

# Архитектура вычислительных систем

Введение в архитектуру вычислительных систем

Романюта Алексей Андреевич

[alexey-r.98@yandex.ru](mailto:alexey-r.98@yandex.ru)

Кафедра вычислительных систем  
Сибирский государственный университет телекоммуникаций и информатики



# Правила игры

## Содержание курса

- ~16-17 лекций
- 5 лабораторных работ
- РГР
- Экзамен
- Автоматы?

# Правила игры

- Посещаемость – желательно

Посещаемость практики обсуждается с преподавателем практики

- Контрольный срок – оценивается по кол-ву сданных лаб и оценке за них
- Автомат – зависит от сданных лабораторных, контрольного срока и оценке за РГР.

Необходимое условие – наличие сданных лабораторных и РГР

- Вопросы по лекциям и практикам задавать можно и *нужно*.
- Лекции состоят из двух частей – обзор темы, затем live section по вопросам

# Программа курса

- Немного о базовых понятиях ВС и где это можно встретить в мире
- Linux, процессы
- Контейнеризация, docker
- Управление вычислительными системами (или «парком» серверов)
- Оркестрация Kubernetes, docker compose
- Распределенные приложения и балансировка нагрузки
- Мониторинг систем и приложений
- Характеристики и модели ВС

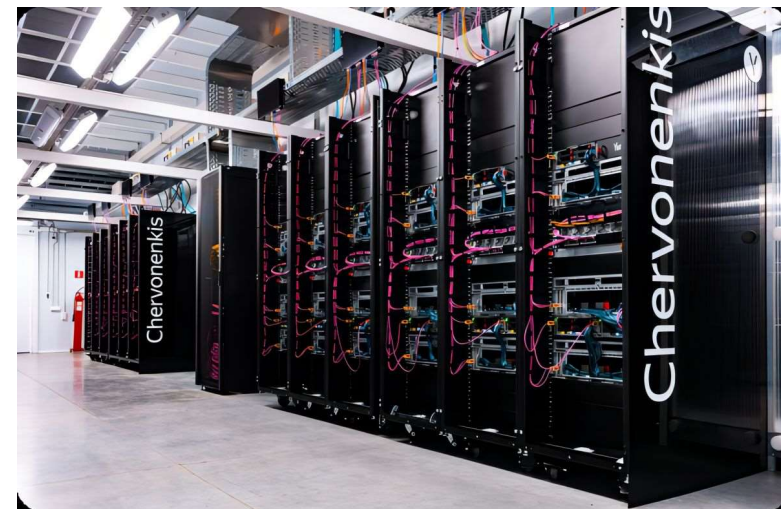
# О вычислительных системах

- **Вычислительная система** – средство обработки информации, базирующееся на модели коллектива вычислителей, т.е. на структурной и функциональной имитации деятельности коллектива людей-вычислителей
- **Вычислительная система** – совокупность взаимосвязанных и одновременно функционирующих аппаратурно-программных вычислителей, которая способна не только реализовать (параллельный) процесс решения сложной задачи, но и априори и в процессе работы автоматически настраиваться и перестраиваться с целью достижения адекватности между своей структурно-функциональной организацией и структурой и характеристиками решаемой задачи.

# О вычислительных системах

- Как правило в их составе больше одного ПК/сервера
- Параллельное исполнение различных программ и обработка событий.  
Моделирование различных физических событий. Решение задач линейной алгебры
- Распределенные данные, файловые системы, распределенные базы данных
- MPI, Hadoop, Spark

# Развитие вычислительных систем



# Примеры вычислительных систем

- Червоненкинс – Яндекс, 199 узлов, 398 cпу (2x AMD EPYC 7702 , 1024 GB RAM; Асс: 8x NVIDIA A100), 29.4 Пфлоп/с
- Галушкин – Яндекс, 136 узлов, 272 cпу (2x AMD EPYC 7702 , 1024 GB RAM; Асс: 8x NVIDIA A100), 20.6 Пфлоп/с
- Кристофари Нео – SberCloud 99 узлов, 198 cпу (2x AMD EPYC 7742, 2048 GB RAM; Асс: 8x NVIDIA A100), 14.9 Пфлоп/с



