работу компьютера
- Узнать, для чего нужны программы
- Разобраться, что такое операционная система и как она работает



# План занятия

- 1 Что такое компьютер
- 2 Что такое программа
- 3 Что такое операционная система
- 4 Как работает операционная система
- 5 Итоги

\*Нажми на нужный раздел для перехода



# Что такое компьютер



1

# Настольный компьютер



VS



## Аппаратное обеспечение (hardware)

Электронные и механические части компьютера

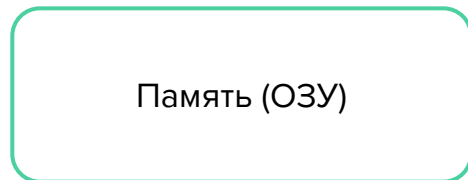
## Программное обеспечение (software)

Программы, запускаемые на компьютере

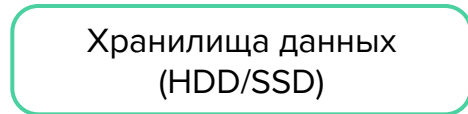
# Упрощённое устройство компьютера



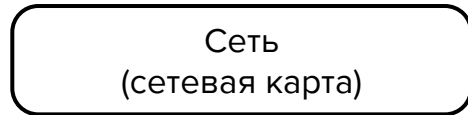
Процессор выполняет инструкции. Инструкции — это операции с данными, которые размещаются в регистрах, куда попадают из оперативного запоминающего устройства (ОЗУ). Большинство современных процессоров многоядерные, это значит, что они могут выполнять сразу несколько инструкций, независимых друг от друга.



Быстрая память для хранения оперативных данных. Чтобы данные попали в процессор, сначала им необходимо оказаться в ОЗУ. Память очищается после выключения питания.



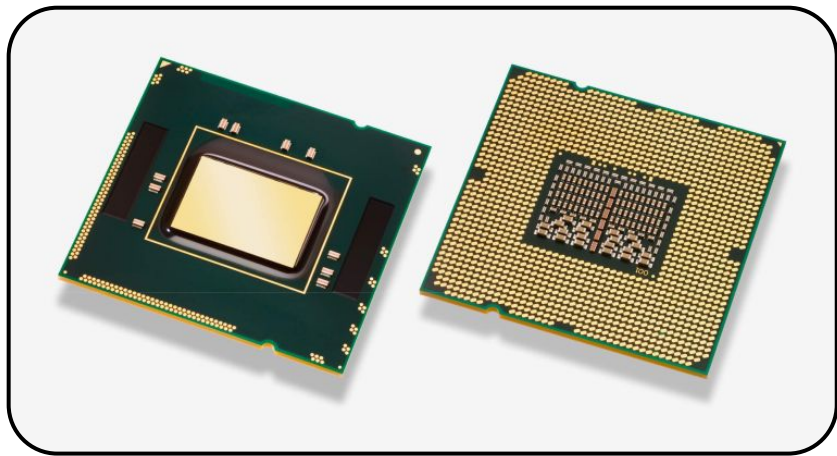
Долговременная память для хранения данных даже после выключения питания.



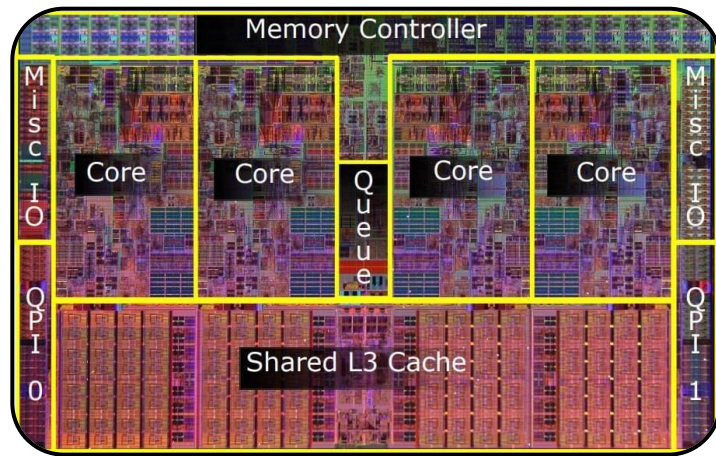
Возможность обмена информацией (передачи байтов) за пределы устройства

# Процессор

**Центральное процессорное устройство (ЦПУ, микропроцессор, central processing unit, CPU)** — компонент компьютера, предназначенный для выполнения программного кода (программы, машинного кода)



Внешний вид процессора

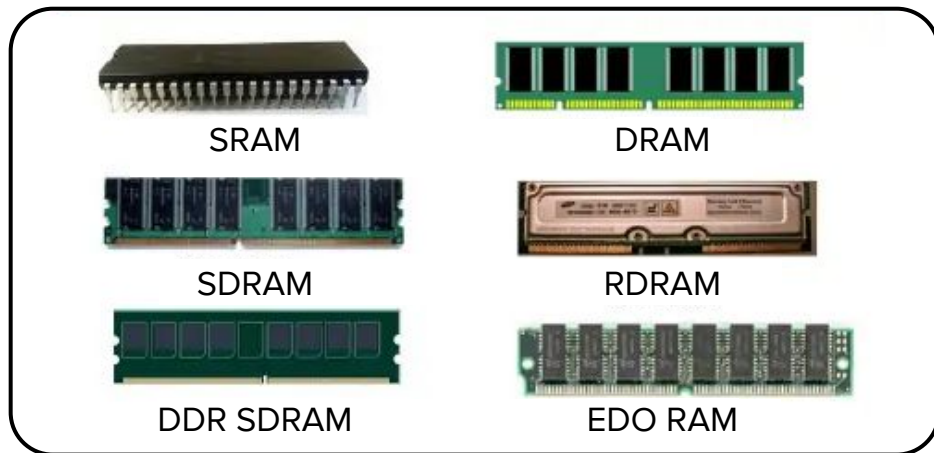


Под микроскопом



# Память

**Оперативная память (оперативное запоминающее устройство — ОЗУ; random access memory — RAM)** — устройство, в котором хранятся все **работающие в текущий момент** программы или промежуточные данные



