Администрирование локальных вычислительных сетей

Лекция 1

Основы работы в Astra Linux

Архитектура ОС GNU/Linux

Пользовательское пространство

Приложения (офисные, графические, браузеры, утилиты и т.д.)

Службы (веб-сервер, СУБД, X сервер и т.д.)

Системные библиотеки (glibc и др.)

Пространство ядра

Системные вызовы (system calls)

Подсистема ввода/вывода

Виртуальная файловая система

Драйверы файловых систем

Драйверы блочных устройств

Драйверы символьных устройств

Сетевой стек

Драйверы сетевых устройств

Подсистема процессов

Межпроцессное взаимодействие

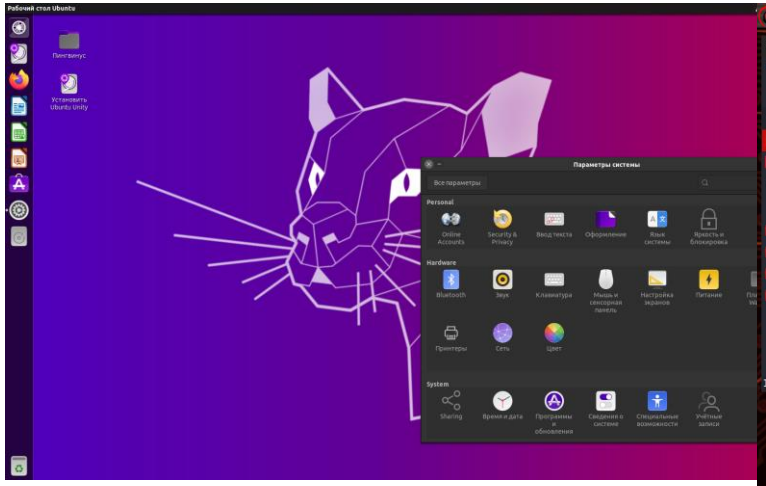
Управление памятью

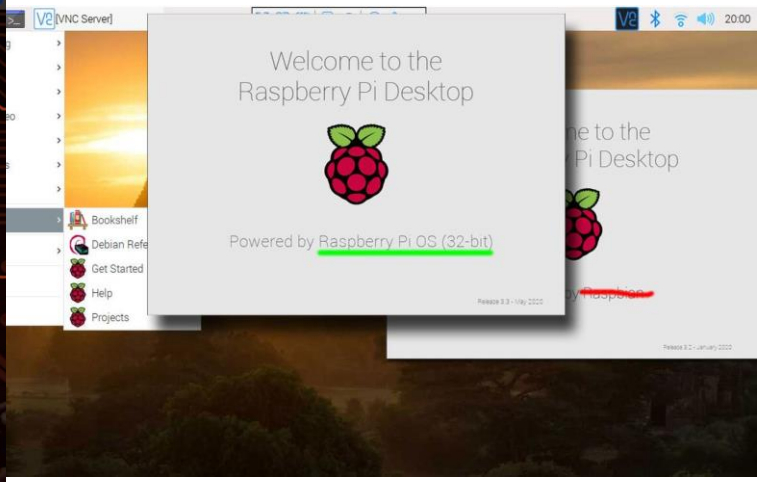
Планировщик процессов

Архитектурно-зависимый код

Аппаратное обеспечение

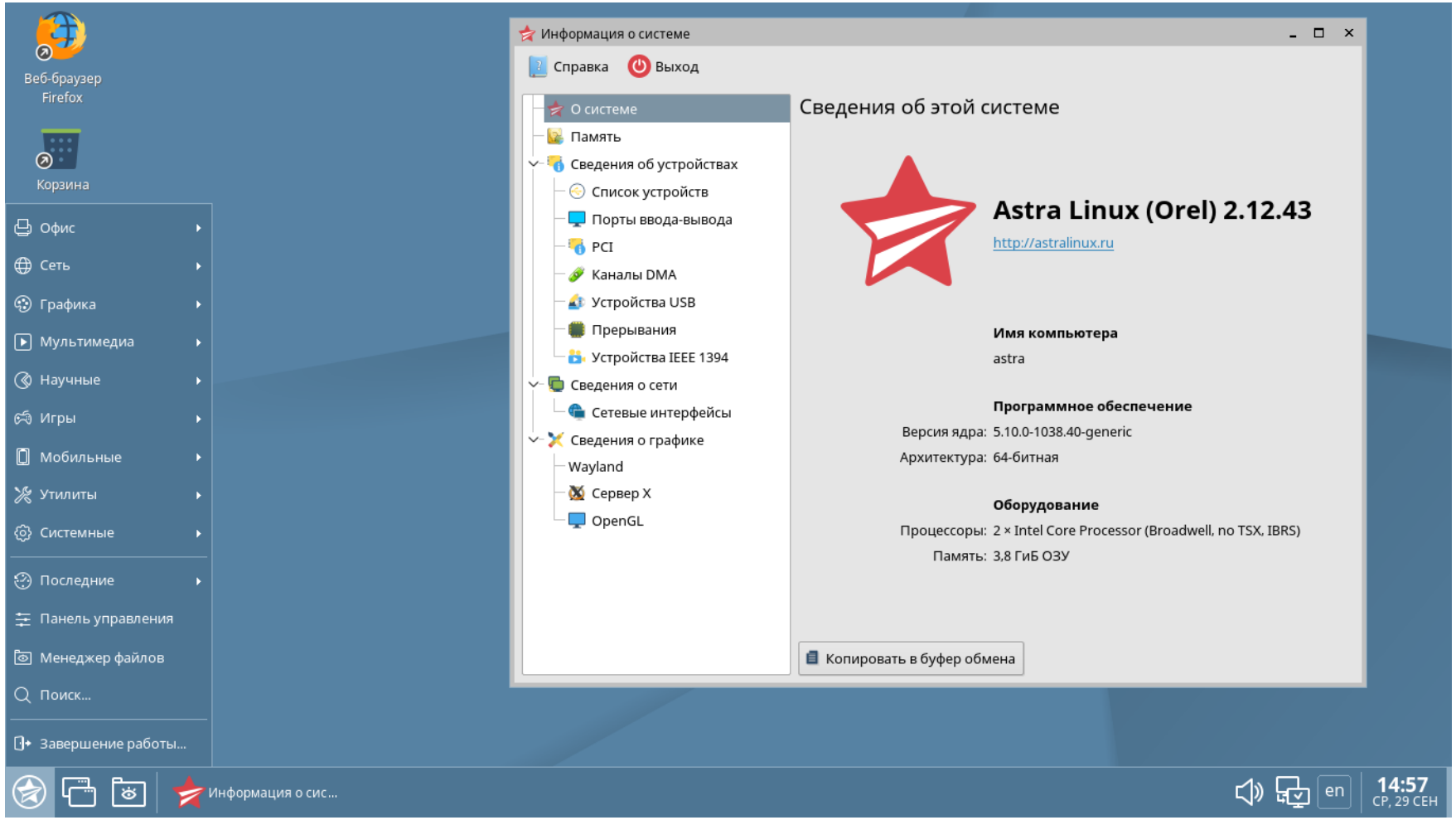
Дистрибутивы Linux





- Дистрибутив Linux - это комплект взаимосогласованного программного обеспечения, включающий ОС на базе ядра Linux, системное программное обеспечение и приложения, а также систему управления программным обеспечением
- Большая часть программного обеспечения в дистрибутиве является свободным (распространяется по свободным лицензиям)
- Минимальным объектом дистрибутива является программный пакет
- Программные пакеты могут собираться как в двоичном виде, так и в исходных кодах
- Дистрибутивы Linux размещаются в специальном хранилище – репозитории (в локальном или сетевом)

Дистрибутивы Astra Linux



Дистрибутивы семейства Astra Linux являются официальными производными дистрибутива Debian (Astra Linux x.7 базируется на Debian 10 Buster)

Уровни защищенности ОС Astra Linux

«ОРЁЛ»

Базовый уровень защищенности

Рекомендовано для обработки общедоступной информации, не требующей специальных средств защиты

«ВОРОНЕЖ»

