



# **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			Подсистема процессов
Виртуальная файловая система		Сетевой стек	Межпроцессное взаимодействие
Драйверы файловых систем	Драйверы символьных устройств		Управление памятью
Драйверы блочных устройств		Драйверы сетевых устройств	Планировщик процессов
			Архитектурно-зависимый код

Аппаратное обеспечение





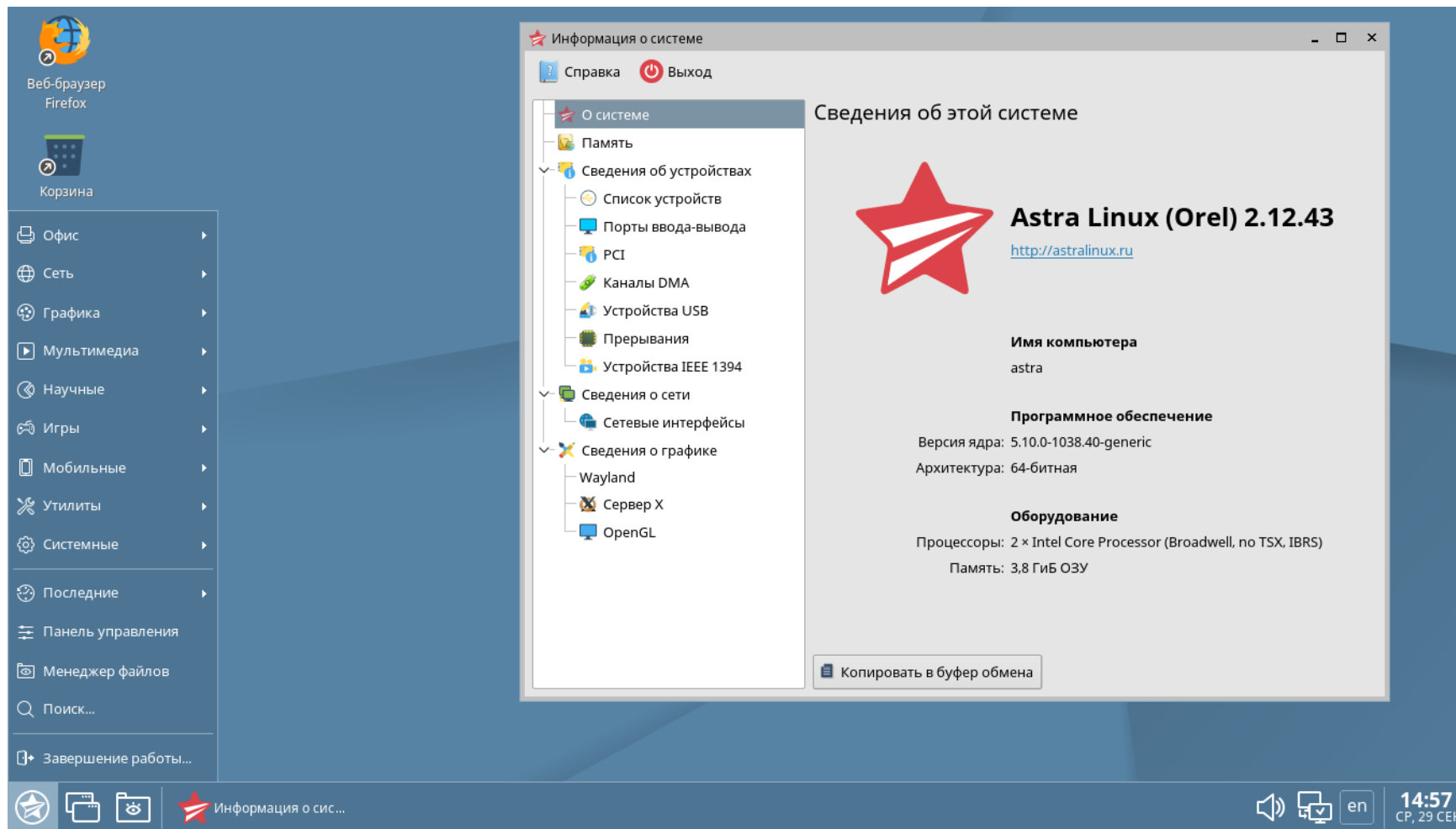
# Дистрибутивы Linux



- 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»