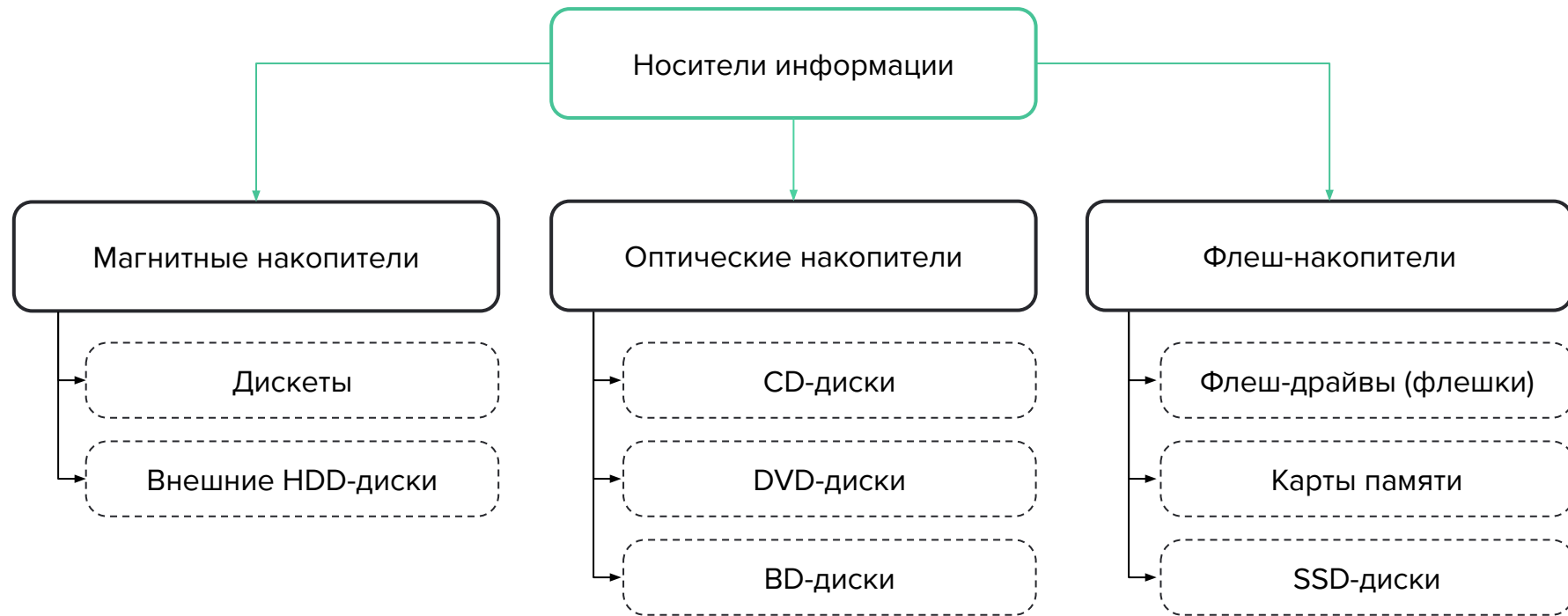
Типы оперативной памяти

# Хранилища данных

**Накопитель (запоминающее устройство, storage)** — устройство для записи и хранения данных



# Хранилища данных



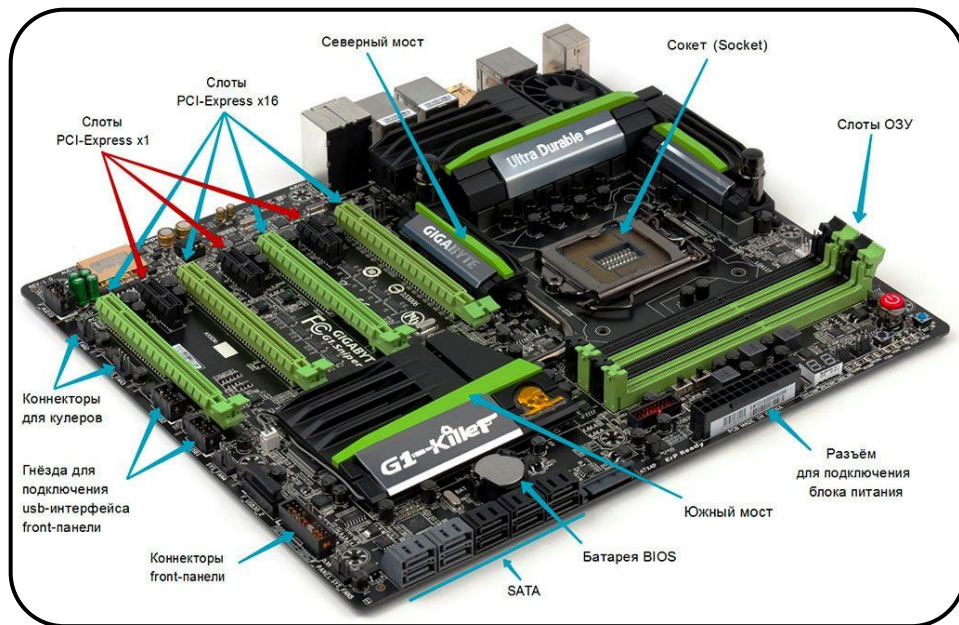
# Сеть

**Сетевая карта (NIC — network interface controller/card)** — устройство для подключения компьютера к сети



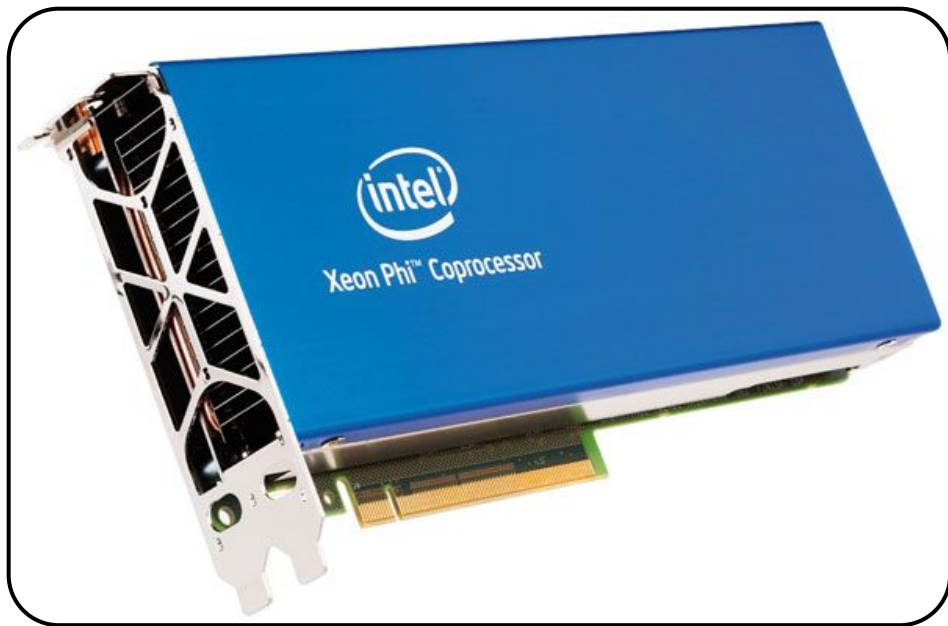
# Материнская плата

**Материнская/системная плата (motherboard)** — устройство, которое служит базой для подключения основных компонентов компьютера