Усиленный уровень защищенности

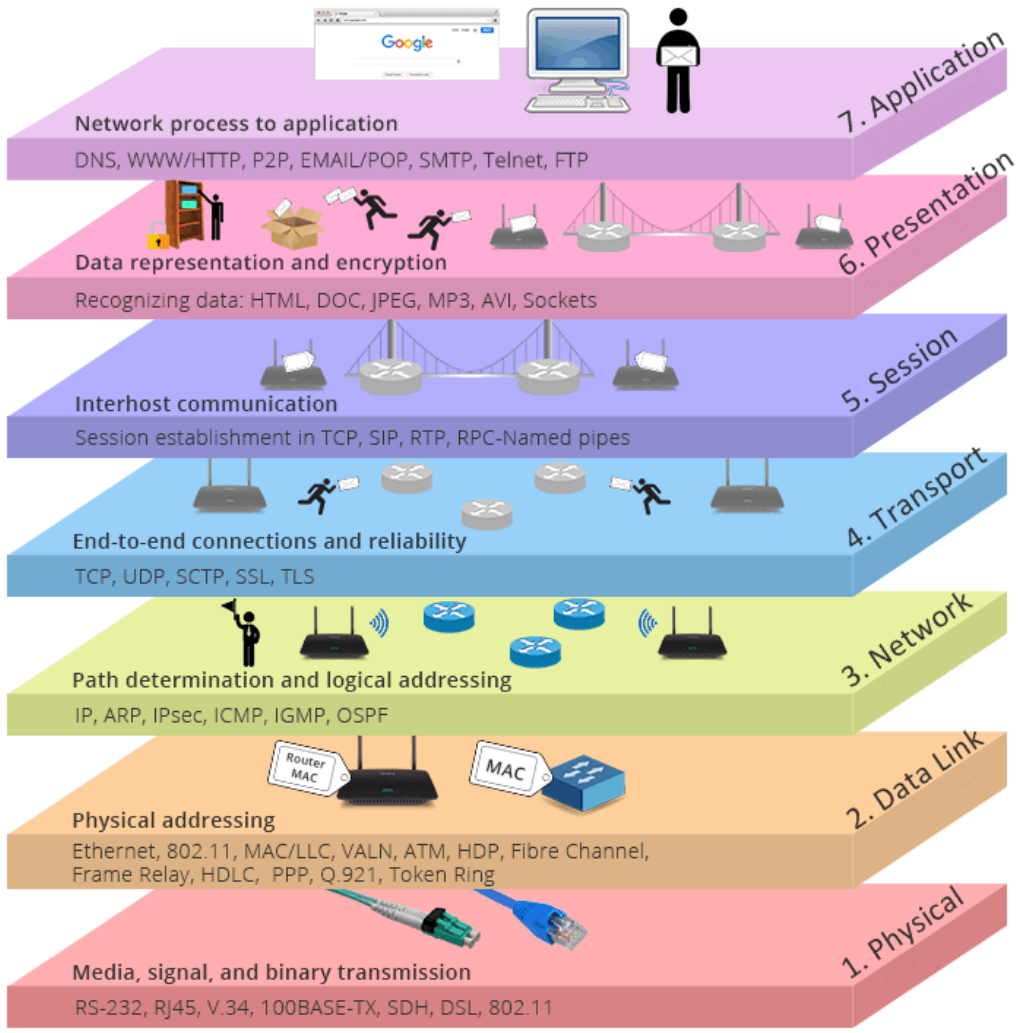
Рекомендовано для обработки конфиденциальной информации не содержащей сведения, составляющие государственную тайну

«СМОЛЕНСК»

Максимальный уровень защищенности

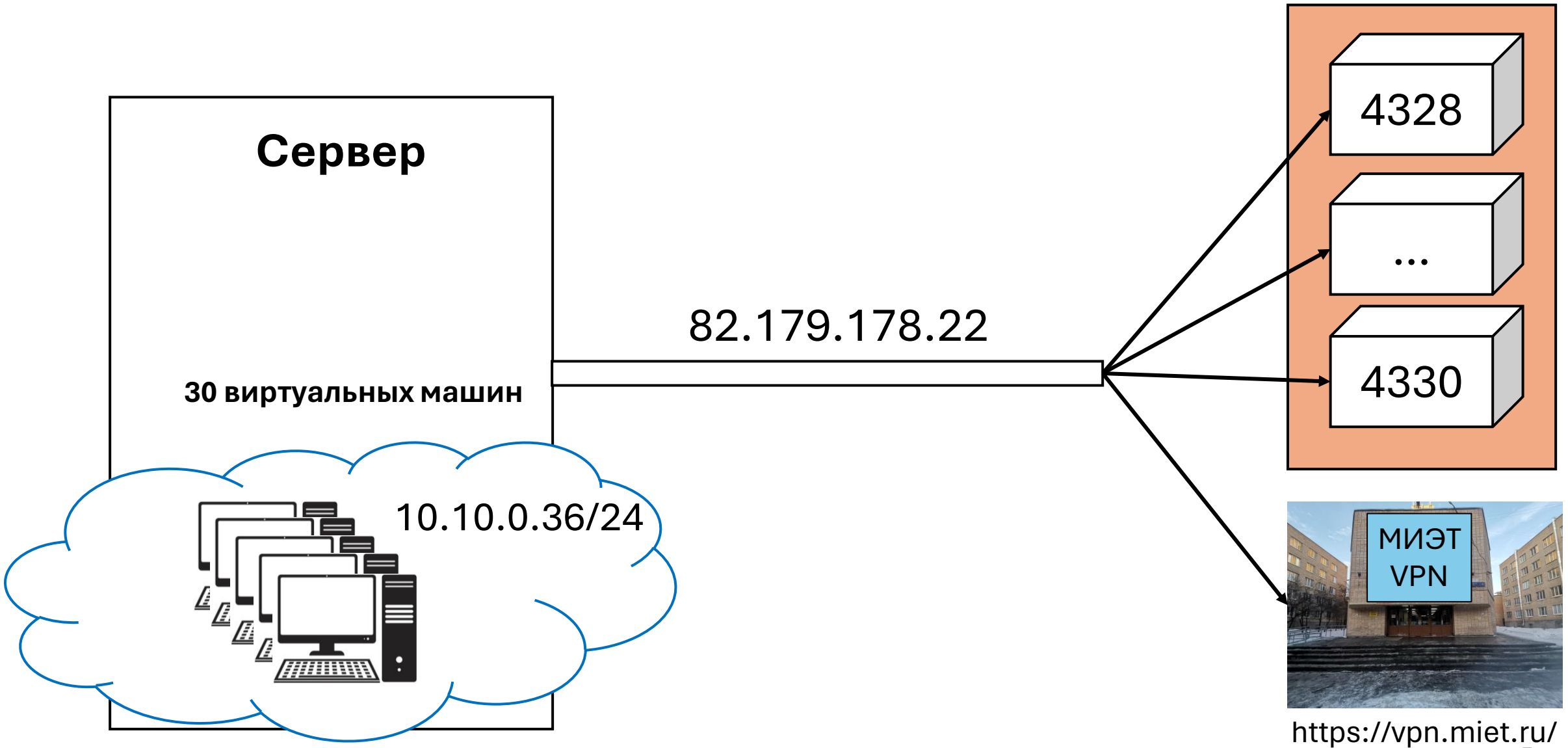
Рекомендовано для обработки информации любой категории доступа в государственных информационных системах

Модель OSI



OSI		TCP/IP (DOD)	
7	**Прикладной уровень** Application layer	4	Уровень приложений Application layer
6	**Уровень представления** Presentation layer		
5	**Сеансовый уровень** Session layer		
4	**Транспортный уровень** Transport layer	3	**Транспортный уровень** Transport layer
3	**Сетевой уровень** Network layer	2	Уровень сети Интернет **Internet**
2	**Канальный уровень** Data link layer — LLC / MAC	1	Уровень доступа к сети **Network Access layer**
1	**Физический уровень** Physical layer		

Как подключиться к Astra Linux?



https://vpn.miet.ru/

Как подключиться к Astra Linux?

В браузере ПК вводим адрес виртуальной машины:

82.179.178.22:58(XX+3)/vnc.html

Логин: study0XX

Пароль: qwerty22

Например:
Вариант 10
82.179.178.22:5813/vnc.html
study010
qwerty22

XX – ваш номер по списку

ftp сервер

В менеджере файлов Астра Линукс вводим:

ftp://10.10.0.36

Логин: studyftp

Пароль: qwerty21

FTP-сервер —сервер, работающий по протоколу File Transfer Protocol и предназначенный для обмена файлами через Интернет или локальную компьютерную сеть.

Работа с командной строкой

```
sab@SAB:.../Lesson1$
```

№	Команда	Описание	Пример
0	man	Описание работы команды	man ls
1	pwd	Показать текущее местонахождение	~/SAB$ pwd /home/user/SAB
2	ls	Позволяет просмотреть содержимое текущего каталога	~/SAB$ ls 1 1.txt
3	cd <путь к директории>	Перейти в другую директорию	~$ cd ~ /SAB/2 (полный путь) или ~/SAB$ cd 2 (короткий путь)
4	mkdir <название директории>	Создание директории	~/SAB$ mkdir 1
5	touch <название файла>	Создание файла	~/SAB$ touch 1.txt
6	nano <название файла>	Редактирование файла	~/SAB$ nano 1.txt
7	cp <что_копировать куда_копировать>	Копирование файла	~/SAB/1$ cp 1.txt ~/SAB/2
8	cp -r <путь_к_папке путь_к_новому_месту>	Копирование директории	~/SAB/1$ cp -r 1 ~/SAB/2
9	mv <что_переместить куда_переместить>	Переместить файл	~/SAB/1$ mv 1.txt ~/SAB/2
10	rm <название файла>	Удалить файл	~/SAB/1$ rm 1.txt
11	rm -r <название файла>	Удалить директорию	~/SAB/1$ rm -r 1

Перенаправление ввода и вывода

Ввод и вывод распределяется между тремя стандартными потоками:

- **stdin** — стандартный ввод (клавиатура), - - 0
- **stdout** — стандартный вывод (экран), - 1
- **stderr** — стандартная ошибка (вывод ошибок на экран). - 2

< file — использовать файл как источник данных для стандартного потока ввода.