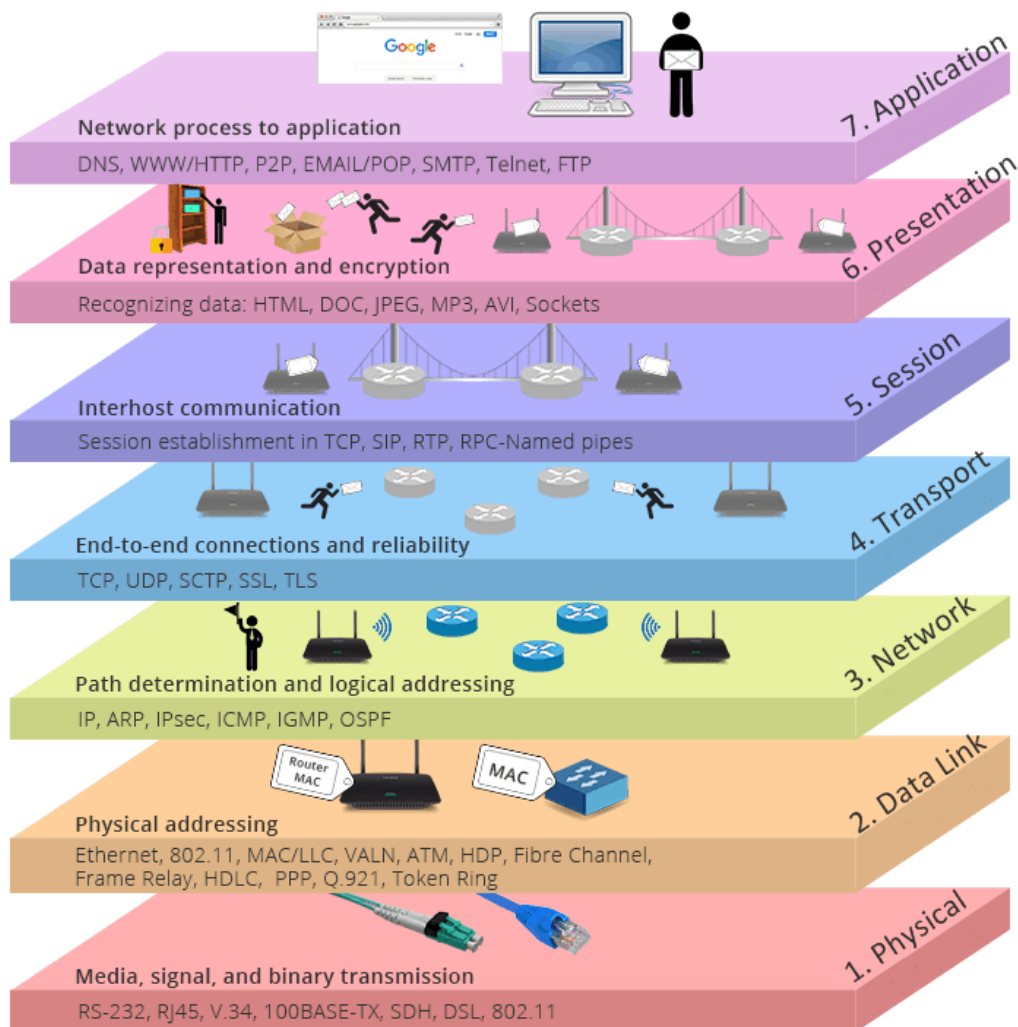
### Максимальный уровень защищенности

Рекомендовано для обработки  
информации любой категории  
доступа в государственных  
информационных системах