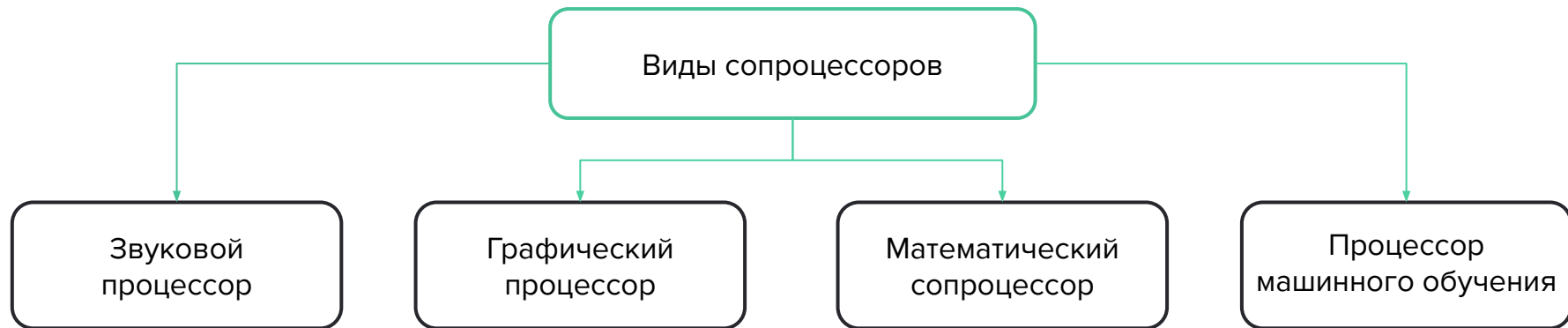


# Сопроцессор

**Сопроцессор** — специализированный процессор, дополняющий возможности центрального процессора

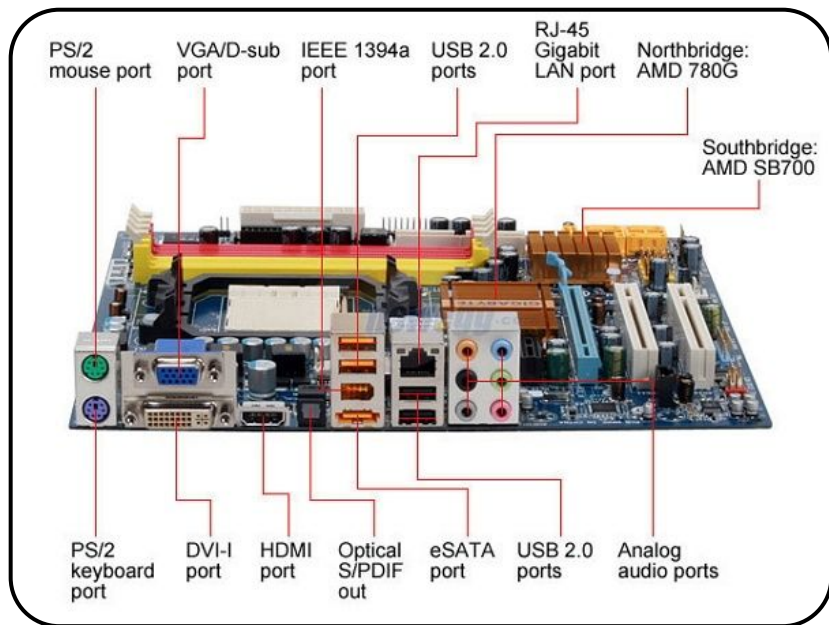


# Сопроцессор



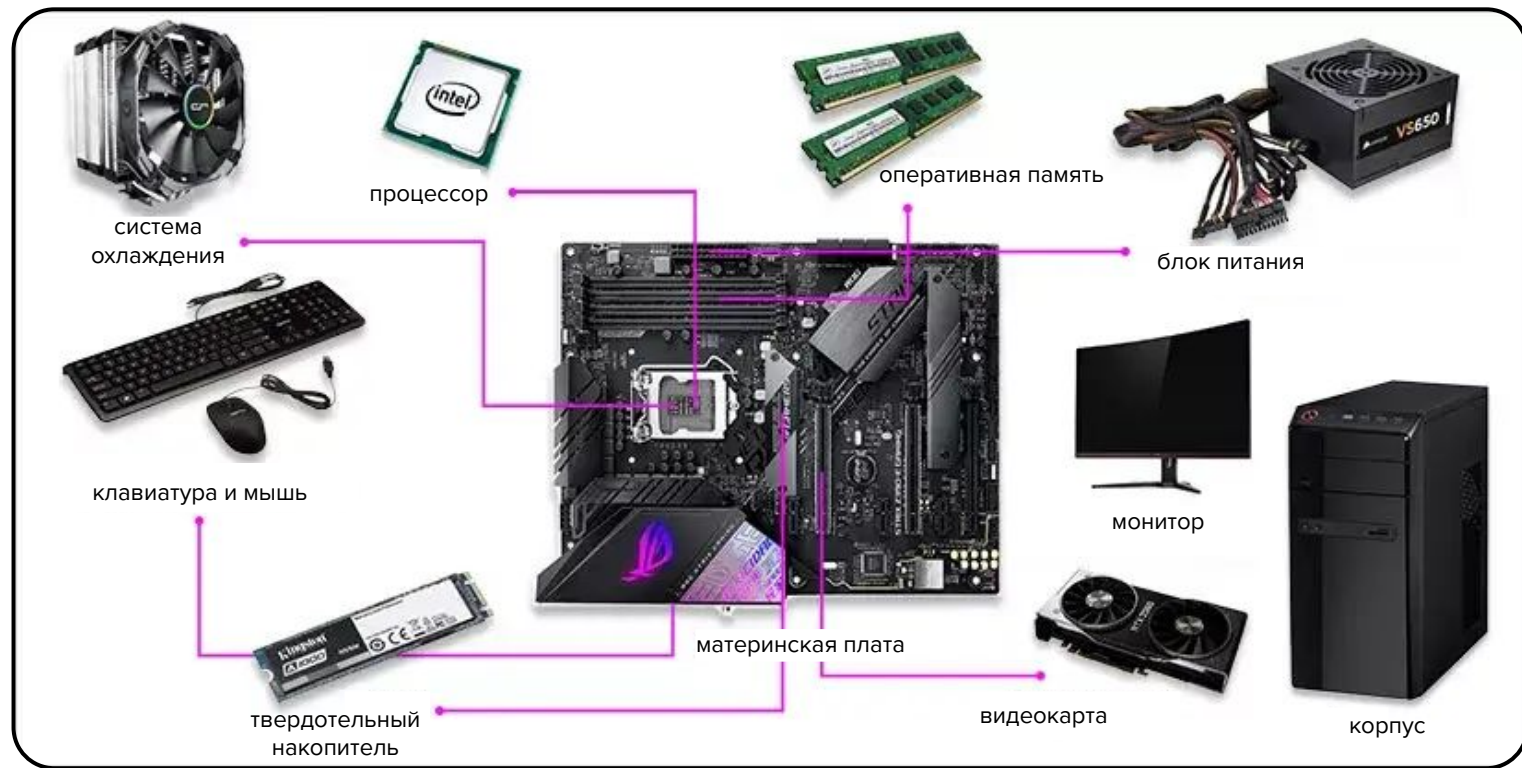
# Порты

**Порт ввода-вывода (I/O port)** — специальное решение (микросхема и разъём), позволяющее подключать внешние устройства

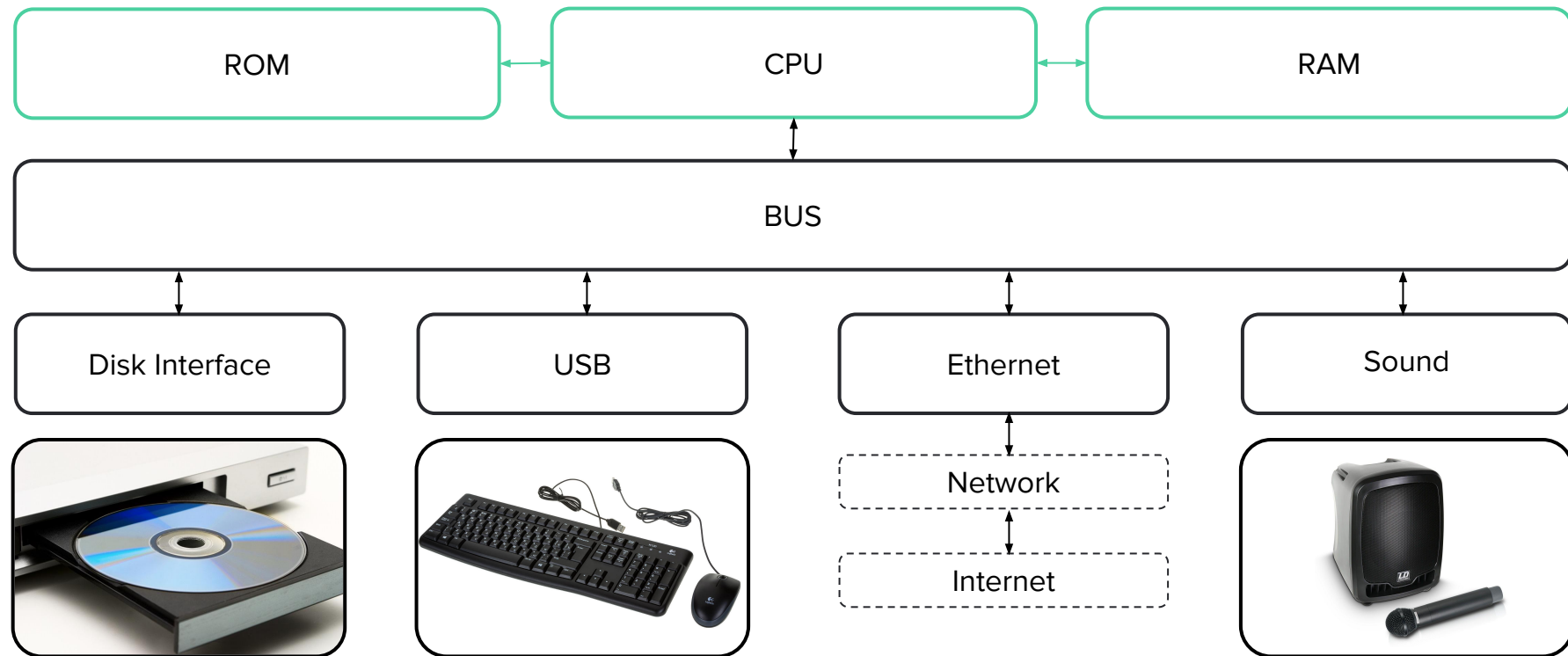




# Как всё это собирается вместе



# Как всё это работает



# Виды компьютеров

- Настольный ПК
- Неттоп
- Моноблок
- Ноутбук или нетбук
- Планшет
- Рабочая станция
- Сервер
- Встраиваемая система



# Настольный компьютер

**Настольный компьютер (десктоп, desktop computer)** — стационарный компьютер, предназначенный для работы в офисе или дома



# Неттоп

**Неттоп** — это миниатюрная версия настольного компьютера



# Моноблок

**Моноблок** — стационарный компьютер, который имеет общий корпус с монитором



# Ноутбук, ультрабук, нетбук

- **Ноутбук (notebook, laptop)** — переносной компьютер
- **Ультрабук** — тонкий и лёгкий ноутбук
- **Нетбук** — маленький ноутбук (до 10”) с ограниченной производительностью



# Планшет

**Планшет (tablet computer)** — переносной компьютер с сенсорным экраном встроенной клавиатуры.

Чаще всего работает под управлением мобильной операционной системы (ОС)





# Рабочая станция

**Рабочая станция (workstation)** — специализированный настольный компьютер.

Особенности:

- мощный процессор (Xeon)
- большой объём памяти — до 512 Гб–1 Тб
- специализированная графика
- мощный блок питания

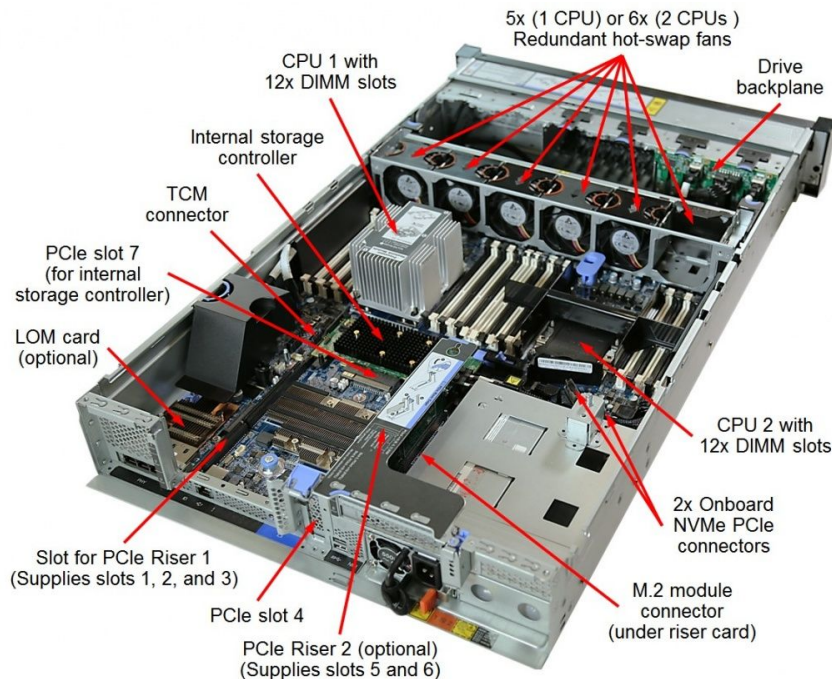


# Сервер

**Сервер (server)** — мощный компьютер, предназначенный для решения производственных задач.

Особенности:

- надёжные компоненты
- чаще всего монтируется в телекоммуникационный шкаф
- большие возможности для расширения
- наличие специализированных контроллеров