> file — направить стандартный поток вывода в файл (перезапись)

2> file — направить стандартный поток ошибок в файл (перезапись)

>>file — направить стандартный поток вывода в файл (добавление)

2>>file — направить стандартный поток ошибок в файл. (добавление)

&>file или **>&file** — направить с.п. вывода и с.п. ошибок в файл.

Перенаправление ввода и вывода

< file — использовать файл как источник данных для стандартного потока ввода.
> file — направить стандартный поток вывода в файл (перезапись)
2> file — направить стандартный поток ошибок в файл (перезапись)
>>file — направить стандартный поток вывода в файл (добавление)
2>>file — направить стандартный поток ошибок в файл. (добавление)
&>file или **>&file** — направить с.п. вывода и с.п. ошибок в файл.

```
sab@SAB: /$ ps > 1.txt
sab@SAB: /$ cat 1.txt
sab@SAB: /$ ps >> 1.txt
sab@SAB: /$ ps qq > 1.txt
sab@SAB: /$ ps qq 2> 1.txt
```

Grep

Grep *(global regular expression printer) – утилита командной строки, позволяющая производить поиск строки в файле.*

grep [*ключи*] шаблон [*имя_файла* ...]

Ключ	Описание
-c	Выдает только количество строк, содержащих выражение.
-h	Скрывает вывод названия файла, в котором было обнаружено вхождение. Используется при поиске по нескольким файлам.
-i	Игнорирует регистр символов при поиске.
-l	Выдает только имена файлов, содержащих сопоставившиеся строки.
-n	Выдает перед каждой строкой ее номер в файле (строки нумеруются с 1).
-s	Скрывает выдачу сообщений о не существующих или недоступных для чтения файлах.
-v	Выдает все строки, за исключением содержащих выражение.
-E	Поиск с использованием регулярных выражений
-o	Вывод только обнаруженных символов
-r	Рекурсивный поиск

Grep (примеры)

```
sab@SAB: /$ grep Hello file.txt
```

Ищем строку ***Hello*** в файле ***file.txt***

```
sab@SAB: /$ grep -r Hello .
```

Ищем строку ***Hello*** во ***всех*** файлах текущей директории

```
sab@SAB: /$ grep -c Hello file.txt
```

Ищем число вхождений строки ***Hello*** в файле ***file.txt***

Grep (пример с директориями)

Задача:

Необходимо быстро найти пароль от телефона среди множества других паролей.

```
sab@SAB: /$ ./script.sh 5 5 5 100
```

Генерируем 5 директорий, в каждой 5 поддиректорий, в каждой 5 файлов, в каждом из которых 100 строк. В одной из них содержится строка: Password_phone:XXXXXX

```
sab@SAB: /$ grep -R "Password_phone" ./Files
```

Производим поиск по директории *Files*

```
sab@SAB: /$ rm -rf Files/
```

Удаляем созданные директории

Регулярные выражения

Регулярные выражения - инструмент для поиска текста по шаблону.

Метасимвол	Описание работы
****	начало буквенного спецсимвола
^	указывает на начало строки
$	указывает на конец строки
*****	указывает, что предыдущий символ может повторяться 0 или больше раз
+	указывает, что предыдущий символ должен повторится больше один или больше раз
?	предыдущий символ может встречаться ноль или один раз
{n}	указывает сколько раз (n) нужно повторить предыдущий символ
{N,n}	предыдущий символ может повторяться от N до n раз
.	любой символ кроме перевода строки
[az]	любой символ, указанный в скобках
x|y	символ x или символ y
[^az]	любой символ, кроме тех, что указаны в скобках
[a-z]	любой символ из указанного диапазона
[^a-z]	любой символ, которого нет в диапазоне
[[:alpha:]]	является алфавитным символом
[[:digit:]]	является числом

Регулярные выражения (примеры)

Поиск содержимого файлов, начинающихся с символов A или B

```
sab@SAB: /$ grep -re '^[AB]' .
```

Поиск количества строк в каждом файле, содержащие подряд две буквы «в»

```
sab@SAB: /$ egrep -rc "[в]{2}" .
```

Поиск содержимого файлов, содержащих слова «Вы или вы»

```
sab@SAB: /$  grep -re '[Вв]ы' .
```

Поиск в файле IPv4 адресов (*для любителей реальной практики*)

```
sab@SAB: /$ grep -E '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' ip.txt
```

Задача на подумать...

Поиск в файле IPv4 адресов (*для любителей реальной практики*)

```
sab@SAB: /$ grep -E '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' ip.txt
```

Необходимо ввести ограничение на *ip* адреса

Адреса:

123.321.234.712	-	(>255)
999.999.999.999	-	
192.168.5.5	+	
10.0.0.4	+	
10.000.000.04	-	(не более 1 нуля подряд)

find

find *– утилита, с помощью которой возможно найти файл по его имени*

```
find [адрес начала поиска] ... [выражение]
```

Вывести все файлы с информацией о них

```
sab@SAB: /$ find . -printf '%M %n %s %Tb %p\n'
```

Вывести все файлы, размером более 100 МБ, но менее 2ГБ

```
sab@SAB: /$ find . -size +100M -size -2G
```

Вывести все файлы, изменённых не ранее 30 и не позднее 20 минут назад

```
sab@SAB: /$ find . -type f -mmin +20 -mmin -30
```

cut

cut *– утилита, с помощью которой возможно вырезать символы из каждой переданной ей строки*

```
cut [ключи] ... [ имя_файла ... ]
```

Ключ	Описание
-c	Вырезать указанную последовательность символов
-d	Указать символ разделения (по умолчанию TAB)
-f	Выбрать символы, разделенные определенным символом
-s	Указание на пропуск любой строки, в которой нет разделителя

Форматы для задания списка полей или колонок:

A-B Поля или колонки от A до B включительно

A- От поля или колонки A до конца строки

-B С начала строки до поля или колонки B

A, B Поля или колонки A и B

cut (примеры)

Вывести первые пять символов файла

```
sab@SAB: /$ cut -c 1-5 1.txt
```

Выделить второй столбец из строки, символ разделения - двоеточие

```
sab@SAB: /$ cut -d ':' -f 2 1.txt
```

Выделить первое слово из текста (символ разделения - пробел), если их нет в строке – строка пропускается

```
sab@SAB: /$ cut -d' ' -f1 -s 1.txt
```

tr

tr (translate) *– утилита командной строки, позволяющая посимвольно обрабатывать текст*

```
tr [ключи]... набор1 [набор2]
```

Ключ	Описание
-c	Оставить только символы из первого набора, остальные заменить на значение из второго набора
-d	Удалить символы, входящие в первый набор
-s	Заменяет все символы из первого набора, встречающиеся несколько раз подряд на одиночное вхождение

tr (примеры)

Заменить все строчные символы на прописные

```
sab@SAB: /$ tr "a-z" "A-Z" < 1.txt
```

```
sab@SAB: /$ tr "[[:lower:]]" "[[:upper:]]" < 1.txt
```

Заменить все символы, кроме a-z, \n на символ X

```
sab@SAB: /$ tr -c "a-z\n" X < 1.txt
```

Удалить все строчные символы

```
sab@SAB: /$ tr -d "\n" < 1.txt
```

Заменить все повторяющиеся пробелы на один

```
sab@SAB: /$ tr -s " " < 1.txt
```

wc

wc (word count) *– утилита командной строки, выводящая число переводов строк, слов и байт для каждого указанного файла и итоговую строку, если было задано несколько файлов.*

```
wc [ключи]... [ имя_файла ... ]
```

Ключ	Описание
-c	Вывести размер файла в байтах
-m	Вывести количество символов в файле
-l	Вывести количество строк в файле
-w	Вывести количество слов в файле

Вызов без параметров вернёт все значения количества строк, слов и символов в файле

Вывести число строк в файле

```
sab@SAB: /$ wc -l file.txt
```

uniq

uniq *– утилита командной строки, предназначена для поиска одинаковых строк в массивах текста.*

```
uniq [OPTION]... [INPUT [OUTPUT]]
```

Ключ	Описание
-u	Вывести исключительно те строки, у которых нет повторов.
-d	Вывести все строки, удаляя дубликаты
-D	Вывести только повторяющиеся строки.
-c	В начале каждой строки вывести число, которое обозначает количество повторов.