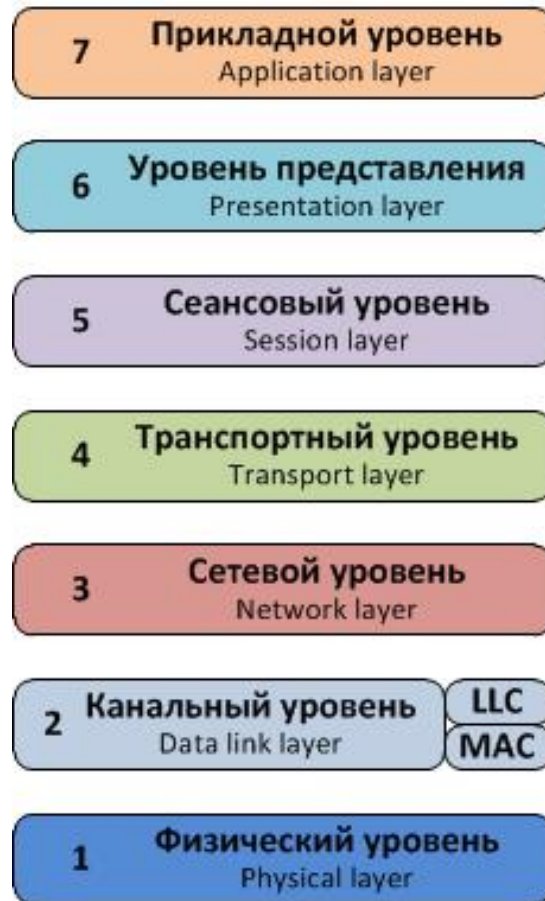




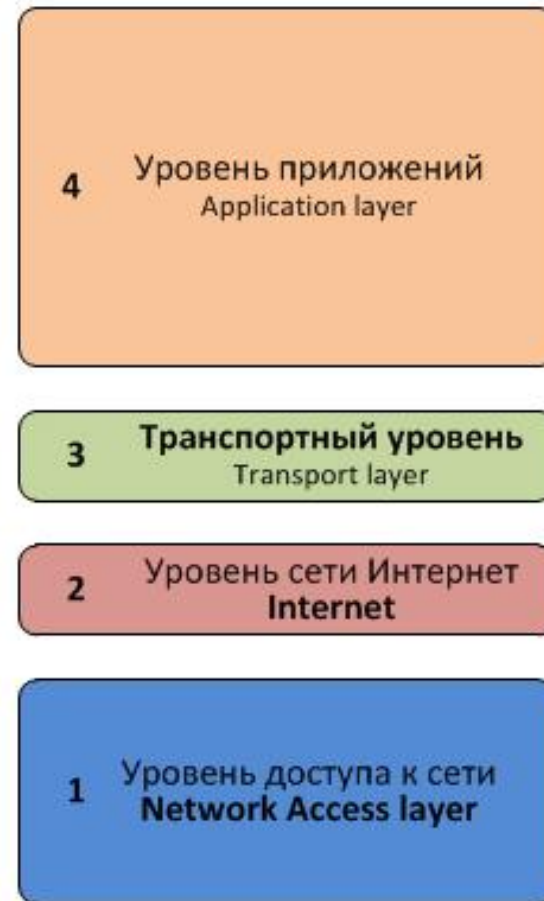
# Модель OSI



## OSI

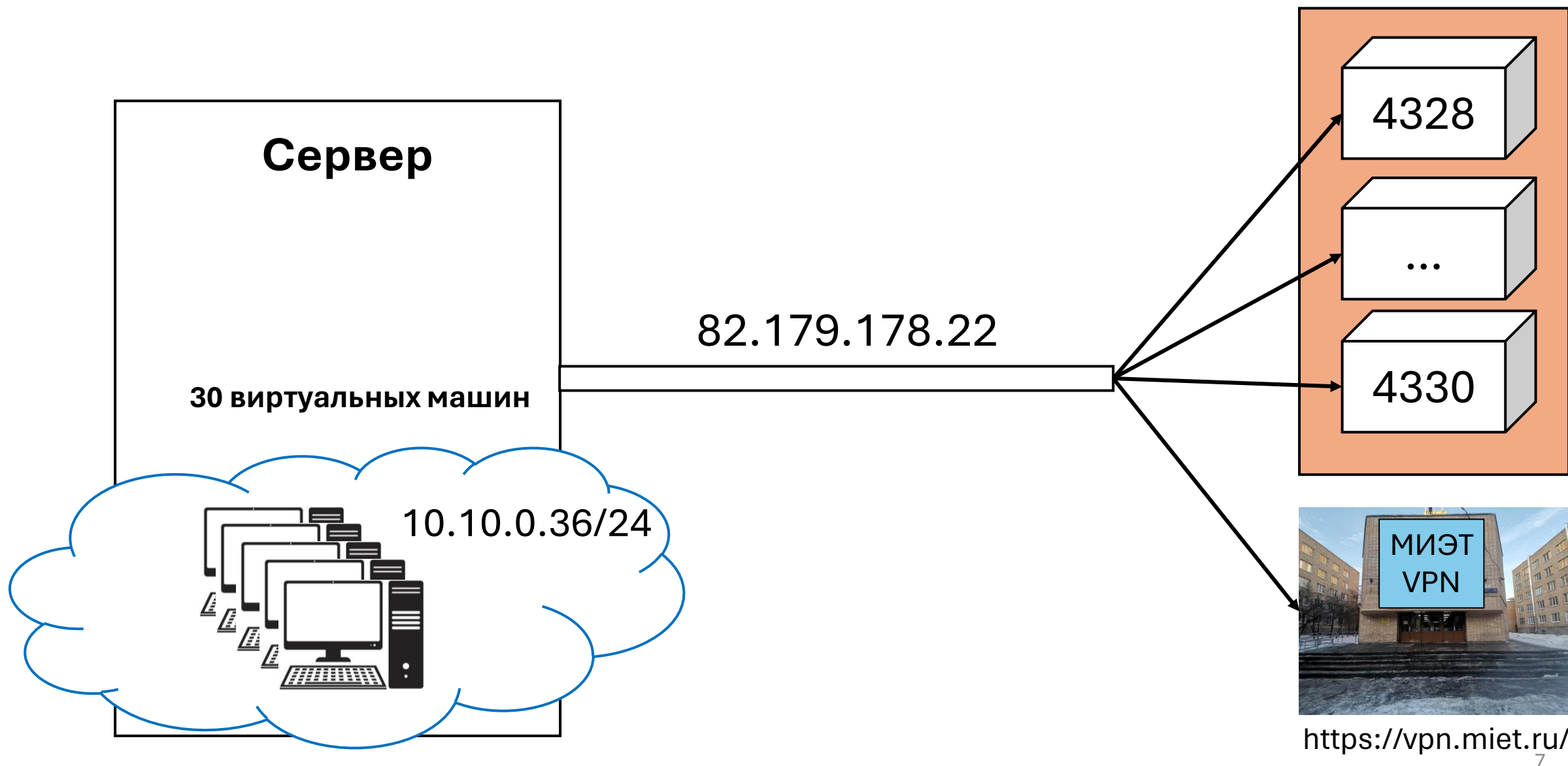


## TCP/IP (DOD)





# Как подключиться к Astra Linux?





## Как подключиться к Astra Linux?

В браузере ПК вводим адрес виртуальной машины:

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

|

**Логин:**

study0XX

**Пароль:**

qwerty22

**Например:**

**Вариант 10**

82.179.178.22:5813/vnc.html

study010

qwerty22

XX – ваш номер по списку





## ftp сервер

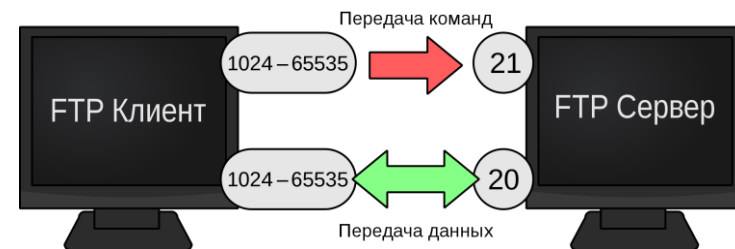
В менеджере файлов Астра Линукс вводим:

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



## Гrep (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 повториться 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 А или В

```
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
<b>-c</b>	Вырезать указанную последовательность символов
<b>-d</b>	Указать символ разделения (по умолчанию TAB)
<b>-f</b>	Выбрать символы, разделенные определенным символом
<b>-s</b>	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
<b>-c</b>	Вывести размер файла в байтах
<b>-m</b>	Вывести количество символов в файле
<b>-l</b>	Вывести количество строк в файле
<b>-w</b>	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
<b>-u</b>	Вывести исключительно те строки, у которых нет повторов.
<b>-d</b>	Вывести все строки, удаляя дубликаты
<b>-D</b>	Вывести только повторяющиеся строки.
<b>-c</b>	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
<b>-F fs</b>	Указать символ разделения (по умолчанию пробел)
<b>-v var=val</b>	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 C – ветвления, циклы

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

Некоторые команды	Описание
<b>print</b>	Вывести строку целиком
<b>length(N)</b>	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

- !<sup>^</sup> 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**

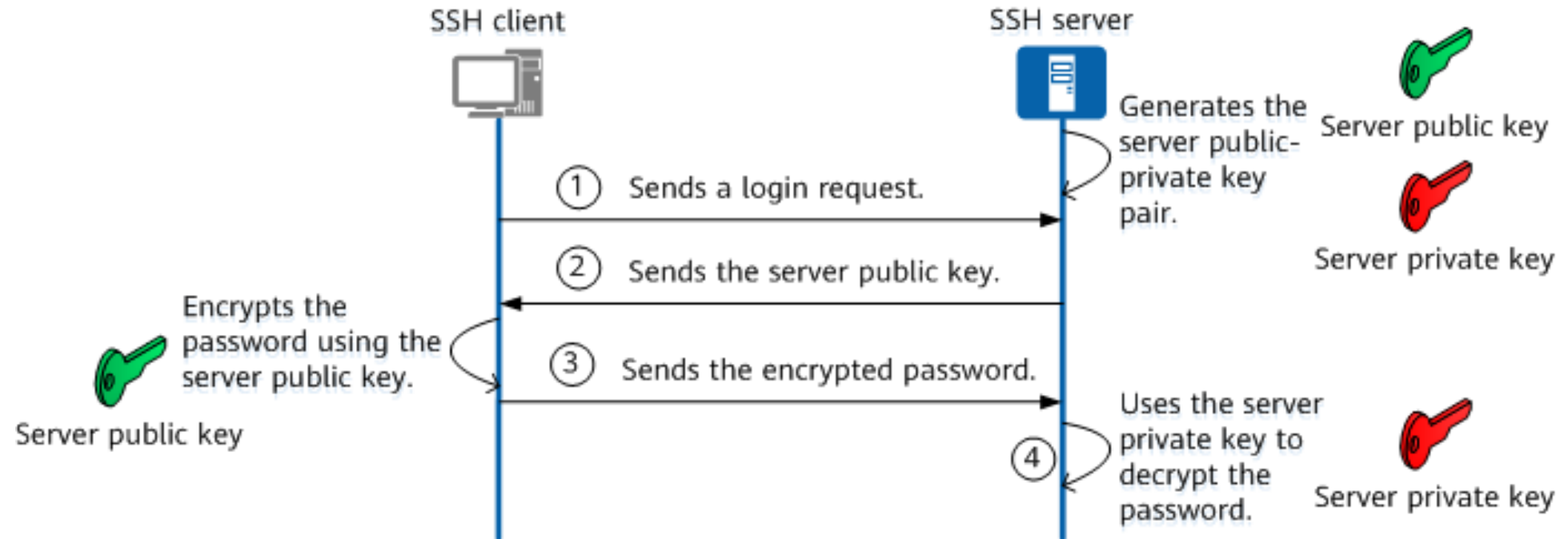


# Основные файлы для настройки сети

/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)*

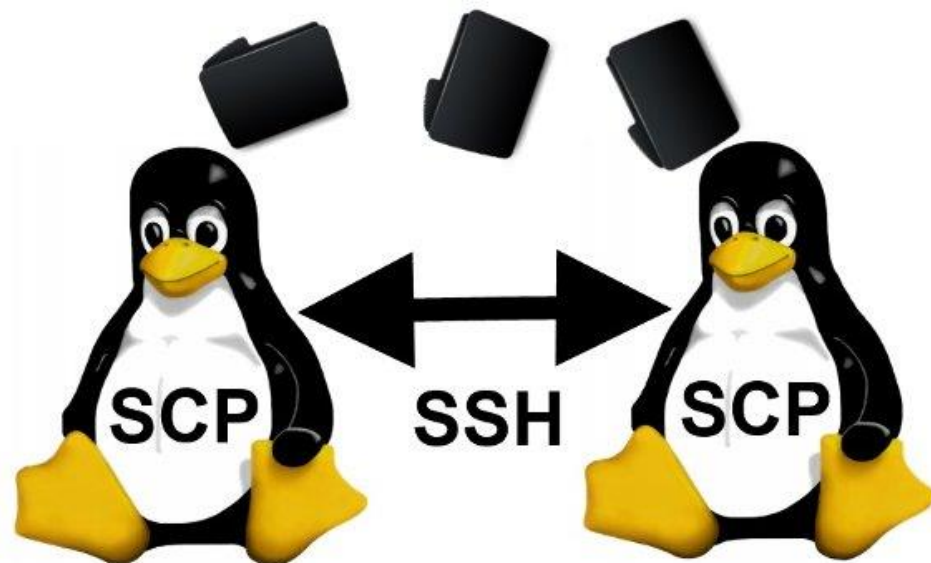
*client (192.168.122.3)*

```
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)



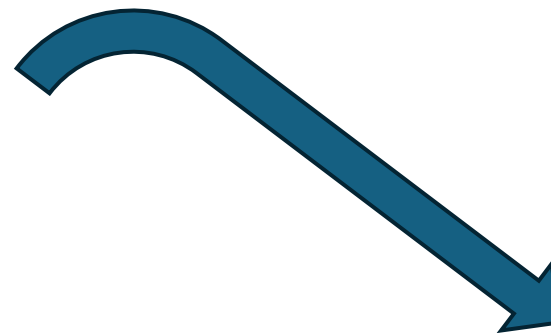
# **Администрирование локальных вычислительных сетей**