# Встраиваемые системы

**Встраиваемая система (embedded system)** — специализированное устройство, состоящее из процессора, памяти, портов ввода-вывода и выполняющее заданную функцию как **компонент другого устройства**



# Промышленный компьютер

**Промышленный компьютер (rugged computer)** — компьютер, используемый в сложных условиях эксплуатации (температура, вибрация, запылённость и т. п.)



# Что такое программа



2

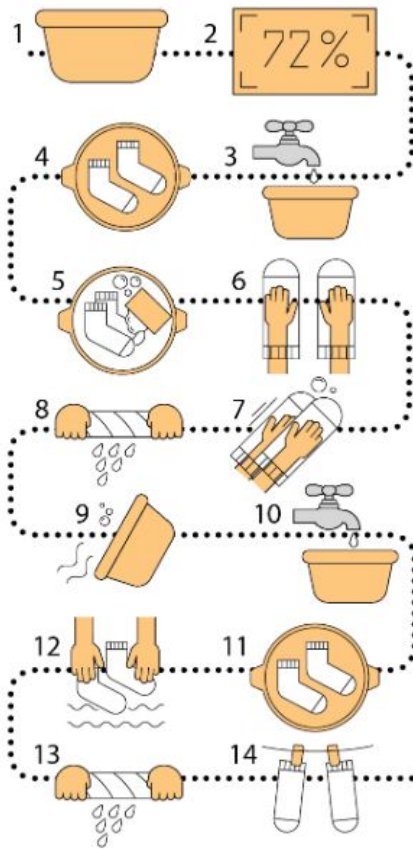
# Что такое программа

Программа — это один из способов автоматизации деятельности человека



# Зачем нужны программы

**Программа** — это набор инструкций, выполняемых компьютером для достижения определённой цели



# С чем работает процессор

Процессор умеет исполнять команды, закодированные в специальном виде — **машинном коде**.

Это набор байтов, который знает, как декодировать процессор и выполнять следующие операции:

- загрузка данных из памяти в процессор
- запись данных из процессора в память
- сложение, вычитание и т. д.

```
ooo
48 c7 c0 01 00 00 00
48 c7 c7 01 00 00 00
48 c7 c6 00 20 40 00
48 c7 c2 0d 00 00 00
0f 05
48 c7 c0 3c 00 00 00
48 c7 c7 00 00 00 00
0f 05
```



# Что такое ассемблер

Кодировать «вручную» биты и байты достаточно сложно.

С этой целью придумали специальные мнемоники, которые позволяют записывать программу в виде текста и затем переводить её в машинный код с помощью особой программы — **ассемблера**.

Соответствующие языки называют **языками ассемблера**



# Так выглядит работа ассемблера

ooo

```
48 c7 c0 01 00 00 00
48 c7 c7 01 00 00 00
48 c7 c6 00 20 40 00
48 c7 c2 0d 00 00 00
0f 05
48 c7 c0 3c 00 00 00
48 c7 c7 00 00 00 00
0f 05
```

ooo

```
mov    $0x1,%rax
mov    $0x1,%rdi
mov    $0x402000,%rsi
mov    $0xd,%rdx
syscall
mov    $0x3c,%rax
mov    $0x0,%rdi
syscall
```

# Высокоуровневые языки

Писать на языке ассемблера большие программы проблематично, поэтому появились высокоуровневые языки, позволяющие не думать об особенностях процессора.