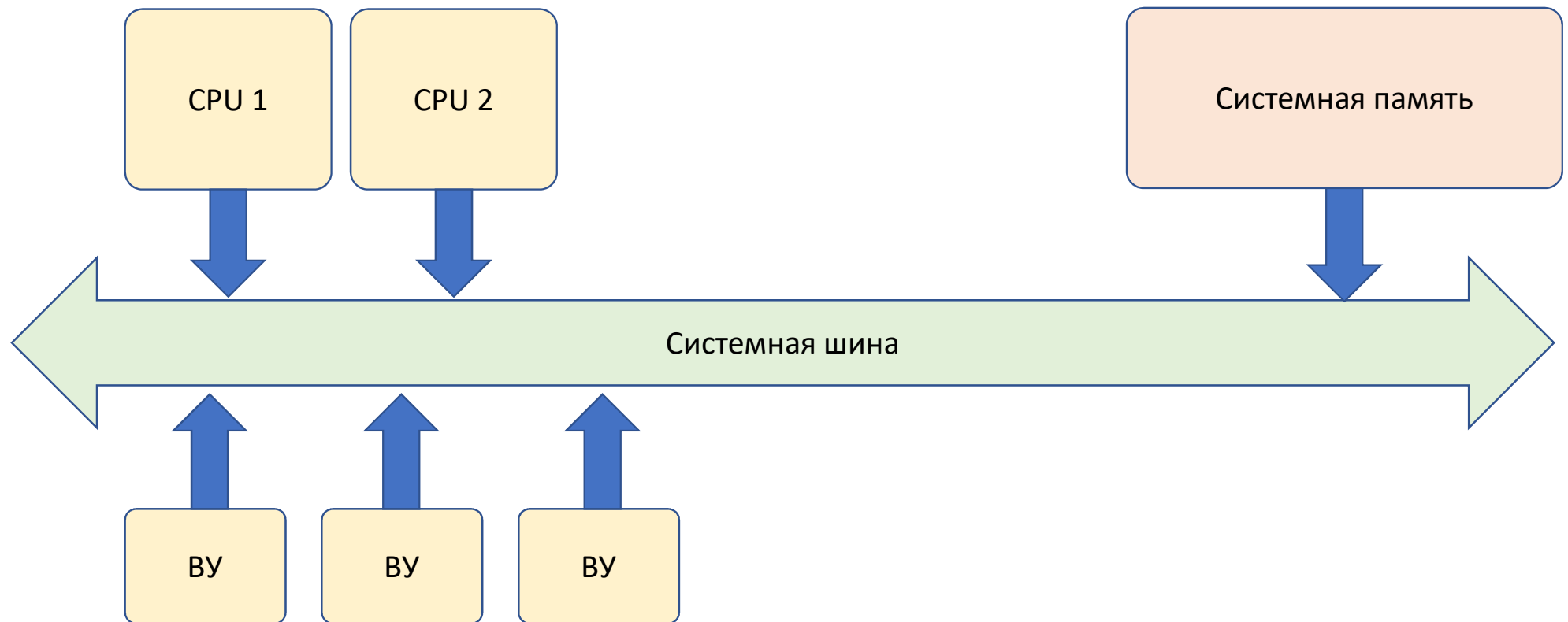
# Классификация вычислительных систем

Классификация Флинна		
	Один поток инструкций	Множество потоков инструкций
Один поток данных	SISD	MISD
Множество потоков данных	SIMD	MIMD

- SISD – последовательный процессор
- SIMD – выполнение одной инструкции над данными: векторные ВС Cray, NEC; наборы векторных инструкций процессора AVX; GPU
- MISD – конвейерные ВС
- MIMD – вычислительные кластеры, MPP-системы (Massive Parallel Processing)

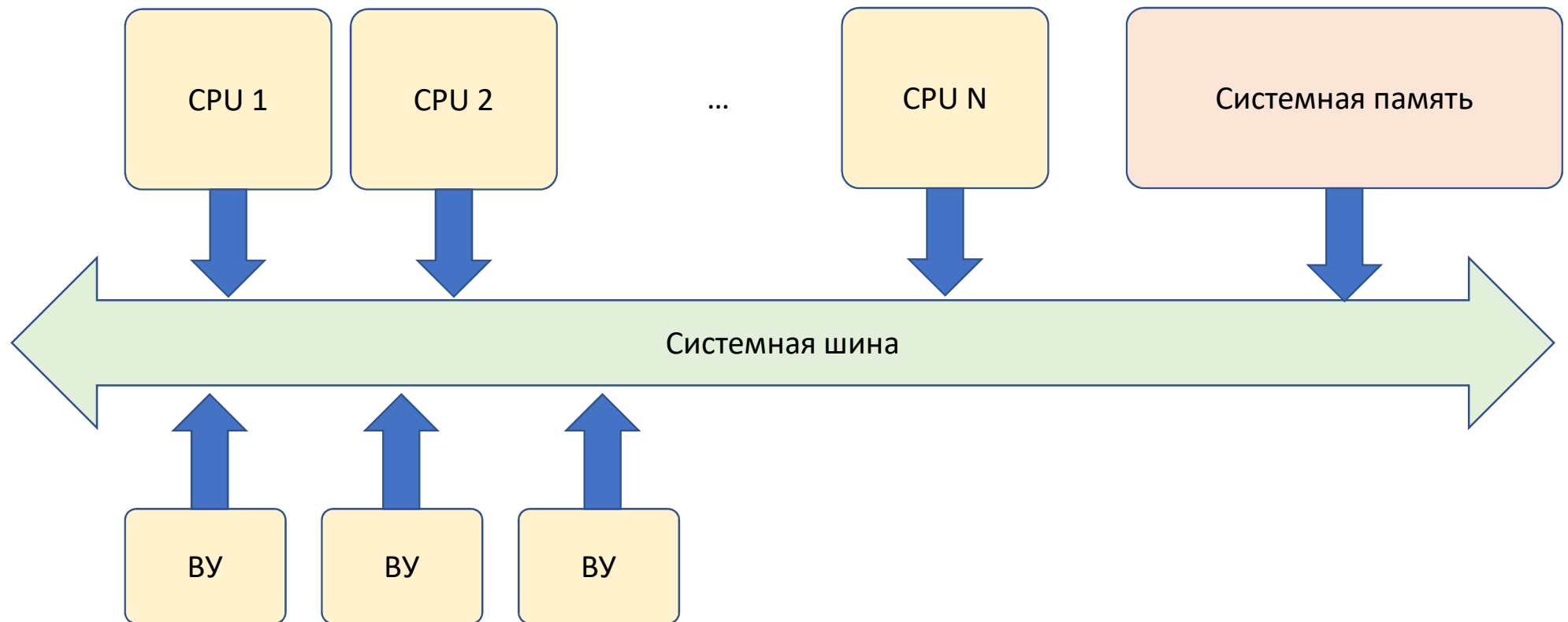
# Классификация вычислительных систем

- UMA/SMP системы – все процессы имеют равноценный доступ к памяти
- Простое масштабирование