**Domain Name System (DNS)**

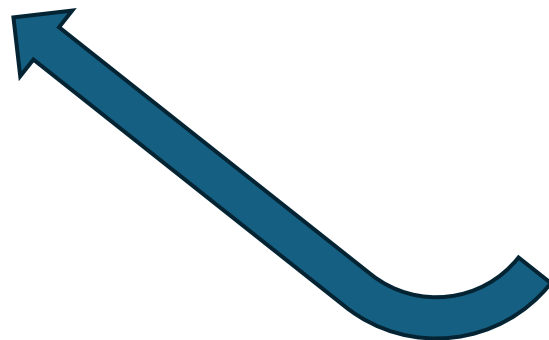


# Система доменных имен

82.179.190.52



miet.ru



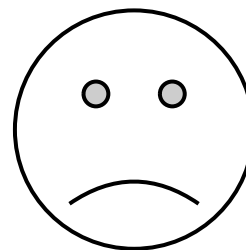


# История системы доменных имен

1970е



Число узлов  $\gg 100$



Предельная нагрузка системы

Конфликты имен

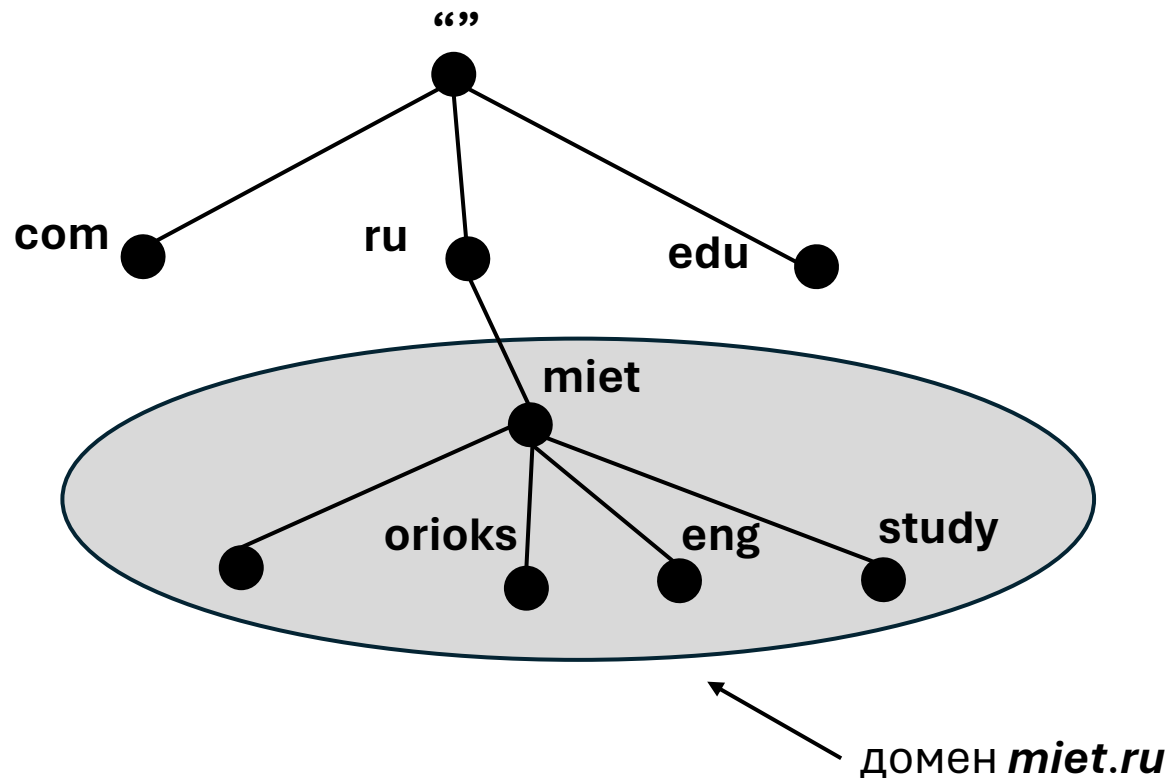
Проблемы синхронизации