Вывести число уникальных строк в файле

```
sab@SAB: /$ uniq -u file.txt
```

awk

awk *– скриптовый язык построчного разбора и обработки входного потока*

awk [*ключи*] '[шаблон] {действие}'

Утилита awk последовательно применяется к каждой из строчек. Для фильтрации строк применяется шаблон, накладывающий ограничение на отбираемые строки

Расширенный вариант *awk*

```
BEGIN {действие}                -> Выполняется до обработки строк
      шаблон {действие}
      шаблон {действие}
END {действие}                  -> Выполняется после обработки строк
```

Ключ	Описание
-F fs	Указать символ разделения (по умолчанию пробел)
-v var=val	Задать значение переменной

awk

Пользователь может использовать встроенные переменные или создавать свои.
Awk рассматривает переменную как строковую. По умолчанию строка инициализируется "0" или пустой строкой

Типы переменных:

- позиционные
- числа с плавающей точкой
- строка символов
- массив

Утилита awk поддерживает стандартные конструкции языка С – ветвления, циклы

Основные встроенные переменные	Описание
$0	Значение обрабатываемой строки
$1 - $N	Значение определенного столбца, полученных в результате разделения строки сепаратором
FS	Разделитель полей записи на вводе
NF	Число полей в текущей записи
NR	Номер записи (общее число считанных записей)

Некоторые команды	Описание
print	Вывести строку целиком
length(N)	Длина N в символах

awk (примеры)

Все примеры производятся над файлом, содержащим значения ФИО студентов, их дату рождения и оценку за экзамен

Ivanov Ivan Ivanovich 15-04-1995 87
Petrov Sergey Aleksandrovich 23-09-1992 78
Smirnov Olga Nikolaevna 07-06-1990 92
Kuznetsov Dmitry Anatolyevich 12-11-1997 89
Popov Elena Ivanovna 02-03-1988 94
Sokolova Anna Alekseevna 19-07-1992 76
.....

awk (примеры)

Напечатать весь текст

```
sab@SAB: /$ awk '{print}' test.txt
```

Напечатать сообщение приветствия и прощания между текстом

```
sab@SAB: /$ awk 'BEGIN {print "Hello"} {print} END {print "Good Bye"}' test.txt
```

Напечатать первый и второй столбец текста (Фамилию и имя)

```
sab@SAB: /$ awk '{print $1, $2}' test.txt
```

Напечатать целиком все строки, содержащие символ F

```
sab@SAB: /$ awk '/F/{print $0}' test.txt
```

Напечатать второй столбец текста, символ разделитель - тире

```
sab@SAB: /$ awk –F- '{print $2}' test.txt
```

Вывести число строк в файле

```
sab@SAB: /$ awk 'END{print NR}' test.txt
```

awk (примеры)

Вывести сумму значений 5го столбца

```
sab@SAB: /$ awk '{sum+=$5} END{print sum}' test.txt
```

Вывести все строки, где пятый столбец равен 90

```
sab@SAB: /$ awk '{if ($5==90) print $0}' test.txt
```

Вывести все строки, где пятый столбец равен 90 и вывести их количество

```
sab@SAB: /$  awk '{if ($5==90) {L+=1; print $0}} END{print L}' test.txt
```

Вывести сообщение о результате аттестации в случае получения оценки

```
sab@SAB: /$ awk '{if ($5>=90) {print $1 " " $2 ": Zachet"} else {print $1 " " $2 ": Peresdacha"}}' test.txt
```

Посчитать количество символов в каждом слове файла

```
sab@SAB: /$ awk '{ for(i=1; i<=NF; i++) {L+=length($i)}} END {print L}' test.txt
```

Пояснение: for(i=1; i<=NF; i++) – стандартный цикл. При обращении к $i получаем значения всех столбцов по очереди с первого по последний.

Pipes (Каналы)

Каналы используются для перенаправления потока из одной программы в другую.

```
sab@SAB: /$ ps | grep p
sab@SAB: /$ ps > 1.txt; ls >> 1.txt; cat 1.txt | grep a
```

Несколько полезных команд Linux

hexdump — показывает шестнадцатеричное представление данных, поступающих на стандартный поток ввода.
cat — считывает данные со стандартного потока ввода и передает их на стандартный поток вывода.
sudo - запуск программы от имени других пользователей, а также от имени суперпользователя.
bc – калькулятор
python3 – среда разработки Python
sort – сортировка переданных значений (-r – обратный порядок)

```
sab@SAB: /$ echo "10*10" | bc
sab@SAB: /$ hexdump 1.txt
```

Несколько горячих клавиш

!^ Первый аргумент
!:2 Второй аргумент
!:2-$ second to last arguments
!:2* second to last arguments
!:2- second to next to last arguments
!:2-3 second to third arguments
!$ last argument
!* all arguments

```
sab@SAB: /$ touch ~/new.txt
sab@SAB: /$ cat !$
```

Потренируемся

Задание 1.

1. Перейдите в директорию lab1.
2. Оставаясь в директории lab1, создайте в каталоге poems/English файл, содержащий текст вашего любимого стихотворения отечественного автора. Название файла должно соответствовать названию стихотворения. Внутри файла, перед текстом произведения укажите название и автора.
3. Перенесите созданный файл из директории English в Russian.
4. Создайте в директории English каталоги, содержащие названия веков и распределите по ним расположенные в ней стихотворения.

Задание 2.

1. Произведите поиск всех стихотворений, названия которых содержит только кириллицу
2. Найдите все файлы с расширением jpeg
3. Найдите все файлы, которые были изменены за последние 20 минут
4. Найдите все файлы, объемом больше 500 Кб

Задание 3.

1. Вычислите у скольких стихотворений вместо названия стоят “***”
2. Выведите напротив каждого файла сообщение о том, содержит ли он восклицательный знак
3. Рассчитайте, сколько раз в тексте стихотворений встречается предлог «на»
4. Вычислите самое часто встречающееся слово в монологе Гамлета

Чтобы посмотреть, когда пользователь последний раз входил в систему, введите:
$ lastlog

В списке отобразится выполняемый перечень команд. Если команд исполняется несколько, они будут отображены столбцом в списке. Также можно посмотреть историю входов пользователей в систему. Для этого есть команда last, она выводит информацию на основе лога /var/wtmp:
$ last –a

Администрирование локальных вычислительных сетей

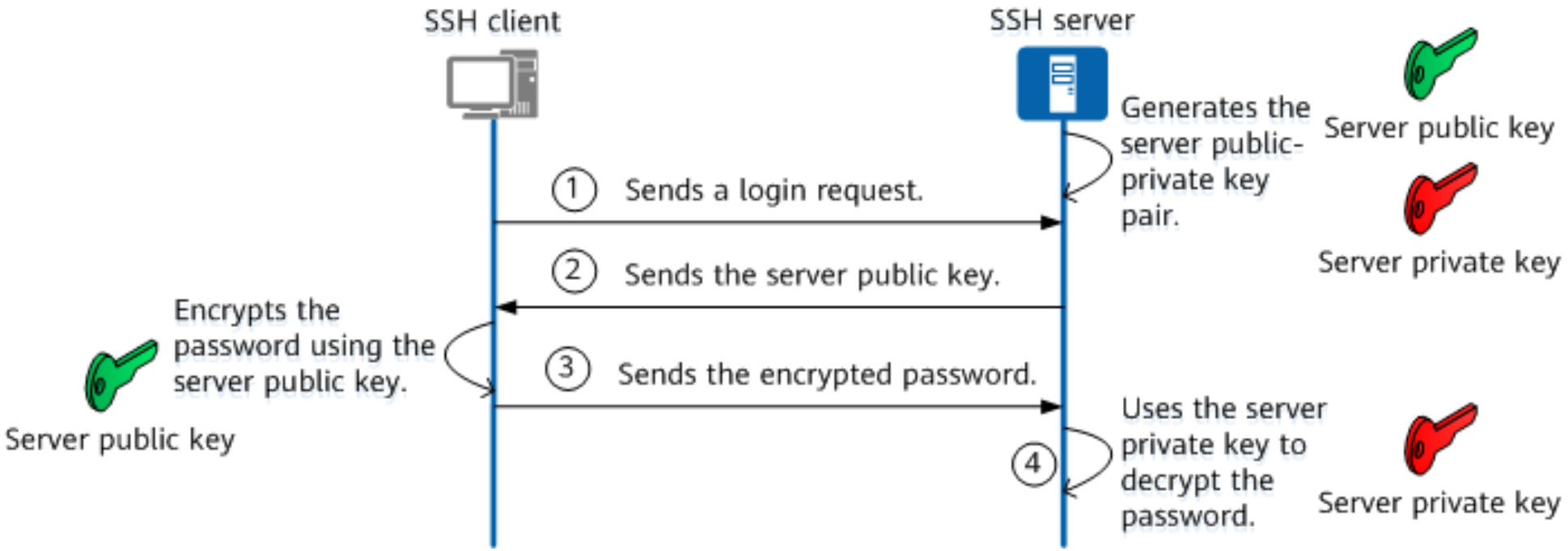
Лекция 2

Основы настройки локальной сети

Основные файлы для настройки сети

/etc/hostname	Настройка имени компьютера
/etc/hosts	Настройка разрешения доменных имен
/etc/resolv.conf	Настройка адресов серверов имен, к которым имеет доступ данная система
/etc/network/interfaces	Настройка сетевых интерфейсов
/etc/apt/sources.list	Настройка списка репозиториев