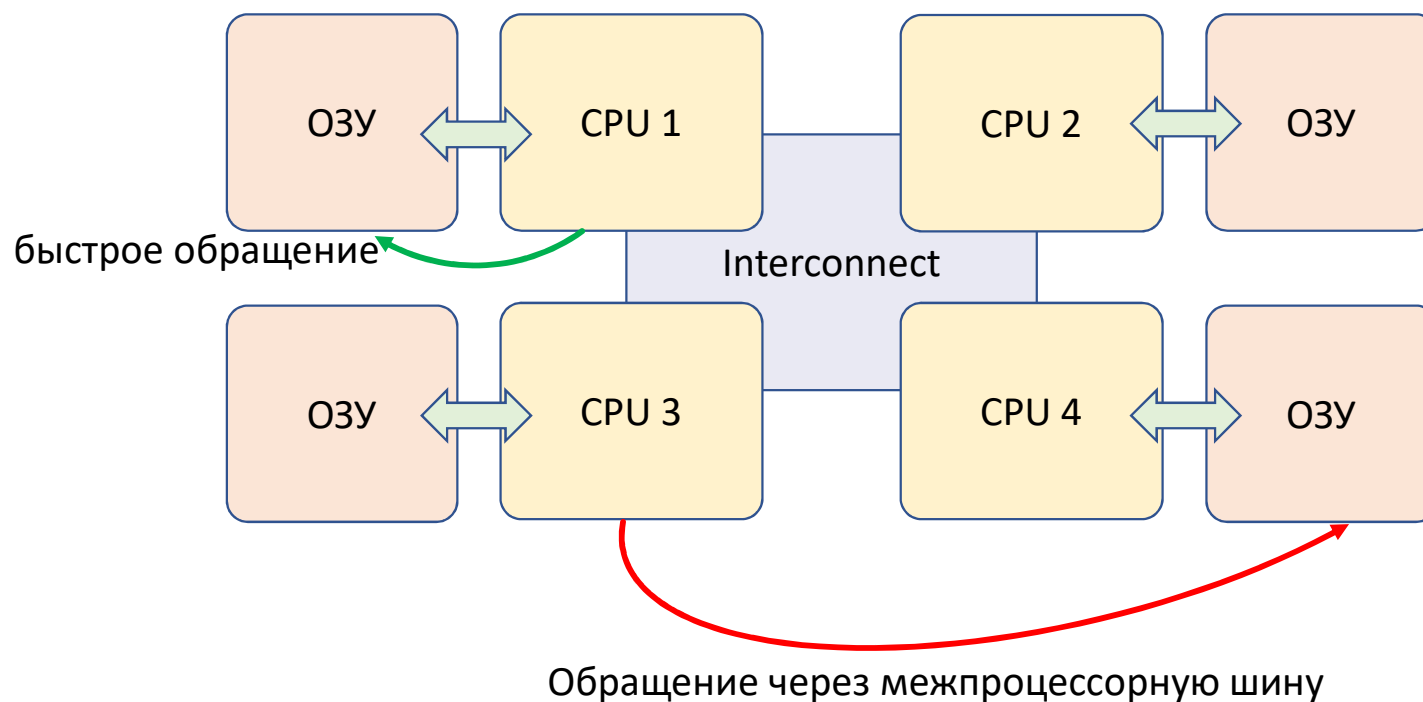
# Классификация вычислительных систем

- UMA/SMP системы – все процессы имеют равноценный доступ к памяти
- Простое масштабирование



# Классификация вычислительных систем

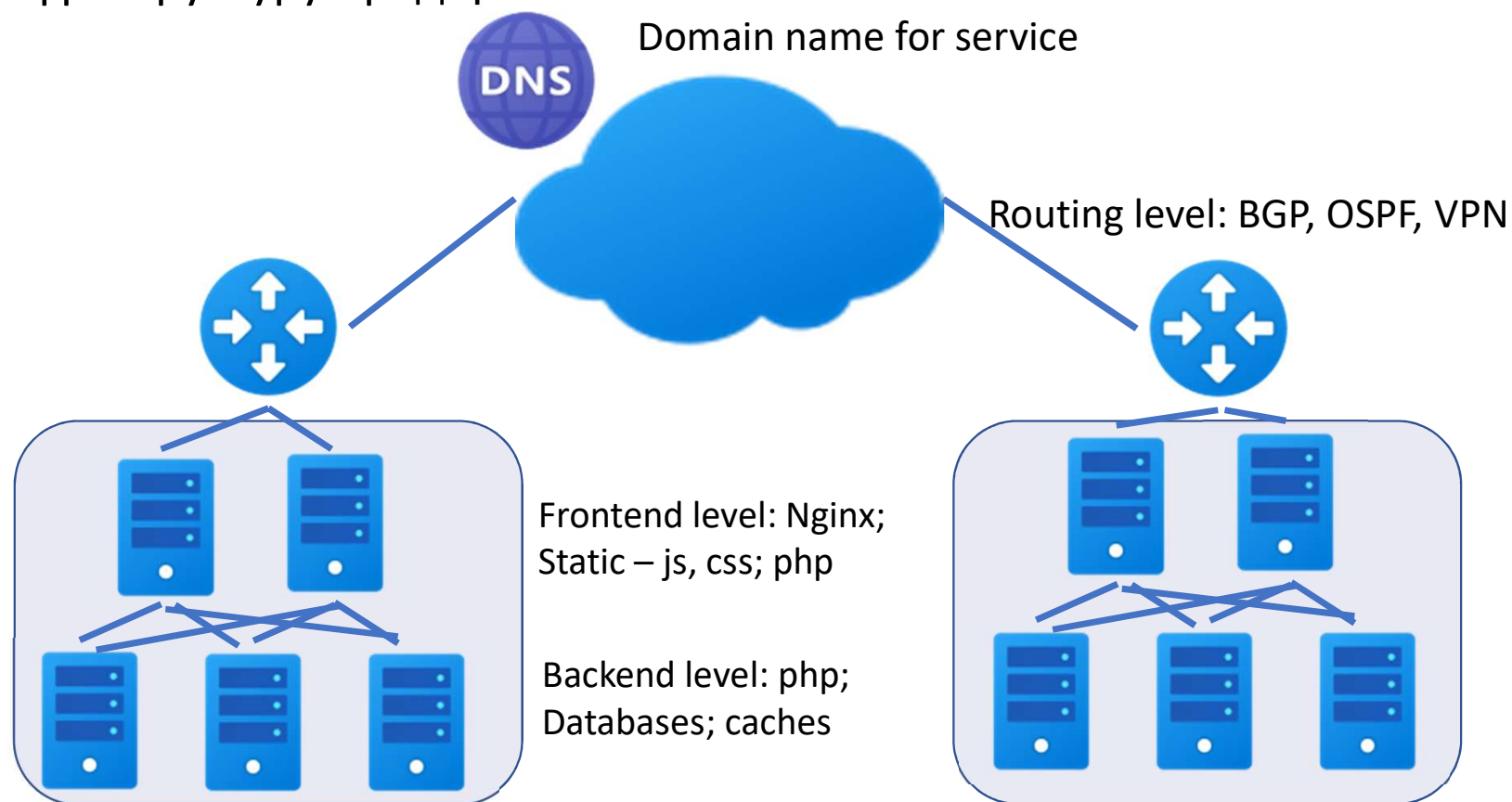
- NUMA системы – размещение банков памяти с привязкой к процессору
- Неоднородный доступ к памяти
- Скорость доступа к памяти, расположенных в банках другого процессора определяется производительностью межпроцессорной шины



```
$ numactl -H
available: 2 nodes (0-1)
node 0 cpus: 0 2 4 6
node 0 size: 11967 MB
node 0 free: 8182 MB
node 1 cpus: 1 3 5 7
node 1 size: 12054 MB
node 1 free: 10419 MB
node distances:
node  0  1
  0:  10  21
  1:  21  10
```