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

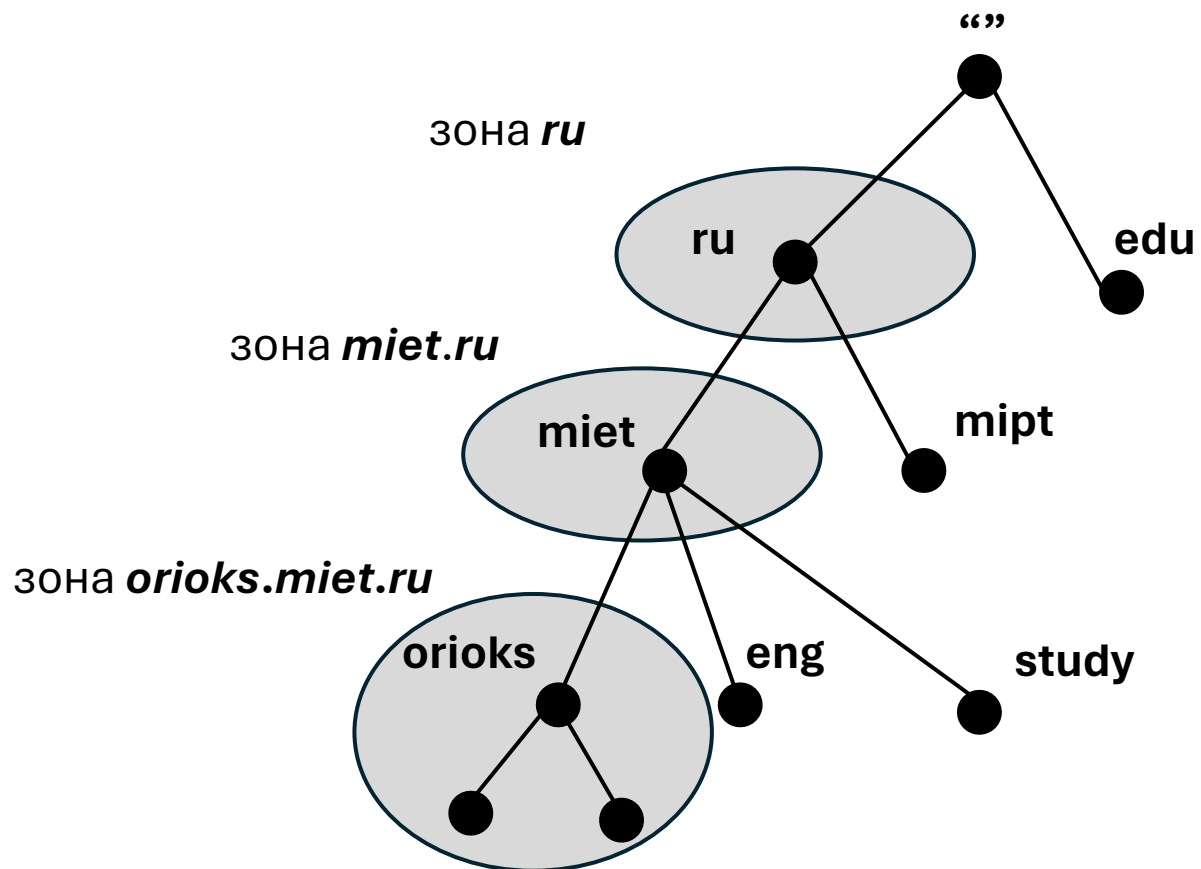
Домены включают в себя узлы и поддомены

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



# Доменная зона

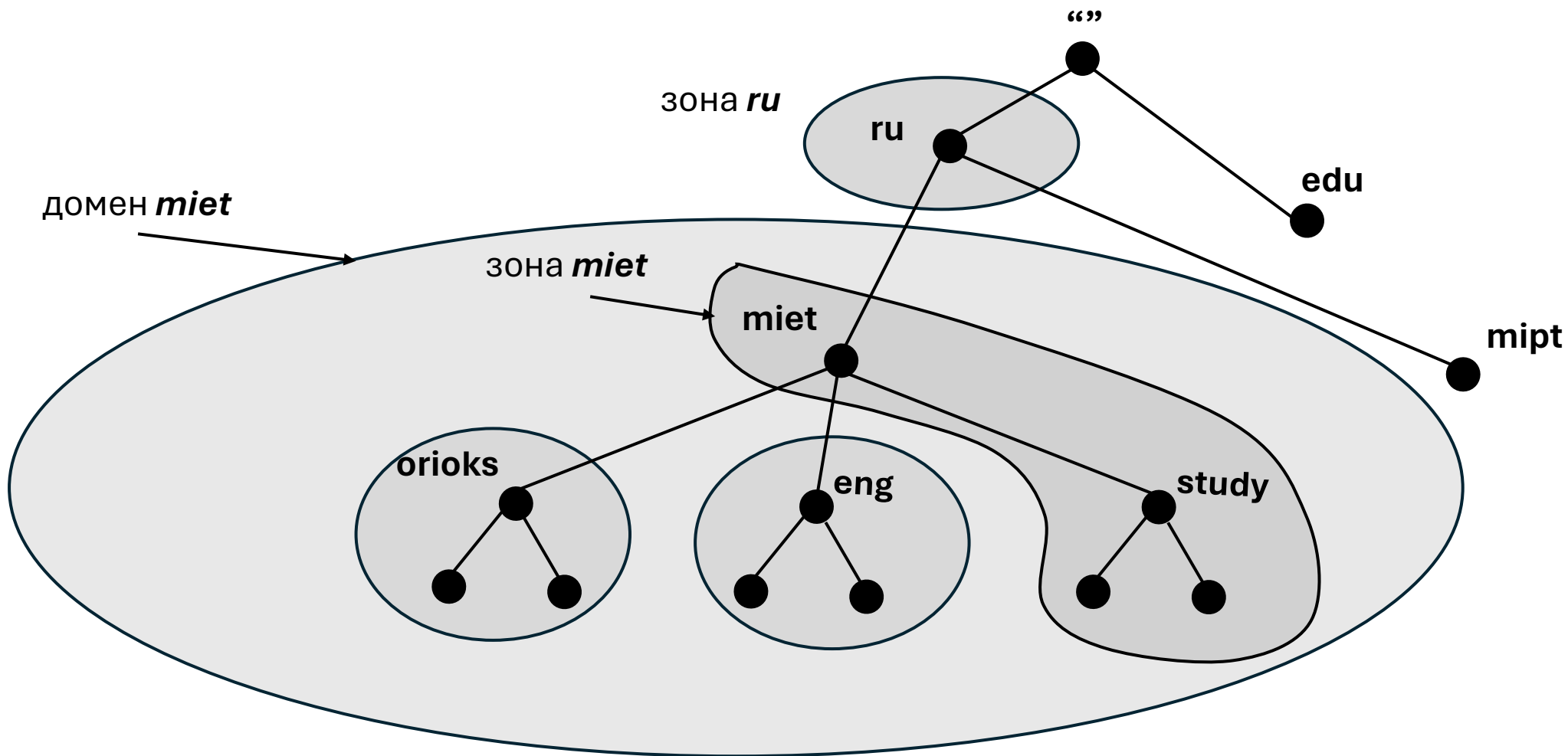
**Доменная зона** - независимо администрируемая часть пространства имен