ssh



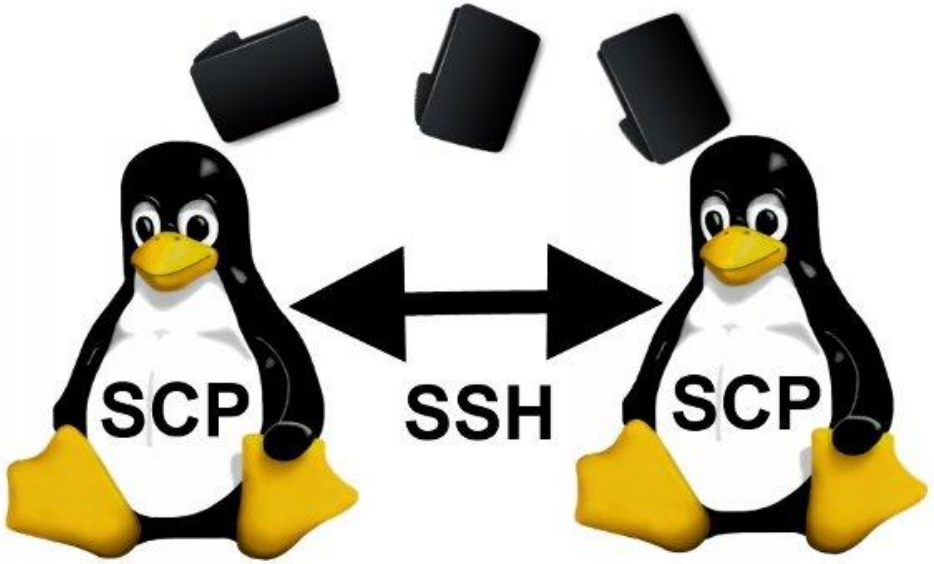
server (192.168.122.2)

client (192.168.122.3)

```
sab@server: /$ ssh-keygen
sab@server: /$ ssh-copy-id adminstd@192.168.122.2
```

```
sab@client: /$ ssh adminstd@192.168.122.2
Last Login: ...
sab@server: /$ hexdump 1.txt
```

scp



$ scp опции пользователь1@хост1:файл пользователь2@хост2:файл

```
sab@server: /$ scp ServerAddress/File user@hostname:ClientAddress
```

```
sab@server: /$ scp user@hostname:address/ClientAddress/File LocalAddress
```

Подключение репозитория

Репозиторий – хранилище, где содержатся данные.

/etc/apt/sources.list

deb *[trusted=yes]* ftp://ru01srw024/astra/ 1.7_x86-64 main contrib non-free

Можно не указывать, если добавить ключ

```
sab@client: /$ sudo apt-key add repo_gpg.key //adding key from server
```

```
sab@client: /$ sudo apt update
sab@client: /$ sudo apt-get install <package>
```

Потренируемся!

Задание 1.

1. Настройка виртуальных машин
    1. Для настройки виртуальных машин используйте значения, указанные в таблице ниже
    2. Проанализируйте файл /etc/network/interfaces. Что содержится в нем?
    3. Проверьте соединение между клиентом и сервером
    4. Настройте сеть, дополнив необходимые конфигурационные файлы, шлюз по умолчанию – 192.168.122.1
2. Попробуйте отправить ping с сервера на клиент используя доменное имя. Получилось ли это сделать? Если нет, то исправьте это.

Задание 2.

1. Подключитесь к ftp серверу. Загрузите с него публичный ключ для подключения к локальному репозиторию.
2. Подключите сервер к локальному репозиторию, расположенному по адресу 10.10.0.191 в директории astra.
3. Скачайте с локального репозитория утилиту tree и установите ее на ваш сервер.

Задание 3.

1. На сервере запустите службу ssh и добавьте ее в автозагрузку
2. На клиенте настройте аутентификацию по ключам с сервером
3. Подключитесь к серверу с машины клиента и создайте в директории /home/study файл с содержимым «Hello world!»
4. Скопируйте с сервера на клиент (командой scp) файл созданный в предыдущем пункте файл.

	Astra001	**Astra002**
hostname	Ваши инициалы_server	Ваши инициалы_client
ip address	192.168.122.(N в группе + 1)	192.168.122.(N в группе + 2)

Администрирование локальных вычислительных сетей

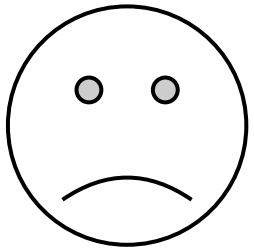
Domain Name System (DNS)

Система доменных имен

История системы доменных имен

1970е

Число узлов >> 100



Предельная нагрузка системы

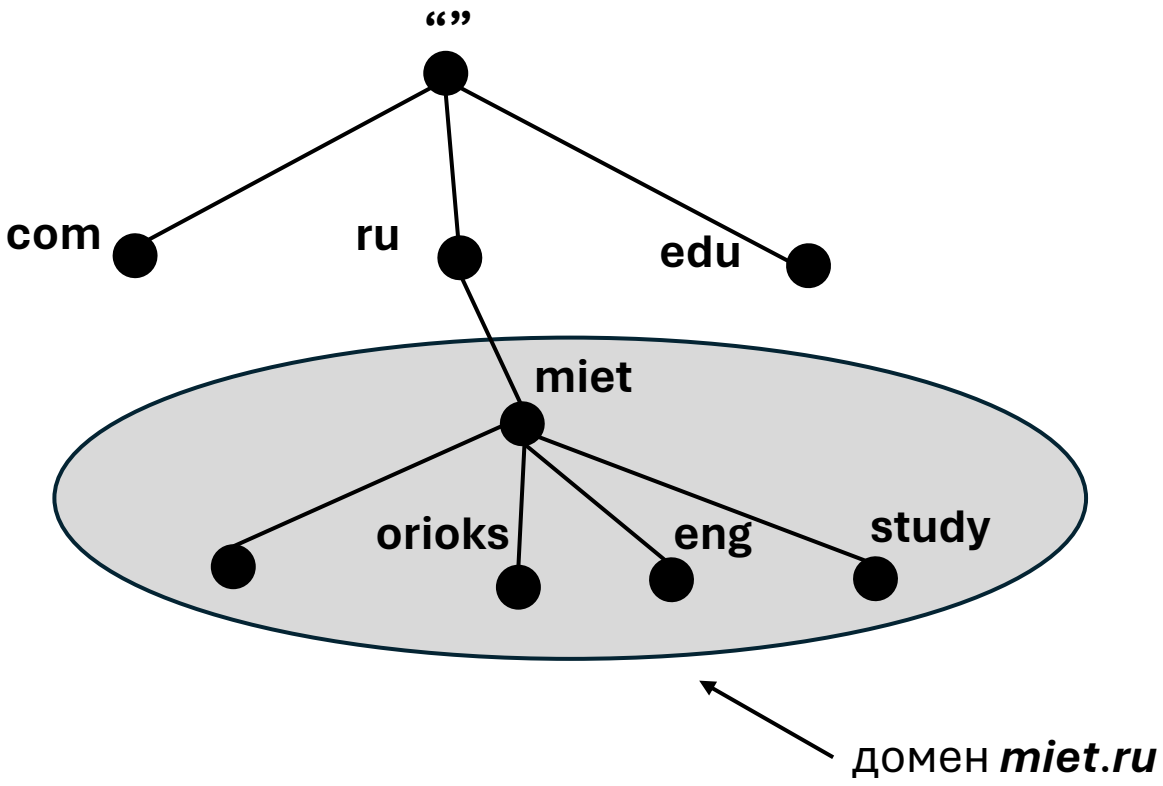
Конфликты имен

Проблемы синхронизации

Число узлов: несколько сотен
Файл пополняли вручную и синхронизировали

Общее устройство DNS

DNS — это распределенная база данных, которая содержит информацию о компьютерах, включенных в сеть Internet.