# Применение вычислительных систем

- Вычислительной системой можно считать информационные системы и инфраструктуру предприятий



# Классификация вычислительных систем

## Характеристики вычислительных систем

- Интенсивность отказа -  $\lambda$
- Интенсивность восстановления -  $\mu$

## Вычислительные системы со структурной избыточностью

- Для  $N$  узлов системы подразумевается наличие  $n$  резервных узлов ( $N + n$  узлов)

## Живучие вычислительные системы

- Для  $N$  узлов системы подразумевается работоспособность при  $n$  отказавших узлов (Необходимо минимум  $N - n$  работоспособных узлов)

# Применение вычислительных систем

Вычислительные системы со структурной избыточностью

- Кластер серверов
- Программное обеспечение – оркестратор: Kubernetes, proxmox, openstack
- Наличие Horizontal autoscaler
- Предоставление метрик, определяющих необходимость масштабирования: CPU, RAM

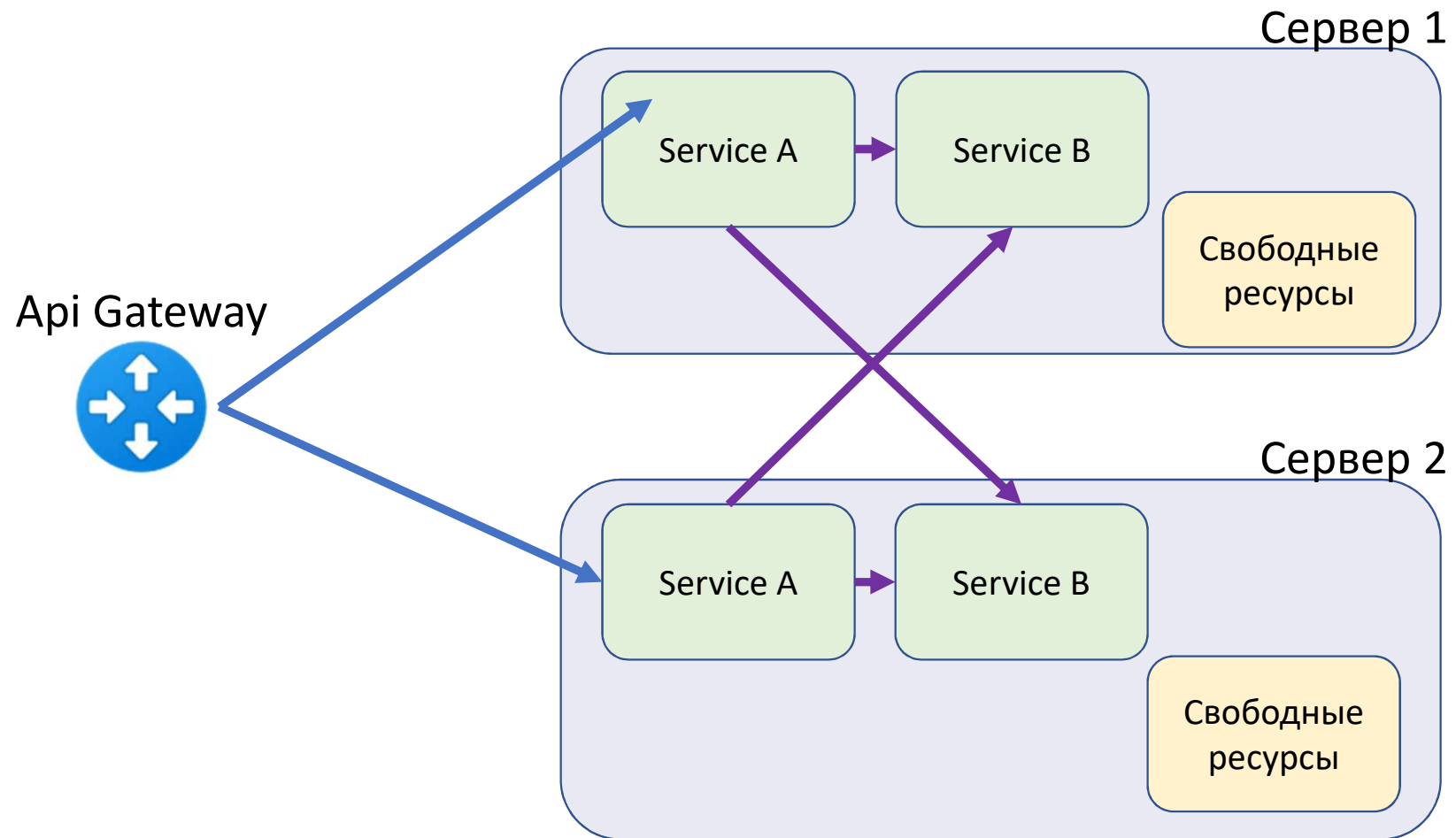
# Применение вычислительных систем

## Живучие вычислительные системы

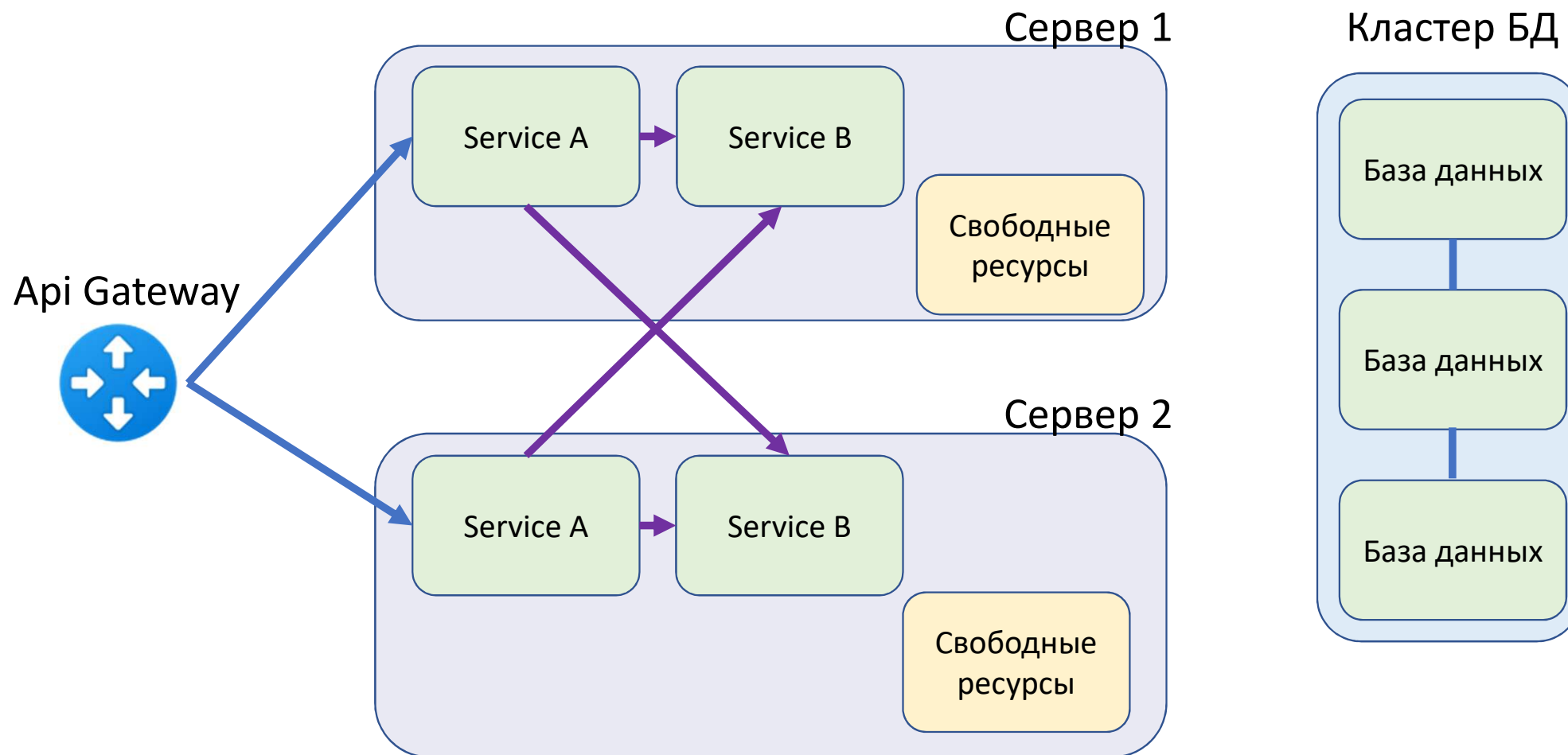
- Кластер серверов
- Программное обеспечение, работающее по принципам multi-master или использующие алгоритмы выбора лидера, кворум – Postgresql, Cassandra, galera, zookeeper, etcd, kafka, rabbitmq
- Кворум: Необходимо иметь больше половины работоспособных узлов приложения  
Достаточно для кворума  $(N + 1)/2$  голосующих экземпляров приложения



# Применение вычислительных систем



# Применение вычислительных систем



# Основы linux

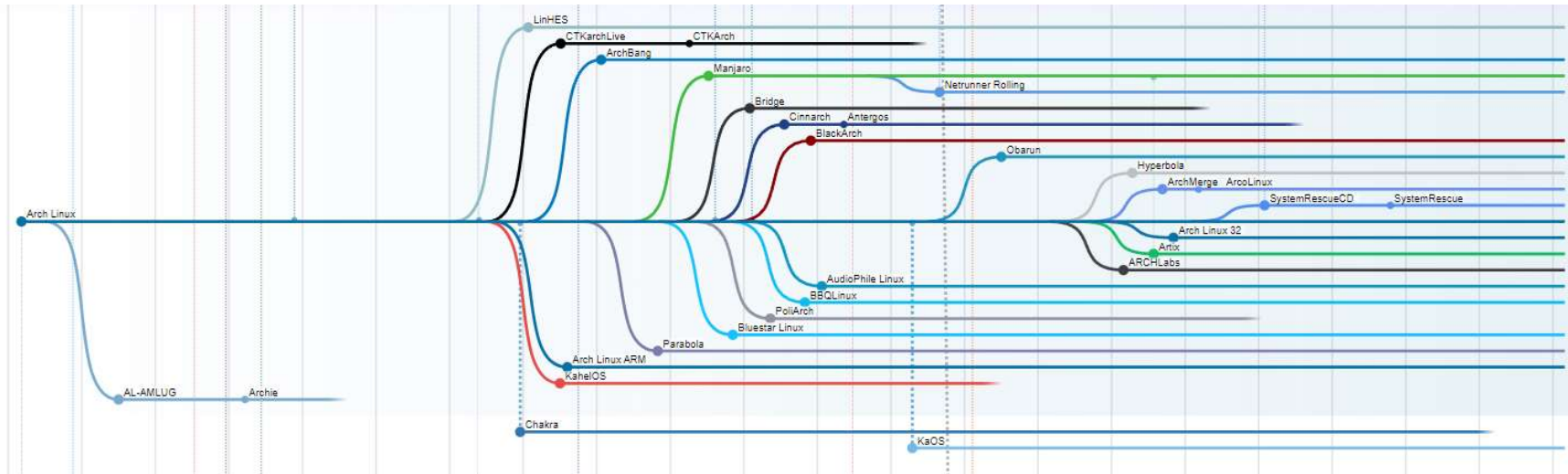
## Из истории

- 1991 год – эмулятор терминала
- 1992 – первые дистрибутивы
- 1994 – Linux с GUI
- 1996 – Linux конкурирует с unix (solaris, HP-UX), поддержка многопроцессорных систем и больших объемов памяти
- 1998 – IBM, Oracle начинают поддержку linux (СУБД), появляется RHEL (2002)