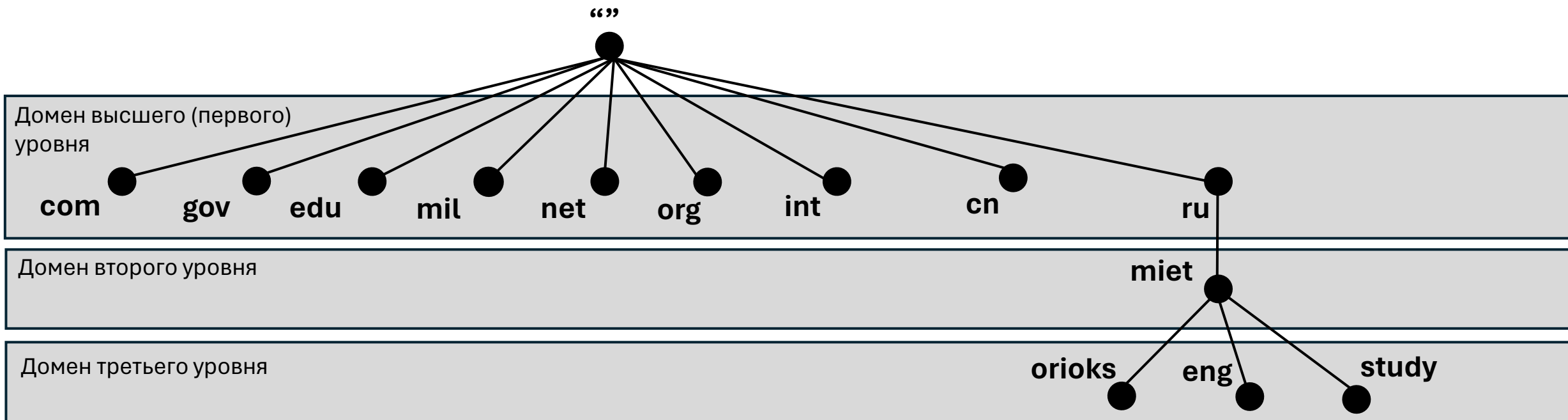
# Доменная зона ≠ домен

**Доменная зона** - независимо администрируемая часть пространства имен





# Пространство доменных имен сети Интернет



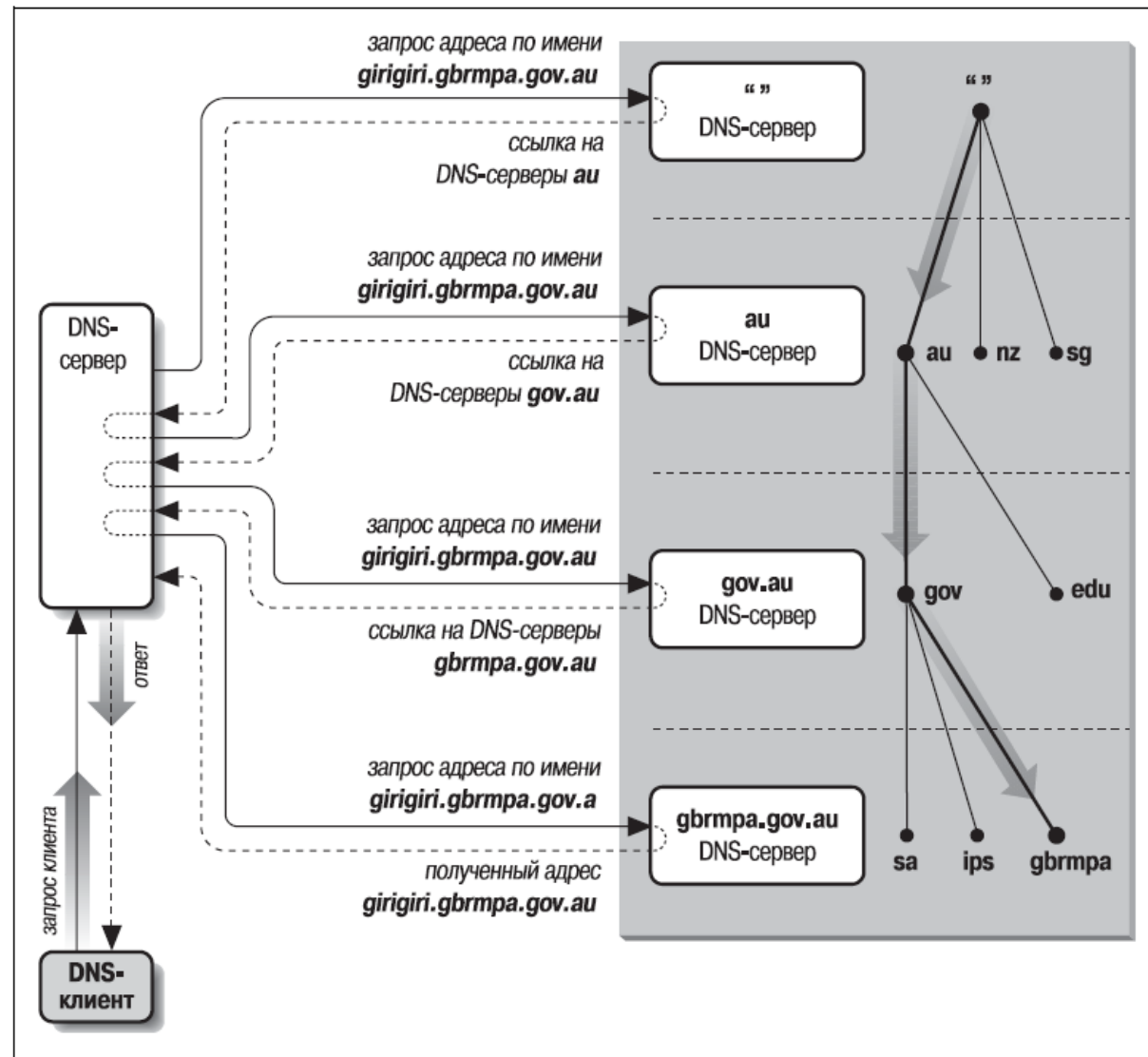




# Процесс разрешения имен

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

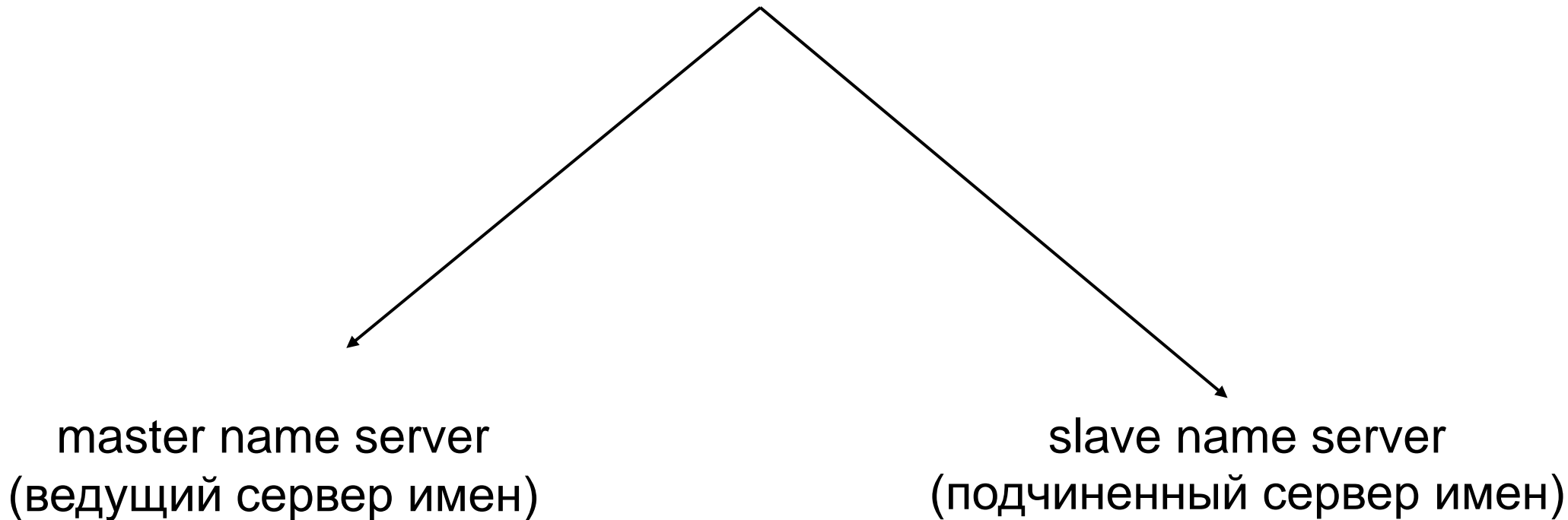
- Итеративный сервер выдает клиенту только **авторитетные** ответы о зонах, которые он сопровождает, и информацию о корневых серверах