К ним относятся Pascal, Fortran, C++, Python и другие.

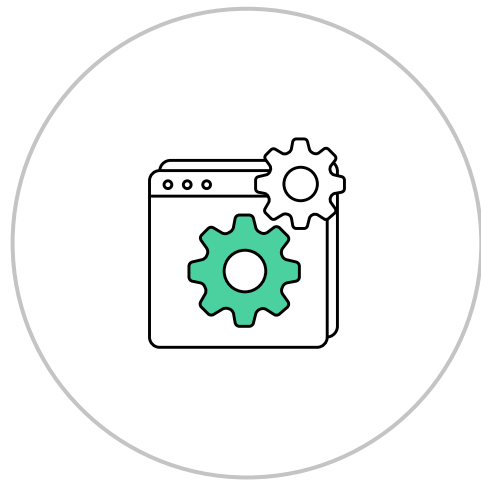
**Пример на языке C++:**

```
int i1, i2;  
bool res;  
i1 = 1 + 2 + 3;
```

# Компилятор

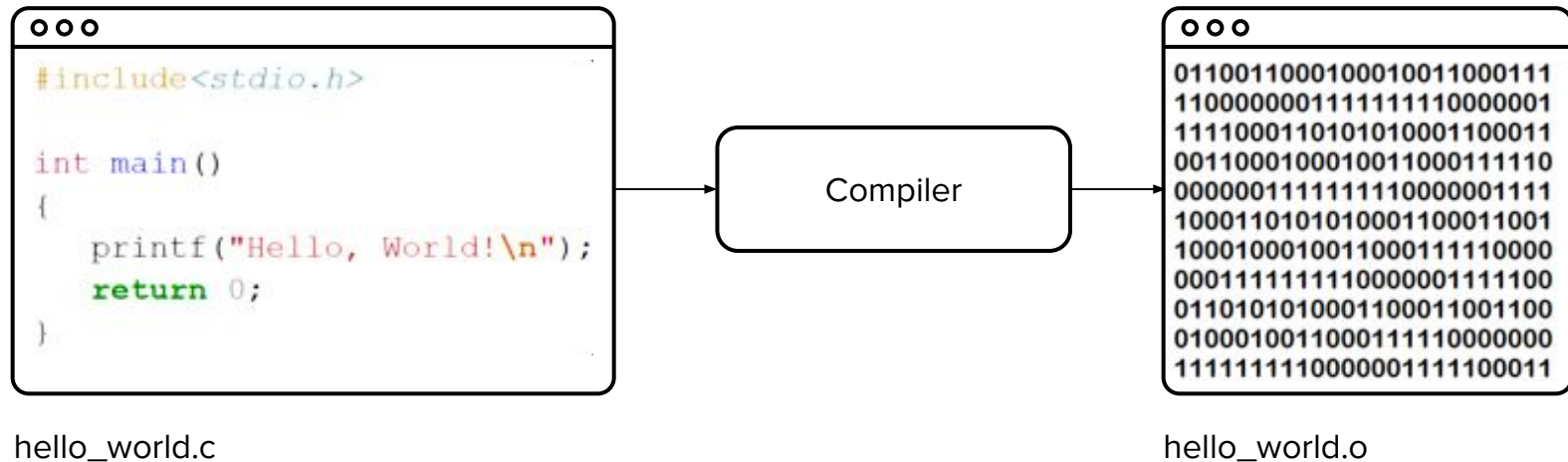
**Компиляция** — процесс перевода исходного текста программы, написанного программистом, в машинный код.

Компиляция предполагает создание исполняемого файла из программы

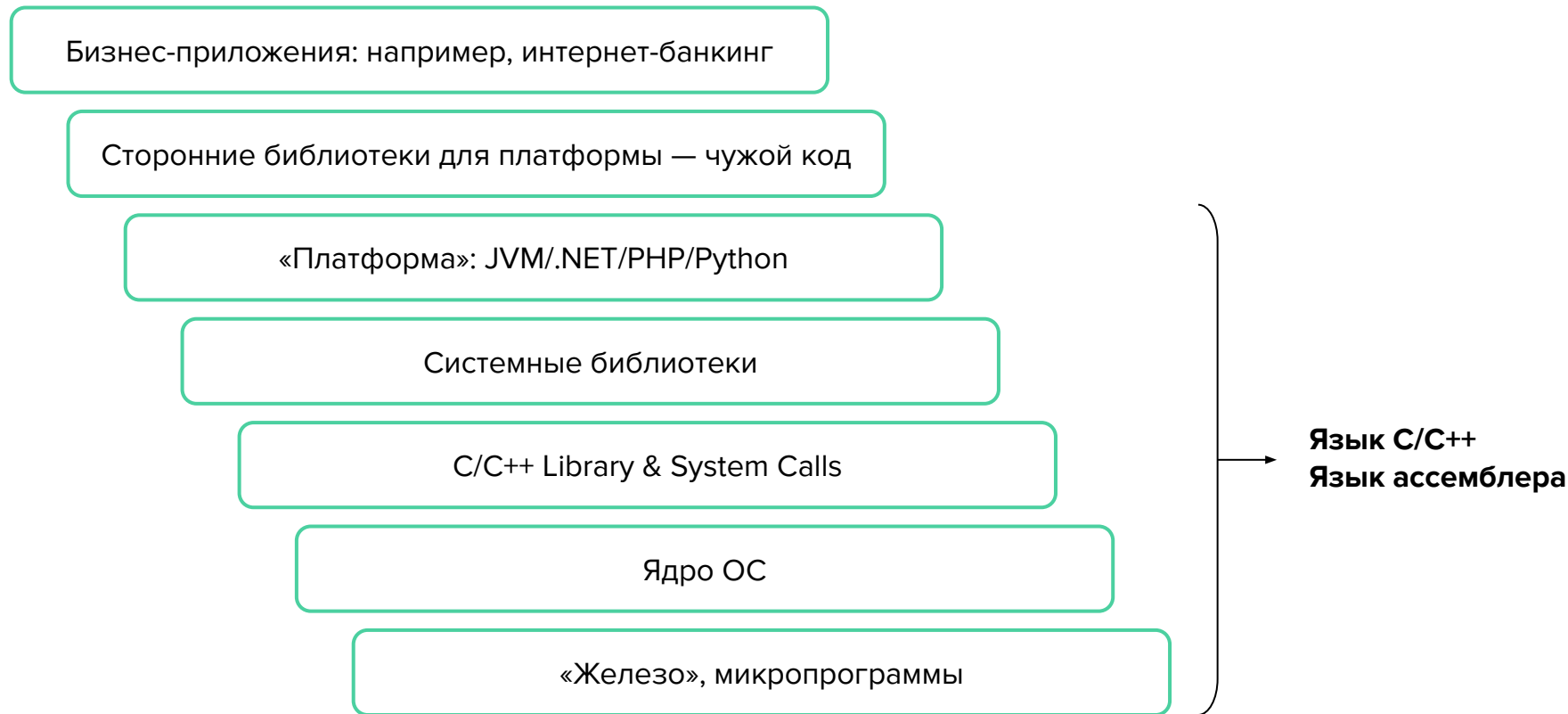


# Компилятор

Программа, выполняющая процесс компиляции, называется **компилятором (compiler)**