# Основы linux

## Дистрибутивы

- Debian-based
- RHEL/Fedora-based
- Suse-based
- Arch-based
- FreeBSD



## Различия

- Косметика – имена пакетов, пути, окружение
- Существенные – распространение ПО (пакетный менеджер, исходный код)

[https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux\\_Distribution\\_Timeline.svg](https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux_Distribution_Timeline.svg)

# Основы linux

## Распространение ПО

- Gentoo – исходный код
- Ubuntu – пакетный менеджер

## Политика распространения ПО

- Последние версии – Archlinux
- Стабильность и обратная совместимость – ubuntu, RHEL

# Основы linux

Известная политика – всё есть файл

- Диски и спец. устройства представлены в виде файлов - /dev
  - /dev/null – черная дыра
  - /dev/zero – поставщик нулей
  - /dev/random и /dev/urandom – нотка случайности в системе
- Системные параметры - /sys
- Информация о процессах - /proc

# Основы linux

Файловая система

- Журналируемые – ext3, ext4. Распределенные - cephfs
- Не журналируемые – xfs

В чем различие?

# Основы linux

## Файловая система

- Журналируемые – ext3, ext4. Распределенные - cephfs
- Не журналируемые – xfs

## В чем различие?

- Журналируемые – восстановление после сбоев, сначала пишется журнал.  
Write-ahead log
- Не журналируемые – журнал не пишется 😊 Но может работать быстрее и не убивать диск



# Основы linux

## Файловая система

- У каждого файла есть дескриптор – в терминах ФС это inode
- Inode создается фиксированное количество при создании ФС. Это значение нельзя изменить без пересоздания ФС

```
$ stat stat.example
File: stat.example
Size: 0          Blocks: 0          IO Block: 4096   regular empty file
Device: 0,41     Inode: 10408663   Links: 1
Access: (0644/-rw-r--r--)  Uid: (   0/   root)   Gid: (   0/   root)
Context: unconfined_u:object_r:user_home_t:s0
Access: 2024-07-11 01:59:45.129809698 +0700
Modify: 2024-07-11 01:59:45.129809698 +0700
Change: 2024-07-11 01:59:45.129809698 +0700
Birth: 2024-07-11 01:59:45.129809698 +0700
```

```
$ df -i /boot
Filesystem      Inodes IUsed IFree IUse% Mounted on
/dev/nvme0n1p2  65536   43 65493   1% /boot
```

# Основы linux: Процессы

- Информация о процессах хранится в `/proc/${PID?}`
  - PID – номер в системе
  - fd – директория со ссылками на открытые файлы (имя ссылки – номер дескриптора)
  - exe – ссылка на исполняемый файл
  - cmdline – команда запуска процесса
  - Environ – переменные окружения, доступные процессу

Процессы содержат

- Образ кода
- Адресное пространство
- Атрибуты доступа - UID/GID: кто запустил процесс и права какого пользователя он имеет
- Контекст процессора

# Основы linux: Процессы

```
# ls /proc/$$ -l | column
-r--r--r--. 1 0 0 0 Jul 11 03:40 arch_status      dr-xr-xr-x. 60 0 0 0 Jul 11 03:40 net
dr-xr-xr-x. 2 0 0 0 Jul 11 03:40 attr        dr-x--x--x. 2 0 0 0 Jul 10 22:05 ns
-rw-r--r--. 1 0 0 0 Jul 11 03:40 autogroup    -r--r--r--. 1 0 0 0 Jul 11 03:40 numa_maps
-r-----. 1 0 0 0 Jul 11 03:40 auxv         -rw-r--r--. 1 0 0 0 Jul 11 03:40 oom_adj
-r--r--r--. 1 0 0 0 Jul 11 03:40 cgroup       -r--r--r--. 1 0 0 0 Jul 11 03:40 oom_score
--w-----. 1 0 0 0 Jul 11 03:40 clear_refs   -rw-r--r--. 1 0 0 0 Jul 11 03:40 oom_score_adj
-r--r--r--. 1 0 0 0 Jul 10 22:28 cmdline      -r-----. 1 0 0 0 Jul 11 03:40 pagemap
-rw-r--r--. 1 0 0 0 Jul 11 03:40 comm         -r-----. 1 0 0 0 Jul 11 03:40 patch_state
-rw-r--r--. 1 0 0 0 Jul 11 03:40 coredump_filter -r-----. 1 0 0 0 Jul 11 03:40 personality
-r--r--r--. 1 0 0 0 Jul 11 03:40 cpu_resctrl_groups -rw-r--r--. 1 0 0 0 Jul 11 03:40 projid_map
-r--r--r--. 1 0 0 0 Jul 11 03:40 cpuset       lrwxrwxrwx. 1 0 0 0 Jul 10 22:56 root -> /
lrwxrwxrwx. 1 0 0 0 Jul 10 22:05 cwd -> /cwdpath -rw-r--r--. 1 0 0 0 Jul 11 03:40 sched
-r-----. 1 0 0 0 Jul 10 22:28 environ         -r--r--r--. 1 0 0 0 Jul 11 03:40 schedstat
lrwxrwxrwx. 1 0 0 0 Jul 10 22:56 exe -> /usr/bin/bash -r--r--r--. 1 0 0 0 Jul 11 03:40 sessionid
dr-x-----. 2 0 0 4 Jul 10 22:56 fd           -rw-r--r--. 1 0 0 0 Jul 11 03:40 setgroups
dr-xr-xr-x. 2 0 0 0 Jul 10 22:56 fdinfo        -r--r--r--. 1 0 0 0 Jul 11 03:40 smaps
-rw-r--r--. 1 0 0 0 Jul 11 03:40 gid_map      -r--r--r--. 1 0 0 0 Jul 11 03:40 smaps_rollup
-r-----. 1 0 0 0 Jul 11 03:40 io           -r-----. 1 0 0 0 Jul 11 03:40 stack
-r-----. 1 0 0 0 Jul 11 03:40 ksm_merging_pages -r--r--r--. 1 0 0 0 Jul 10 22:05 stat
-r-----. 1 0 0 0 Jul 11 03:40 ksm_stat     -r--r--r--. 1 0 0 0 Jul 11 03:40 statm
-r--r--r--. 1 0 0 0 Jul 11 03:40 latency      -r--r--r--. 1 0 0 0 Jul 10 22:28 status
-r--r--r--. 1 0 0 0 Jul 11 03:40 limits       -r-----. 1 0 0 0 Jul 11 03:40 syscall
-rw-r--r--. 1 0 0 0 Jul 11 03:40 loginuid     dr-xr-xr-x. 3 0 0 0 Jul 11 03:40 task
dr-x-----. 2 0 0 0 Jul 11 03:40 map_files    -rw-r--r--. 1 0 0 0 Jul 11 03:40 timens_offsets
-r--r--r--. 1 0 0 0 Jul 10 22:56 maps         -r--r--r--. 1 0 0 0 Jul 11 03:40 timers
-rw-----. 1 0 0 0 Jul 11 03:40 mem          -rw-rw-rw-. 1 0 0 0 Jul 11 03:40 timerslack_ns
-r--r--r--. 1 0 0 0 Jul 11 03:40 mountinfo    -rw-r--r--. 1 0 0 0 Jul 11 03:40 uid_map
-r--r--r--. 1 0 0 0 Jul 11 03:40 mounts       -r--r--r--. 1 0 0 0 Jul 11 02:08 wchan
-r-----. 1 0 0 0 Jul 11 03:40 mountstats
```