● - Узел ветви дерева DNS

Домен - ветвь дерева DNS

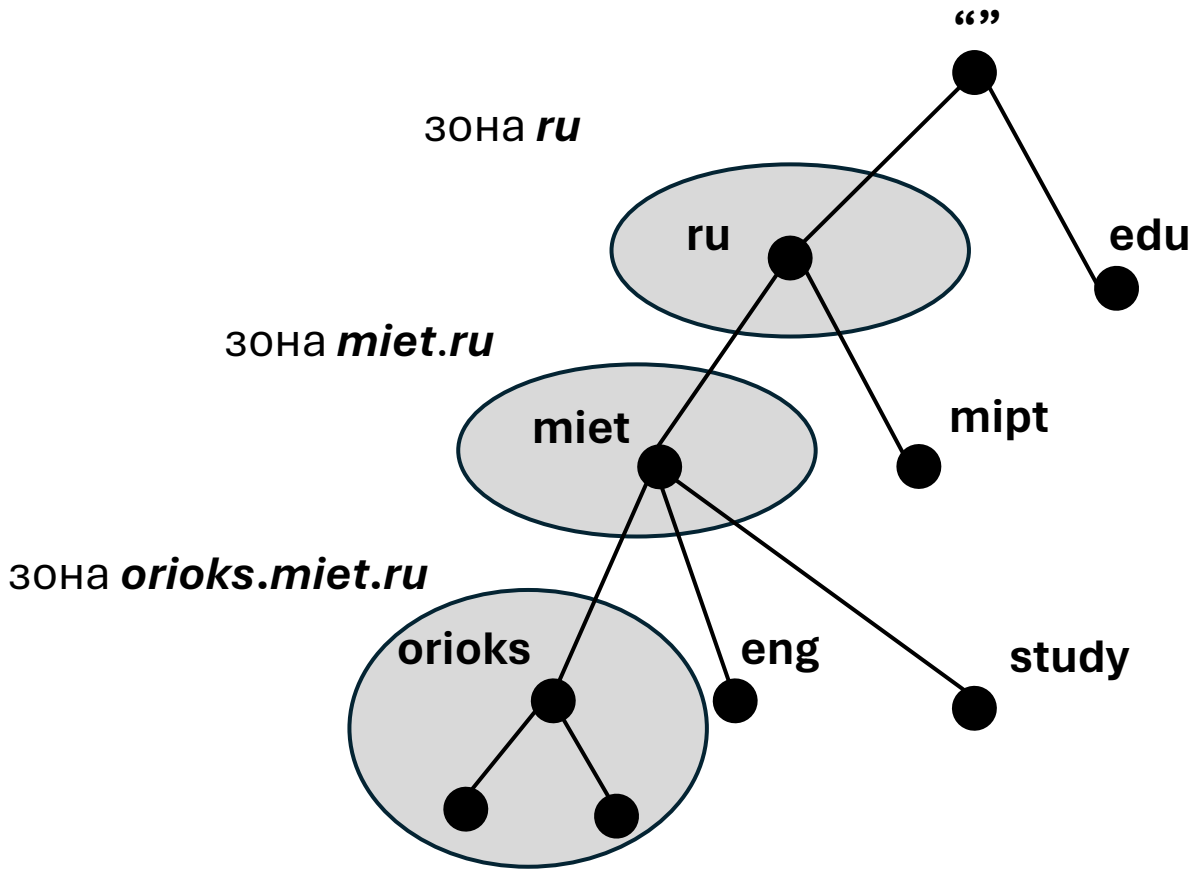
com, ru, edu – домены

Домены включают в себя узлы и поддомены

Полное ***доменное имя*** произвольного узла дерева – это последовательность меток в пути от этого узла до корня.