# Иерархия программ в компьютере



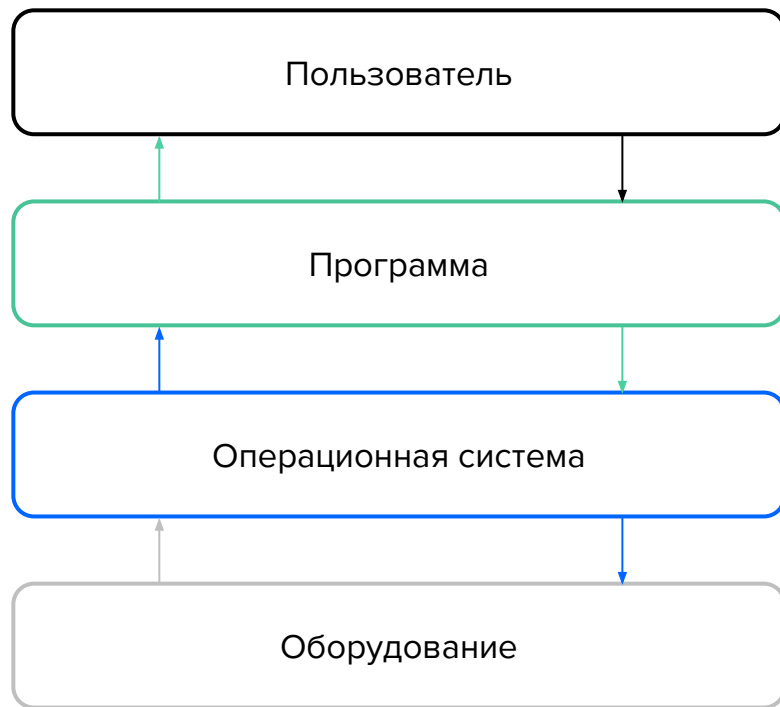
# Что такое операционная система



3

# Операционная система

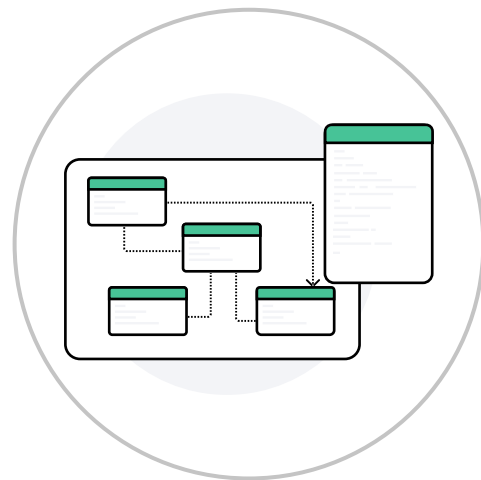
**Операционная система (operation system)** — специальная программа, управляющая ресурсами компьютера и предоставляющая остальным программам ряд базовых механизмов





# Функции операционной системы

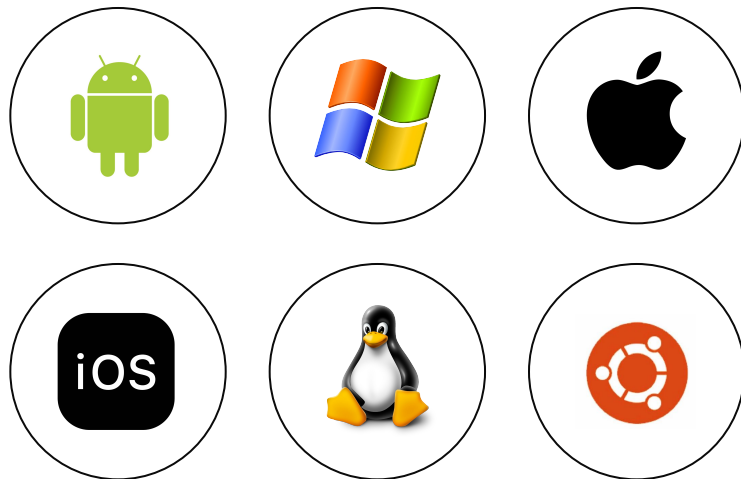
- Запуск программ на компьютере
- Управление всеми подключёнными устройствами
- Распределение памяти, устройств и других ресурсов между работающими программами
- Управление безопасностью всей системы
- Организация взаимодействия с пользователем — пользовательский интерфейс
- Предоставление диагностической информации о работе компьютера и программ



# Виды операционных системы

Популярность операционных систем на пользовательских устройствах, по данным на ноябрь 2022 года, согласно [statcounter.com](https://statcounter.com):

- **Android** — 43,43%
- **Windows** — 29,3%
- **iOS** — 17,22%
- **OS X** — 6,08%
- **Linux** — 1,09%
- **другие** — 1,64%



# Android

**Android** — открытая ОС от компании Google для мобильных устройств.

Особенности:

- бесплатная ОС
- установлена на большинстве моделей мобильных устройств
- удобный графический интерфейс

android 

# Windows

**Windows** — операционная система компании Microsoft.

Особенности:

- платная ОС
- множество программ для пользователя
- удобный графический интерфейс
- поддержка почти любого оборудования
- под Windows создано более 95% игр



# iOS

**iOS** — операционная система для мобильных устройств компании Apple. Устанавливается на все мобильные устройства компании.

Особенности:

- закрытая ОС
- устанавливается только на iPhone
- частые обновления ОС
- только один магазин приложений — App Store
- в магазине более миллиона приложений



# OS X

**OS X** — операционная система для компьютеров и ноутбуков компании Apple. Устанавливается на все компьютеры компании, кроме мобильных устройств.

Особенности:

- закрытая ОС
- устанавливается только на Mac
- частые обновления ОС
- только один магазин приложений — App Store
- большинство приложений платные
- мало популярных игр :)



# Linux

**Linux** — бесплатная ОС с открытым кодом для компьютеров, ноутбуков и ряда устройств.

Особенности:

- открытая ОС
- настройка ОС требует опыта
- не поддерживается многими производителями коммерческого ПО
- поддерживается не всё «железо»



# Серверные ОС

**Серверная ОС** — операционная система, предоставляющая дополнительные возможности для управления инфраструктурой компании:

Особенности:

- Windows Server
- Red Hat Enterprise
- Ubuntu Server
- FreeBSD



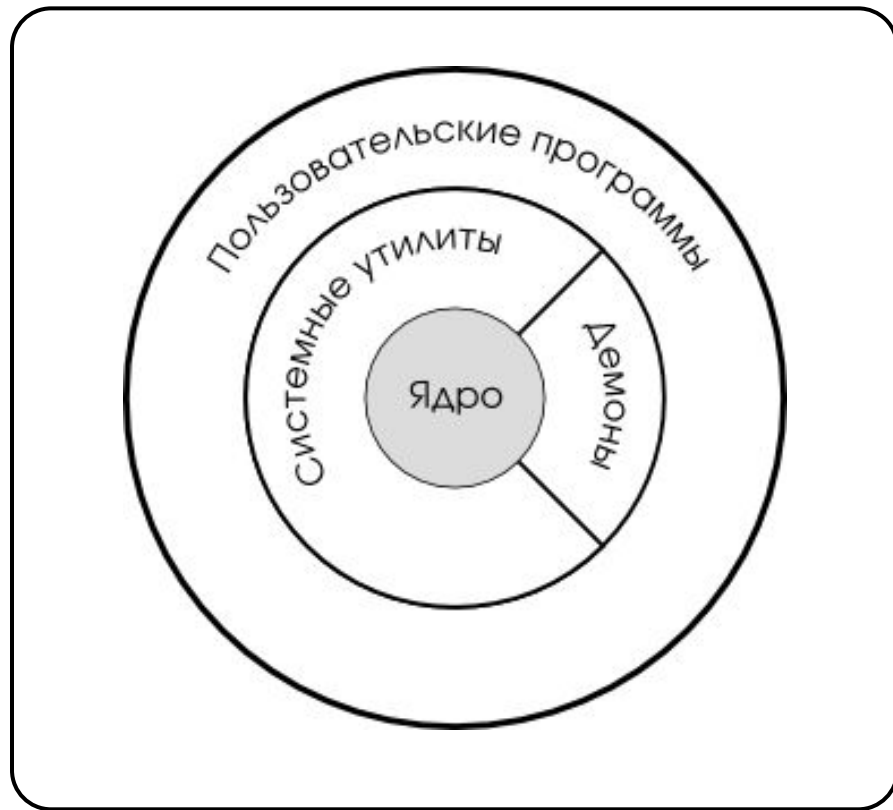
# Как работает операционная система



4

# Архитектура операционной системы

- **Ядро** — самая важная часть ОС, реализующая доступ к ресурсам и управление программами (процессами)
- **Службы (демоны)** — программы, запускаемые ОС, работающие в фоновом режиме без взаимодействия с пользователем
- **Системные утилиты** — программы, с помощью которых возможно проводить дополнительные настройки и оптимизацию ОС
- **Пользовательские программы** — программы, запускаемые пользователем