# Основы linux: Процессы

```
# cat /proc/2980628/maps
00400000-00401000 r--p 00000000 00:20 10462310      hello-world-with-pause
00401000-00402000 r-xp 00001000 00:20 10462310      hello-world-with-pause
00402000-00403000 r--p 00002000 00:20 10462310      hello-world-with-pause
00403000-00404000 r--p 00002000 00:20 10462310      hello-world-with-pause
00404000-00405000 rw-p 00003000 00:20 10462310      hello-world-with-pause
00bd9000-00bfa000 rw-p 00000000 00:00 0           [heap]
7fcc1d454000-7fcc1d457000 rw-p 00000000 00:00 0
7fcc1d457000-7fcc1d47f000 r--p 00000000 00:20 39477739  /usr/lib64/libc.so.6
7fcc1d47f000-7fcc1d5e8000 r-xp 00028000 00:20 39477739  /usr/lib64/libc.so.6
7fcc1d5e8000-7fcc1d636000 r--p 00191000 00:20 39477739  /usr/lib64/libc.so.6
7fcc1d636000-7fcc1d63a000 r--p 001de000 00:20 39477739  /usr/lib64/libc.so.6
7fcc1d63a000-7fcc1d63c000 rw-p 001e2000 00:20 39477739  /usr/lib64/libc.so.6
7fcc1d63c000-7fcc1d644000 rw-p 00000000 00:00 0
7fcc1d65c000-7fcc1d65e000 rw-p 00000000 00:00 0
7fcc1d65e000-7fcc1d65f000 r--p 00000000 00:20 39477736  /usr/lib64/ld-linux-x86-64.so.2
7fcc1d65f000-7fcc1d686000 r-xp 00001000 00:20 39477736  /usr/lib64/ld-linux-x86-64.so.2
7fcc1d686000-7fcc1d690000 r--p 00028000 00:20 39477736  /usr/lib64/ld-linux-x86-64.so.2
7fcc1d690000-7fcc1d692000 r--p 00032000 00:20 39477736  /usr/lib64/ld-linux-x86-64.so.2
7fcc1d692000-7fcc1d694000 rw-p 00034000 00:20 39477736  /usr/lib64/ld-linux-x86-64.so.2
7ffe4f851000-7ffe4f872000 rw-p 00000000 00:00 0       [stack]
7ffe4f8c4000-7ffe4f8c8000 r--p 00000000 00:00 0       [vvar]
7ffe4f8c8000-7ffe4f8ca000 r-xp 00000000 00:00 0       [vdso]
fffffffffff600000-fffffffffff601000 --xp 00000000 00:00 0 [vsyscall]
```

# Основы linux: Процессы

```
#include <stdio.h>
#include <unistd.h>

const char mystring[] = "Hello world in char constant\n";

int main() {
    printf("PID is %d\n", getpid());
    printf("Hello world string constant in printf!\n");
    sleep(9999);
    return 0;
}
```

```
# dd if=/proc/2980628/mem of=/dev/stdout bs=1 count=4096 skip="$((0x00402000))" | hexdump -C
00000000  01 00 02 00 00 00 00 00  00 00 00 00 00 00 00  |.....|
00000010  48 65 6c 6c 6f 20 77 6f  72 6c 64 20 69 6e 20 63  |Hello world in c|
00000020  68 61 72 20 63 6f 6e 73  74 61 6e 74 0a 00 50 49  |har constant..PI|
00000030  44 20 69 73 20 25 64 0a  00 00 00 00 00 00 00 00  |D is %d.....|
00000040  48 65 6c 6c 6f 20 77 6f  72 6c 64 20 73 74 72 69  |Hello world stri|
00000050  6e 67 20 63 6f 6e 73 74  61 6e 74 20 69 6e 20 70  |ng constant in p|
00000060  72 69 6e 74 66 21 00 00  01 1b 03 3b 2c 00 00 00  |rintf!.....;,...|
00000070  04 00 00 00 b8 ef ff ff  70 00 00 00 08 f0 ff ff  |.....p.....|
00000080  48 00 00 00 38 f0 ff ff  5c 00 00 00 ee f0 ff ff  |H...8...\.....|
00000090  98 00 00 00 00 00 00 00  14 00 00 00 00 00 00 00  |.....|
000000a0  01 7a 52 00 01 78 10 01  1b 0c 07 08 90 01 00 00  |.zR..x.....|
000000b0  10 00 00 00 1c 00 00 00  b8 ef ff ff 26 00 00 00  |.....&...|
000000c0  00 44 07 10 10 00 00 00  30 00 00 00 d4 ef ff ff  |.D.....0.....|
000000d0  05 00 00 00 00 00 00 00  24 00 00 00 44 00 00 00  |.....$.D...|
```