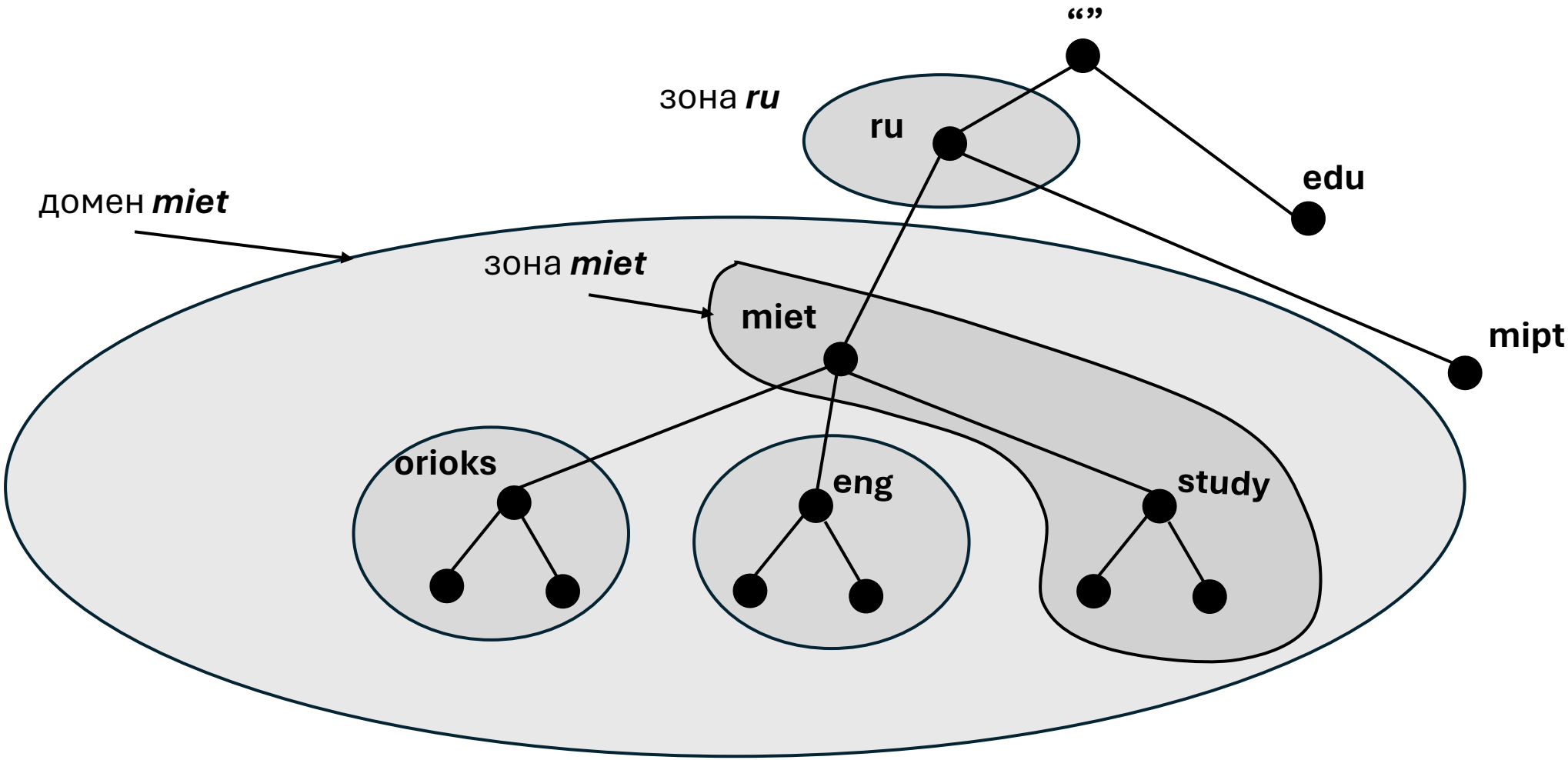
Доменная зона

Доменная зона - независимо администрируемая часть пространства имен

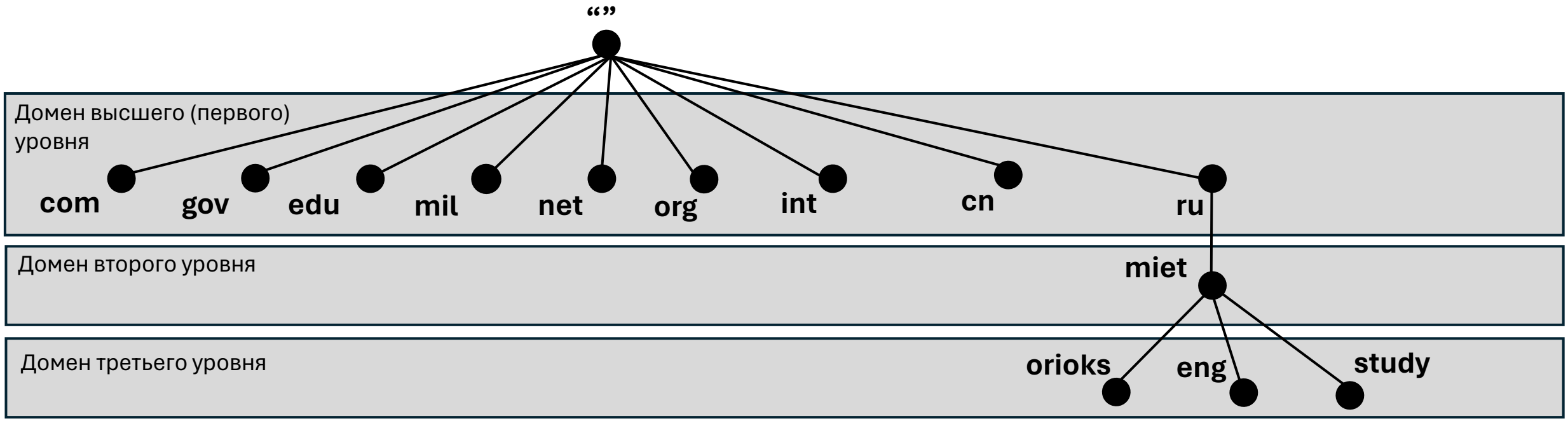


Доменная зона ≠ домен

Доменная зона - независимо администрируемая часть пространства имен



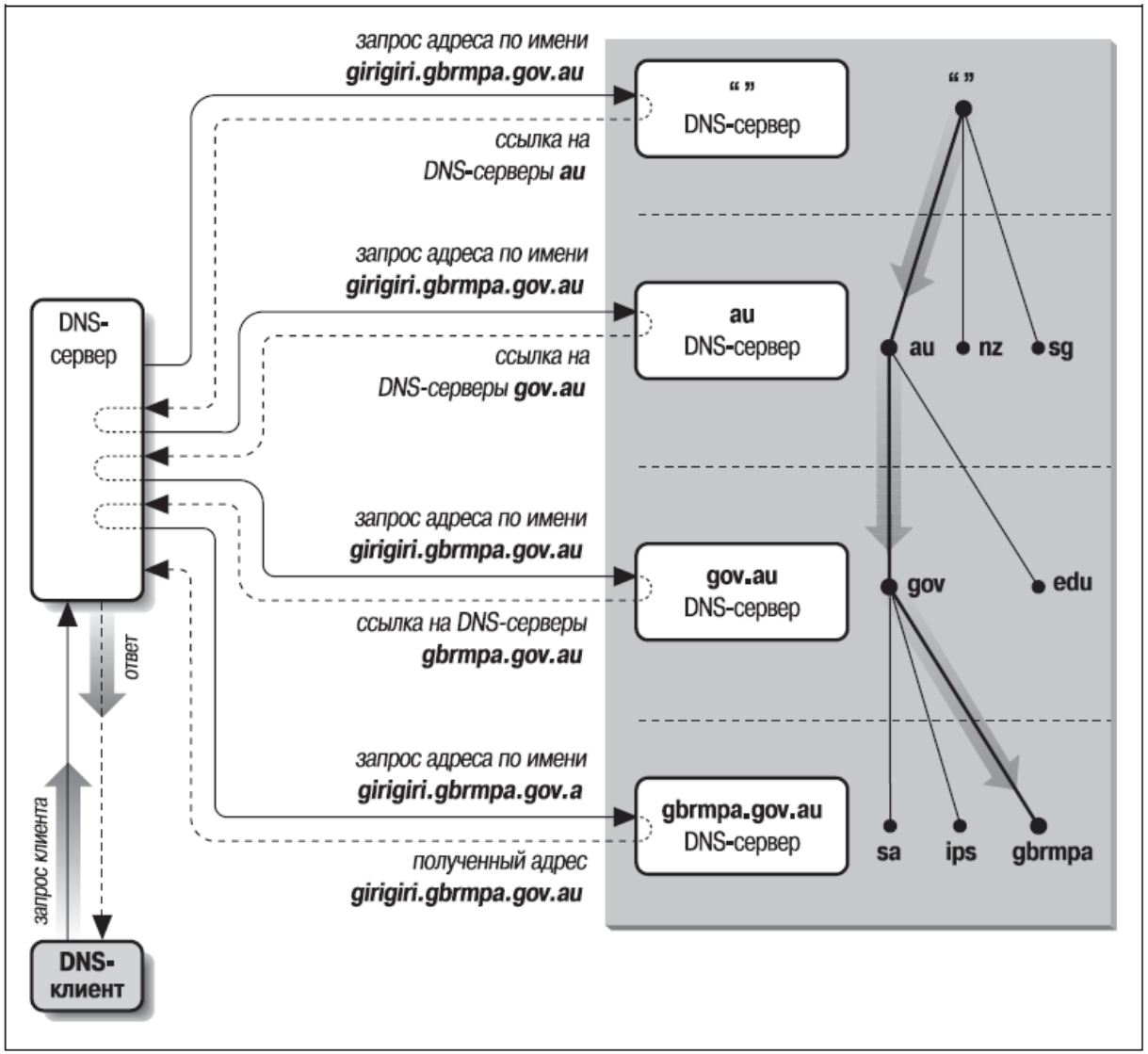
Пространство доменных имен сети Интернет



Процесс разрешения имен

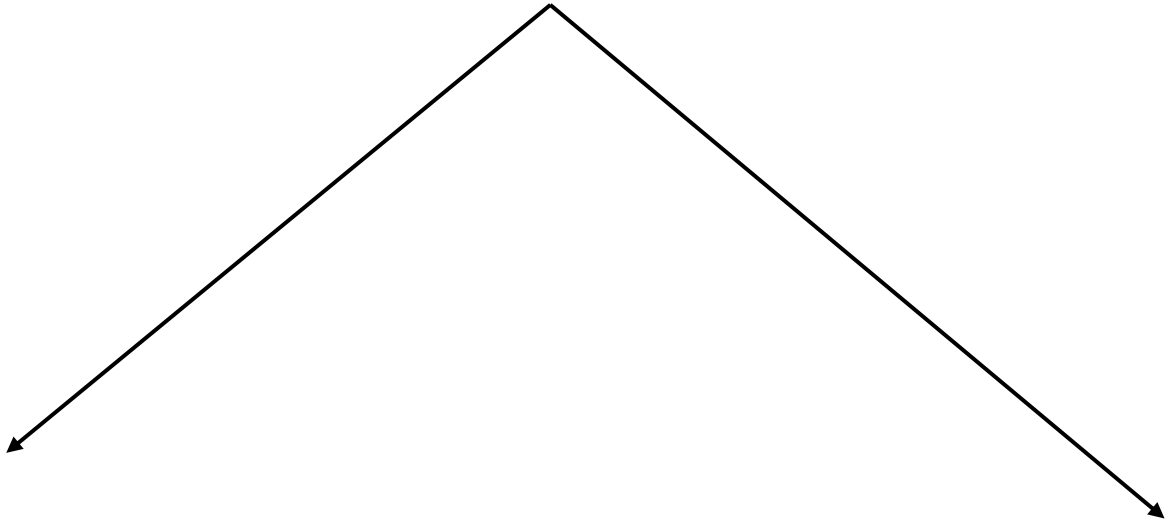
Сервер DNS может работать в рекурсивном или итеративном (нерекурсивном) режиме:

- Итеративный сервер выдает клиенту только **авторитетные** ответы о зонах, которые он сопровождает, и информацию о корневых серверах
- Рекурсивный сервер выдает клиенту, либо авторитетный ответ о зонах, которые он сопровождает, либо неавторитетный ответ из кэша, либо неавторитетный ответ, для получения которого он проходит всю иерархию авторитетных серверов, начиная с корневой зоны



Разновидности DNS серверов

DNS определяет два типа серверов:

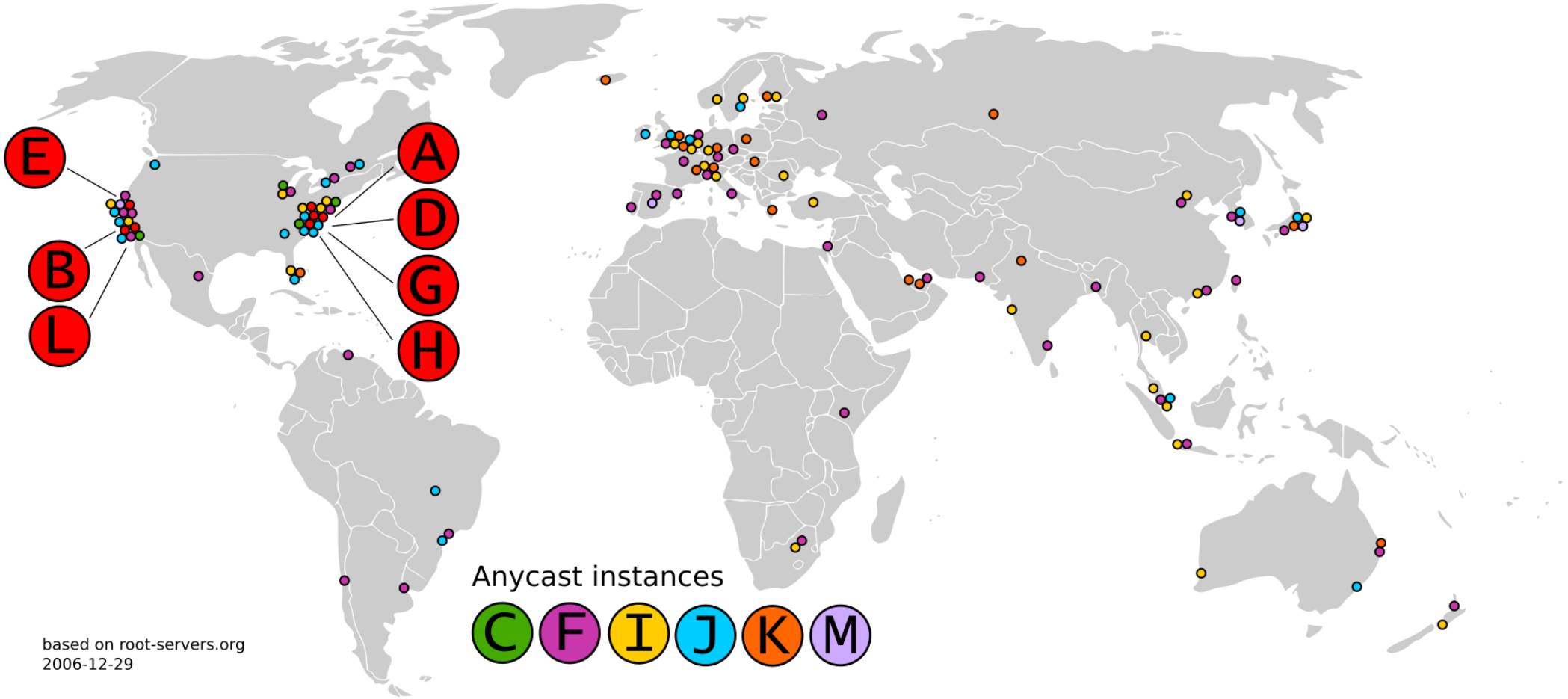


master name server
(ведущий сервер имен)

slave name server
(подчиненный сервер имен)