# Важные элементы операционной системы

- Пользователи
- Файлы
- Каталоги
- Ссылки
- Устройства и драйверы
- Графический интерфейс
- Консоль



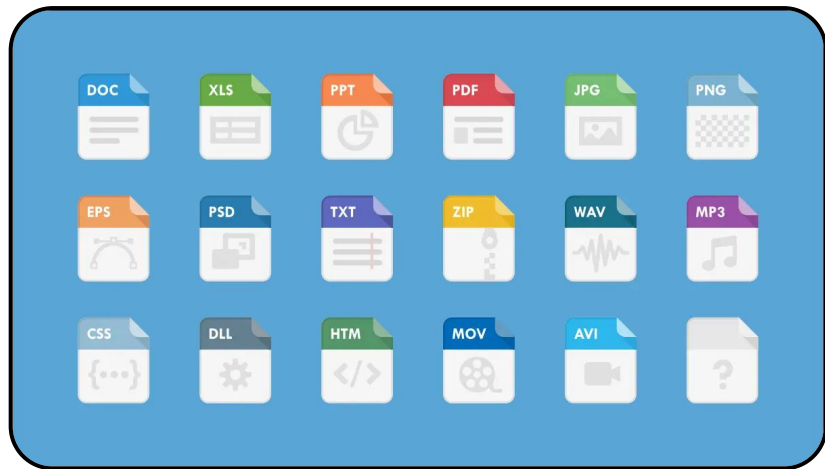
# Пользователи

- **Пользователь (user)** — лицо, которое использует действующую систему для выполнения конкретной функции или получения определённого результата
- **Учётная запись (account)** — данные о пользователе, хранящиеся в информационной системе



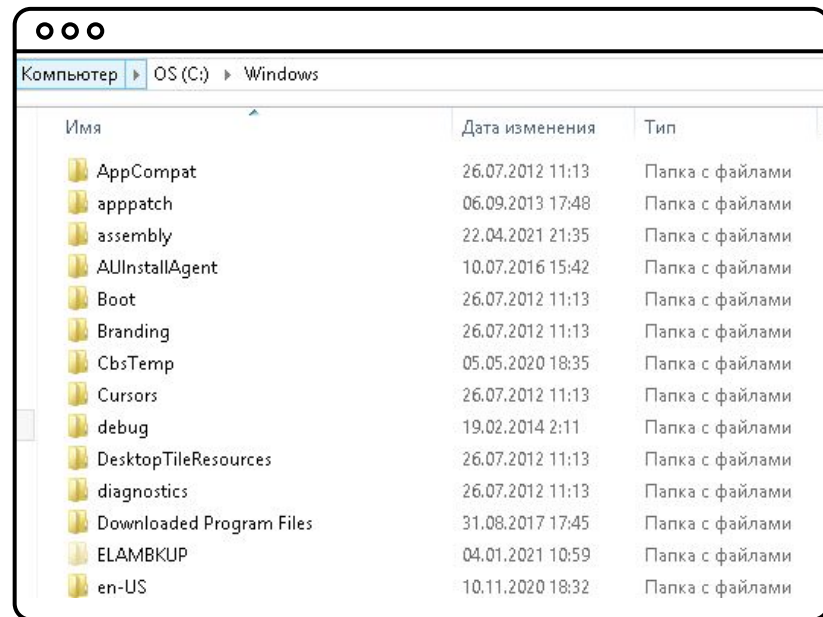
# Пользователи

- **Файл (file)** — область постоянной памяти, в которой хранятся данные и с которой ассоциирован определённый путь внутри файловой иерархии ОС
- **Формат файла** — способ записи информации в файл:  
что и как будет записано в файле



# Каталоги

**Каталог (directory, folder, директория, папка)** — специальный объект (чаще всего файл), упрощающий организацию файлов на компьютере

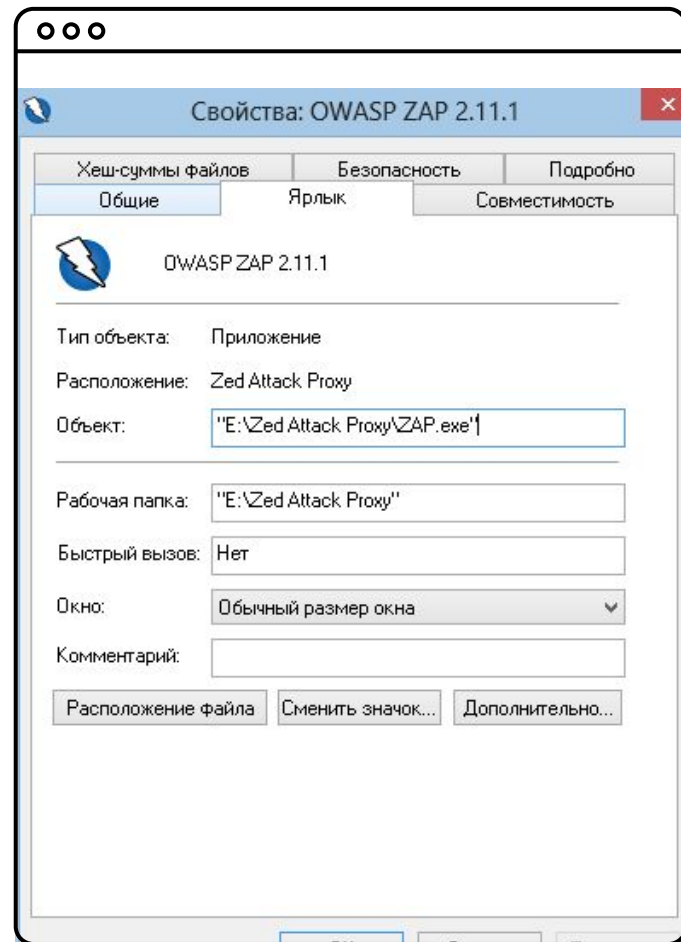


Имя	Дата изменения	Тип
AppCompat	26.07.2012 11:13	Папка с файлами
apppatch	06.09.2013 17:48	Папка с файлами
assembly	22.04.2021 21:35	Папка с файлами
AUIInstallAgent	10.07.2016 15:42	Папка с файлами
Boot	26.07.2012 11:13	Папка с файлами
Branding	26.07.2012 11:13	Папка с файлами
CbsTemp	05.05.2020 18:35	Папка с файлами
Cursors	26.07.2012 11:13	Папка с файлами
debug	19.02.2014 2:11	Папка с файлами
DesktopTileResources	26.07.2012 11:13	Папка с файлами
diagnostics	26.07.2012 11:13	Папка с файлами
Downloaded Program Files	31.08.2017 17:45	Папка с файлами
ELAMBKUP	04.01.2021 10:59	Папка с файлами
en-US	10.11.2020 18:32	Папка с файлами