# Основы linux: Процессы

```
# cat /proc/$$/status
Name:  bash
Umask: 0022
State: S (sleeping)
Pid:   234223
PPid:  234167
TracerPid: 0
Uid:   0      0      0      0
Gid:   0      0      0      0
Threads: 1
SigQ:  4/126873
SigPnd: 0000000000000000
ShdPnd: 0000000000000000
SigBlk: 0000000000010000
SigIgn: 0000000000384004
SigCgt: 000000004b813efb
CapInh: 0000000000000000
CapPrm: 000001fffffffffff
CapEff: 000001fffffffffff
CapBnd: 000001fffffffffff
CapAmb: 0000000000000000
```

# Основы linux: Процессы

```
# lsof -p $$
COMMAND    PID  USER  FD   TYPE DEVICE SIZE/OFF      NODE NAME
bash      234223 root   cwd   DIR   0,41         0 10400605 /path/to/dir
bash      234223 root   rtd   DIR   0,34        158         256 /
bash      234223 root   txt   REG   0,34    1406608 37151060 /usr/bin/bash
bash      234223 root   mem   REG   0,32         37151060 /usr/bin/bash (path dev=0,34)
bash      234223 root   mem   REG   0,32         39476972 /usr/lib/locale/locale-archive (path
dev=0,34)
bash      234223 root   mem   REG   0,32         39477739 /usr/lib64/libc.so.6 (path dev=0,34)
bash      234223 root   mem   REG   0,32         37151217 /usr/lib64/libtinfo.so.6.4 (path dev=0,34)
bash      234223 root   mem   REG   0,32         39497934 /usr/lib64/gconv/gconv-modules.cache (path
dev=0,34)
bash      234223 root   mem   REG   0,32         39477736 /usr/lib64/ld-linux-x86-64.so.2 (path
dev=0,34)
bash      234223 root    0u   CHR 136,59         0t0         62 /dev/pts/59
bash      234223 root    1u   CHR 136,59         0t0         62 /dev/pts/59
bash      234223 root    2u   CHR 136,59         0t0         62 /dev/pts/59
bash      234223 root   255u  CHR 136,59         0t0         62 /dev/pts/59
```

# Основы linux: Процессы

- Лимиты – ulimit
- prlimit – изменение на лету

```
$ cat /proc/$$/limits
```

Limit	Soft Limit	Hard Limit	Units
Max cpu time	unlimited	unlimited	seconds
Max file size	unlimited	unlimited	bytes
Max data size	unlimited	unlimited	bytes
Max stack size	8388608	unlimited	bytes
Max core file size	unlimited	unlimited	bytes
Max resident set	unlimited	unlimited	bytes
Max processes	126873	126873	processes
Max open files	1024	524288	files
Max locked memory	8388608	8388608	bytes
Max address space	unlimited	unlimited	bytes
Max file locks	unlimited	unlimited	locks
Max pending signals	126873	126873	signals
Max msgqueue size	819200	819200	bytes
Max nice priority	0	0	
Max realtime priority	0	0	
Max realtime timeout	unlimited	unlimited	us



# Основы linux: Процессы

top – утилизация ресурсов системы

```
$ top -bn1 | head -n 20
top - 01:33:40 up 25 days,  8:41, 10 users,  load average: 1.94, 1.40, 1.32
Tasks: 502 total,   2 running, 499 sleeping,   0 stopped,   1 zombie
%Cpu(s):  9.4 us,  2.3 sy,  0.0 ni, 88.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 31756.3 total,  4480.4 free, 10114.5 used, 18396.2 buff/cache
MiB Swap:  8192.0 total,  7115.5 free,  1076.5 used. 21641.8 avail Mem
```

# cgroups

- Размещаются в sysfs – `/sys/fs/cgroups`
- Группируются в slice – `system.slice`, `user.slice`

Разделяются на типы

- `cpu`
  - `cpuset`
  - `memory`
  - `hugetlb`
  - `blkio`
- 
- `freezer` позволяет остановить/запустить процессы всей группы или slice

# cgroups

- Размещаются в sysfs – `/sys/fs/cgroups`
- Группируются в slice – `system.slice`, `user.slice`

Разделяются на типы

- `cpu`
  - `cpuset`
  - `memory`
  - `hugetlb`
  - `blkio`
- 
- `freezer` позволяет остановить/запустить процессы всей группы или slice

# Bash

- Bash – усовершенствованная и модернизированная вариация командной оболочки Bourne shell.
- Скрипты в рабочей среде принято писать именно под bash. Реже – под «чистый» sh. Однако для локальной работы более популярны fish и zsh
- Для того, чтобы скрипт корректно работал, первой строкой указывается «шебанг» - программа, которая может этот скрипт выполнять. Например, исполняемый текстовый файл со скриптом на питоне в начале содержит `#!/usr/bin/env python3`

# Ssh

- ssh – основное средство доступа к удаленным системам
  - Аутентификация по паролю, ассиметричным ключам, сертификатам
  - Алгоритмы ключей – rsa, ed25519
  - Публичные ключи на сервере хранятся в файле `authorized_keys` пользователя
  - Позволяет пробрасывать ключ через ssh-agent (`ssh jump host`)
- 
- `ssh [-i ssh_privkey] [-l username] [-p port] ${ipaddress?}`
  - `ssh-keygen -t ed25519 -f file.key [-C comment]`

Live section

Романюта Алексей Андреевич

[alexey-r.98@yandex.ru](mailto:alexey-r.98@yandex.ru)

Кафедра вычислительных систем  
Сибирский государственный университет телекоммуникаций и информатики