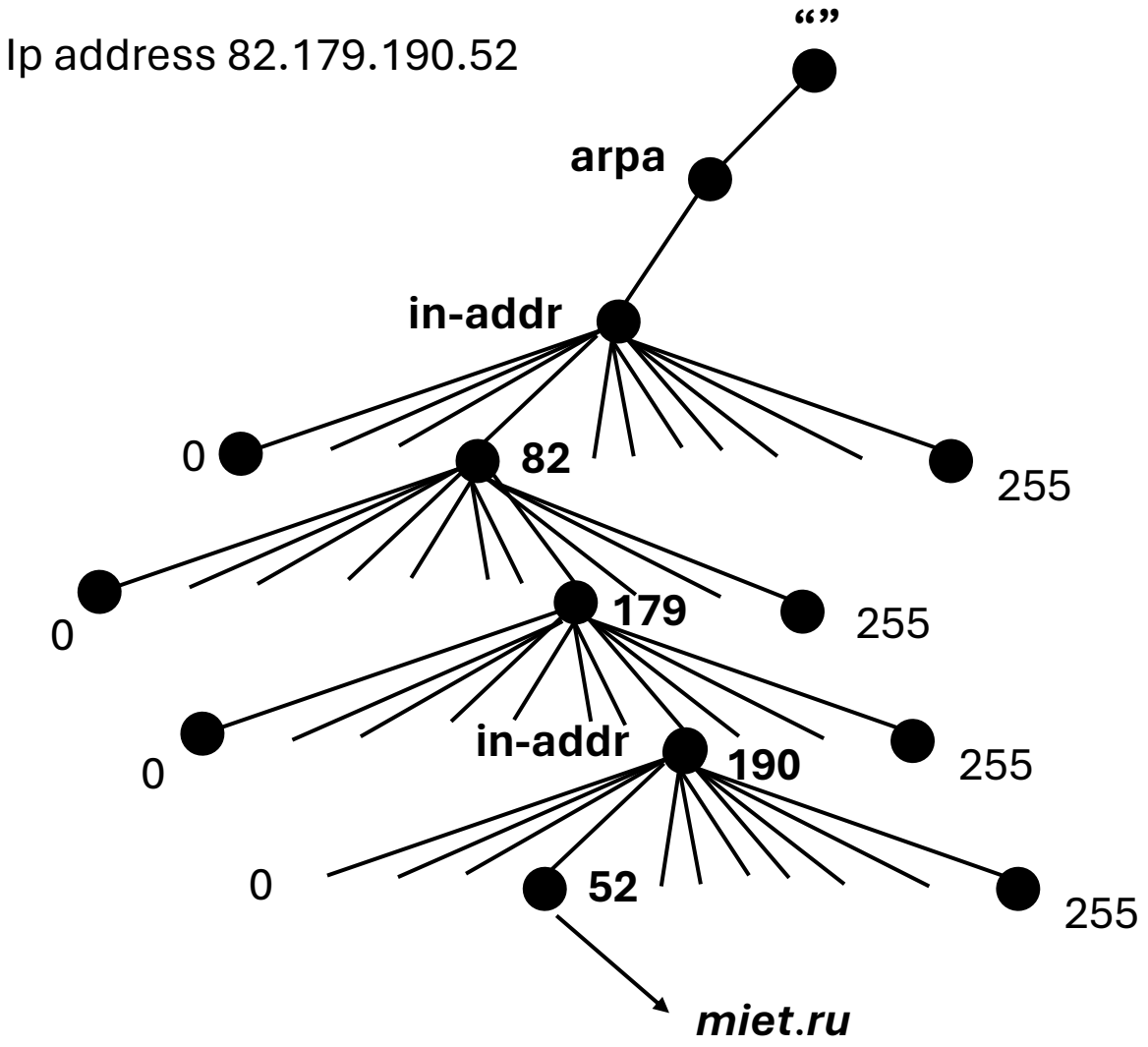
Ведущий и подчиненные серверы имен дают **авторитетные** (уполномоченные) ответы на запросы о ресурсных записях для зоны, которую они сопровождают

Корневые DNS серверы



Географическое распределение корневых серверов DNS (2006)

Отображение адресов в имена



Узлам домена *in-addr.arpa* в качестве меток присваиваются числа в нотации ip-адреса. Домен *in*addr.arpa* может содержать до 256 поддоменов, каждый из которых будет соответствовать одному из возможных значений первого октета IPадреса. Каждый из этих поддоменов может содержать до 256 собственных поддоменов, каждый из которых будет соответствовать одному из возможных значений второго октета.

Настройка DNS

Способ 1 - /etc/hosts

```
# [network]
127.0.0.1       localhost
127.0.1.1       astra001

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
```

Настройка DNS (bind9)

Способ 2 - /etc/bind

/etc/bind/named.conf — основной конфигурационный файл
/etc/bind/named.conf.options — содержит глобальные параметры
/etc/bind/named.conf.local — описание зон
/etc/bind/named.conf.default-zones

/etc/bind/named.conf

```
тип_секции [имя_секции] {
    установки;
    установки;
    …
};
```

Типы секций: zone, options, logging, acl, key, channel
Установки секций:

- directory имя_каталога
- forwarders {список_адресов_DNS_серверов;}
- recursion yes/no
- notify yes/no
- allow-query {список_адресов;}
- allow-transfer {список_адресов;}
- allow-recursion{список_адресов;}
- type тип_сервера (master/slave)
- file "имя_файла"

/etc/bind/named.conf.local

```
zone "mpsu.stu" {
        type master;
        file "/etc/bind/zones/db.miet.stu";
};

zone "122.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/zones/192.168.122";
};
```

Типы секций: zone, options, logging, acl, key, channel
Установки секций:

- directory имя_каталога
- forwarders {список_адресов_DNS_серверов;}
- recursion yes/no
- notify yes/no
- allow-query {список_адресов;}
- allow-transfer {список_адресов;}
- allow-recursion{список_адресов;}
- type тип_сервера (master/slave)
- file "имя_файла"

Файл ресурсных записей

```
$TTL 604800
mpsu.stu.       IN      SOA srv.mpsu.stu. admin@mpsu.stu (
                                2024030901      ;Последовательный номер
                                3h              ;Обновление
                                1h              ;Повтор попытки обновления
                                1w              ;Устаревание через 1 неделю
                                1h              ;TTL отрицательного кэширования
                                )
mpsu.stu.       IN      NS      srv.mpsu.stu.

srv.mpsu.stu.   IN      A       192.168.122.2
cli.mpsu.stu.   IN      A       192.168.122.3

neighbor        IN      CNAME   cli.mpsu.stu.
```

ВРЕМЯ ЖИЗНИ
ИМЯ КЛАСС ТИП ДАННЫЕ

ВРЕМЯ ЖИЗНИ - $TTL (Например, $TTL 604800)
ИМЯ – имя ресурсной записи (Например, miet.stu)
КЛАСС – IN (от INternet)
ТИП – (SOA, A, AAAA, PTR, NS, MX, CNAME, SRV)
ДАННЫЕ – могут состоять из нескольких полей

Файл ресурсных записей (обратный просмотр)

```
$TTL 604800
122.168.192.in-addr.arpa.    IN    SOA srv.mpsu.stu. admin@mpsu.stu (
                                        2024030901      ;Последовательный номер
                                        3h              ;Обновление
                                        1h              ;Повтор попытки обновления
                                        1w              ;Устаревание через 1 неделю
                                        1h              ;TTL отрицательного кэширования
                                        )
122.168.192.in-addr.arpa.    IN    NS    srv.mpsu.stu.

2                            IN    A     srv.mpsu.stu.
3                            IN    A     cli.mpsu.stu.
```

В случае ошибок – можно изучить логи bind

```
sab@server: /$ cat /var/log/syslog
```

Проверка настройка файлов bind

```
sab@server: /$ named-checkconf /etc/bind/named.conf
sab@server: /$ named-checkzone mpsu.stu /etc/bind/zones/db.mpsu.stu
sab@server: /$ named-checkzone 122.168.192.in-addr.arpa /etc/bind/zones/db.mpsu.stu
```

Если все хорошо, то можно перезапустить bind9

```
sab@server: /$ systemctl restart bind9
```

В случае ошибок – можно изучить логи bind

```
sab@server: /$ less /var/log/syslog
```

Проверим корректность работы

```
sab@client: /$ nslookup srv.mpsu.stu
sab@client: /$ nslookup 192.168.122.2
```

Чтобы на машине клиента указать адрес DNS сервера, необходимо его добавить в файле resolv.conf