# Ссылки

**Ссылка (link, shortcut, ярлык)** — специальный файл, в котором содержится путь к другому файлу или каталогу.

Знакомый аналог — ярлык .lnk в Windows. Выполняет схожую функцию, хоть и не является ссылкой в строгом понимании этого термина

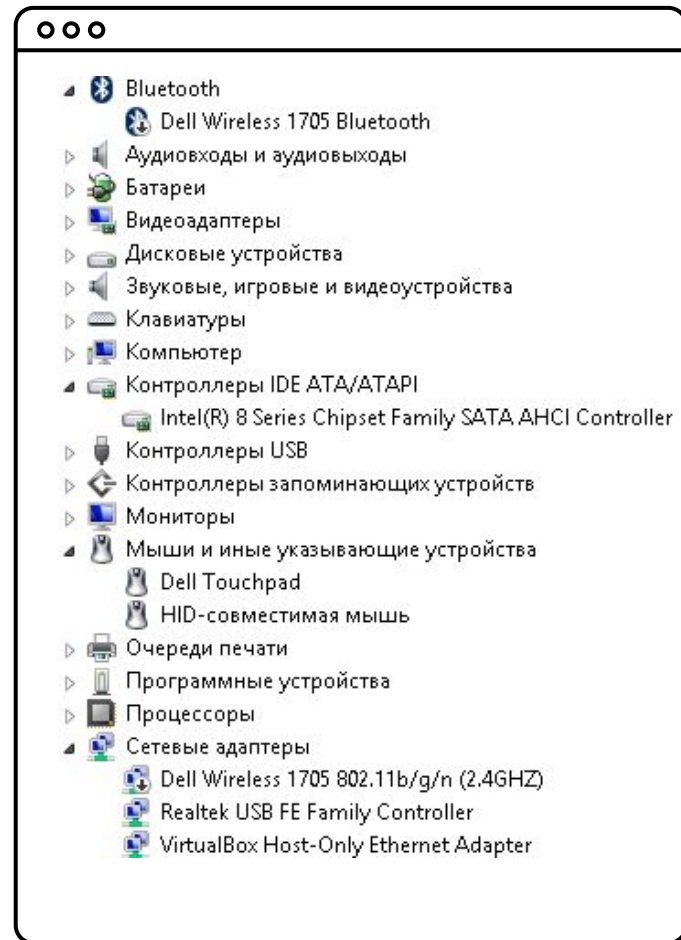


# Устройства и драйверы

## Устройство (device):

- электронный компонент компьютера
- специальный набор данных, помогающий ОС работать с аппаратными средствами компьютера

**Драйвер (driver)** — программа, с помощью которой ОС работает с указанным устройством





# Графический интерфейс

**Графический интерфейс пользователя (graphical user interface, GUI)** — система для взаимодействия пользователя с компьютером, основанная на использовании графических элементов



# Элементы GUI

- Окна
- Кнопки
- Меню
- Списки выбора
- Значки
- Рисунки и т. д.



# Текстовый интерфейс

Командная строка (command line) — средство для взаимодействия пользователя с компьютером, которое основано на использовании текстовой информации

```
ooo
load averages: 0.45, 0.39, 0.37
20 processes: 0% idle, 1 on processor
CPU states: 0.0% user, 0.0% nice, 0.0% system, 0.0% interrupt, 100% idle
CPU states: 0.0% user, 0.0% nice, 0.0% system, 0.0% interrupt, 100% idle
memory: Real: 1081204K act/tot, Free: 1660K Swap: 0/20551 used/tot

PID USER% PRI NI% SZ% RES STATE WAIT TIME CPU COMMAND
26389 nicholas 2 0 1776K 4700K sleep/1 poll 0:00 0.00% md
16366 nicholas 2 0 1520K 4556K sleep/1 poll 1:34 0.00% md
23200 nicholas 2 0 4172K 2944K sleep/0 poll 0:00 0.00% md
2790 nicholas 2 0 336K 1952K sleep/1 poll 0:00 0.00% scmpc
12060 root 2 0 456K 796K sleep/0 kqread 0:00 0.00% apnd
7401 uuu 2 0 1540K 2540K sleep/1 select 0:00 0.00% httpd
10926 root 2 0 1124K 2104K sleep/1 poll 0:01 0.00% logfnon
8064 root 2 0 1844K 1168K sleep/1 poll 0:01 0.00% nfsd
13182 nicholas 2 0 338K 228K sleep/0 select 0:02 0.00% shd
1680 root 2 0 148K 144K idle nfsd 0:02 0.00% nfsd
26590 root 2 0 148K 144K idle nfsd 0:01 0.00% nfsd
76 nicholas 2 0 135K 212K sleep/0 poll 0:00 0.00% taur
20891 root 2 0 612K 952K idle select 0:00 0.00% cron
16340 nicholas 3 0 620K 620K idle ttyin 0:00 0.00% ksh
13971 syslogd 2 0 620K 840K sleep/0 poll 0:00 0.00% syslogd
19661 nicholas 2 0 972K 2704K sleep/1 poll 0:00 0.00% ncmpc
27153 nicholas 2 0 1500K 10K sleep/0 select 0:00 0.00% emacs

client_msg_fn_detach(struct hdr *hdr, struct client_ctx *cctx)
client_msg_fn_detach(struct msg *msg, struct client_ctx *cctx)
{
    if (hdr->size != 0)
        if (msg->hdr.len != MSG_HEADER_SIZE)
            fatalx("bad MSG_DETACH size");
    client_write_server(cctx, MSG_EXITING, NULL, 0);
    _96,9 +107,9 @@
int
client_msg_fn_shutdown(
    struct hdr *hdr, struct client_ctx *cctx)
    struct msg *msg, struct client_ctx *cctx)
{
    if (hdr->size != 0)
        if (msg->hdr.len != MSG_HEADER_SIZE)
            fatalx("bad MSG_SHUTDOWN size");
    client_write_server(cctx, MSG_EXITING, NULL, 0);
    _100,9 +119,9 @@
msg->diff = 1; taur-msg-120diff.diff 17% (134.0) Hg-0 (diff)=====
msg->diff = 1; taur-msg-120diff.diff 17% (134.0) Hg-0 (diff)=====
```

# Итоги занятия

Сегодня мы:

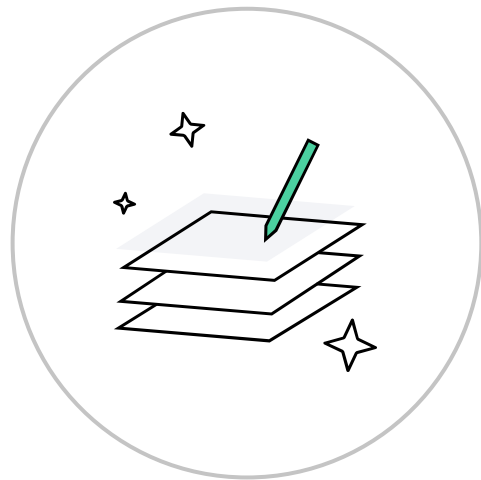
- 1 Изучили устройство компьютера
- 2 Выяснили, для чего нужны программы
- 3 Узнали, что такое операционная система и как она работает



# Домашнее задание

Давайте посмотрим ваше домашнее задание.

- 1 Вопросы по домашней работе задавайте в чате группы
- 2 Задачи можно сдавать по частям
- 3 Зачёт по домашней работе ставят после того, как приняты все задачи



# Задавайте вопросы и пишите отзыв о лекции

Андрей Тряпичников  
Senior Unix Engineer, ZFX