- Рекурсивный сервер выдает клиенту, либо авторитетный ответ о зонах, которые он сопровождает, либо неавторитетный ответ, для получения которого он проходит всю иерархию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





# Разновидности DNS серверов

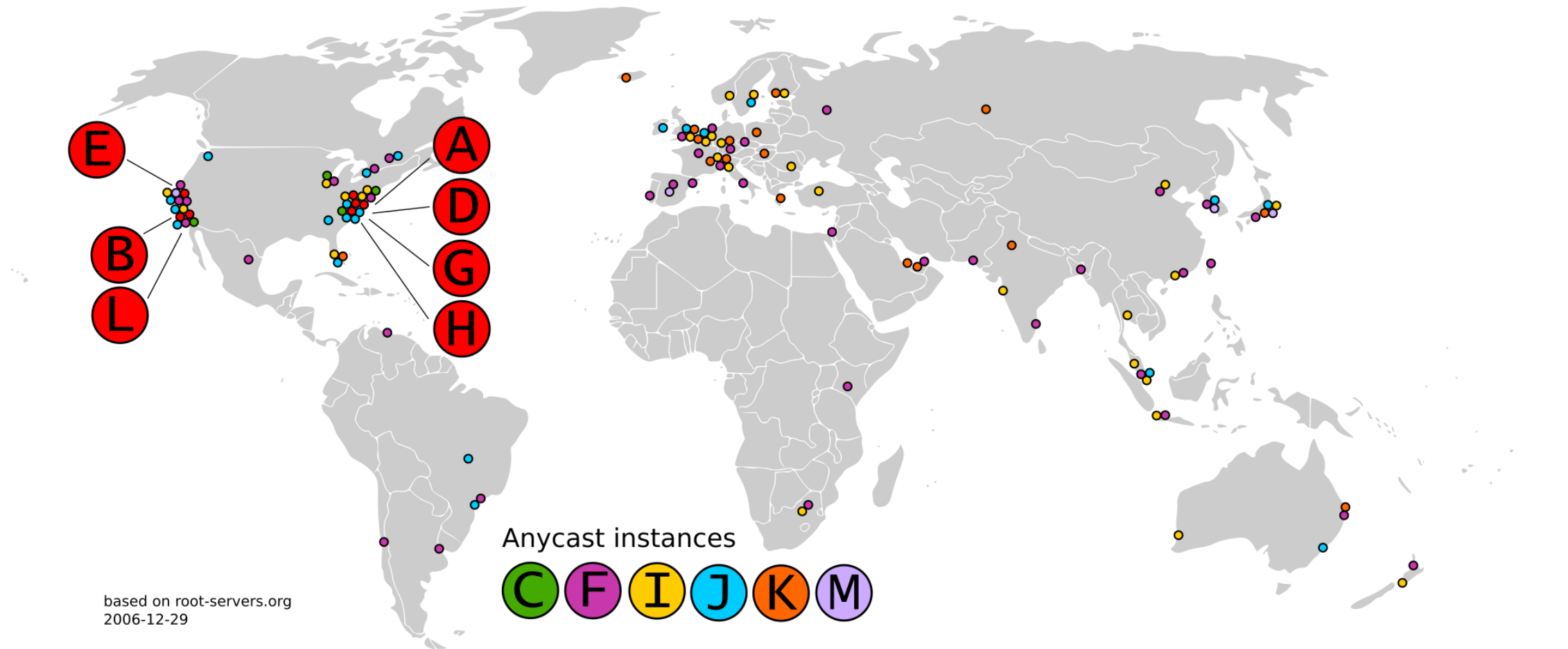
DNS определяет два типа серверов:



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



# Корневые DNS серверы

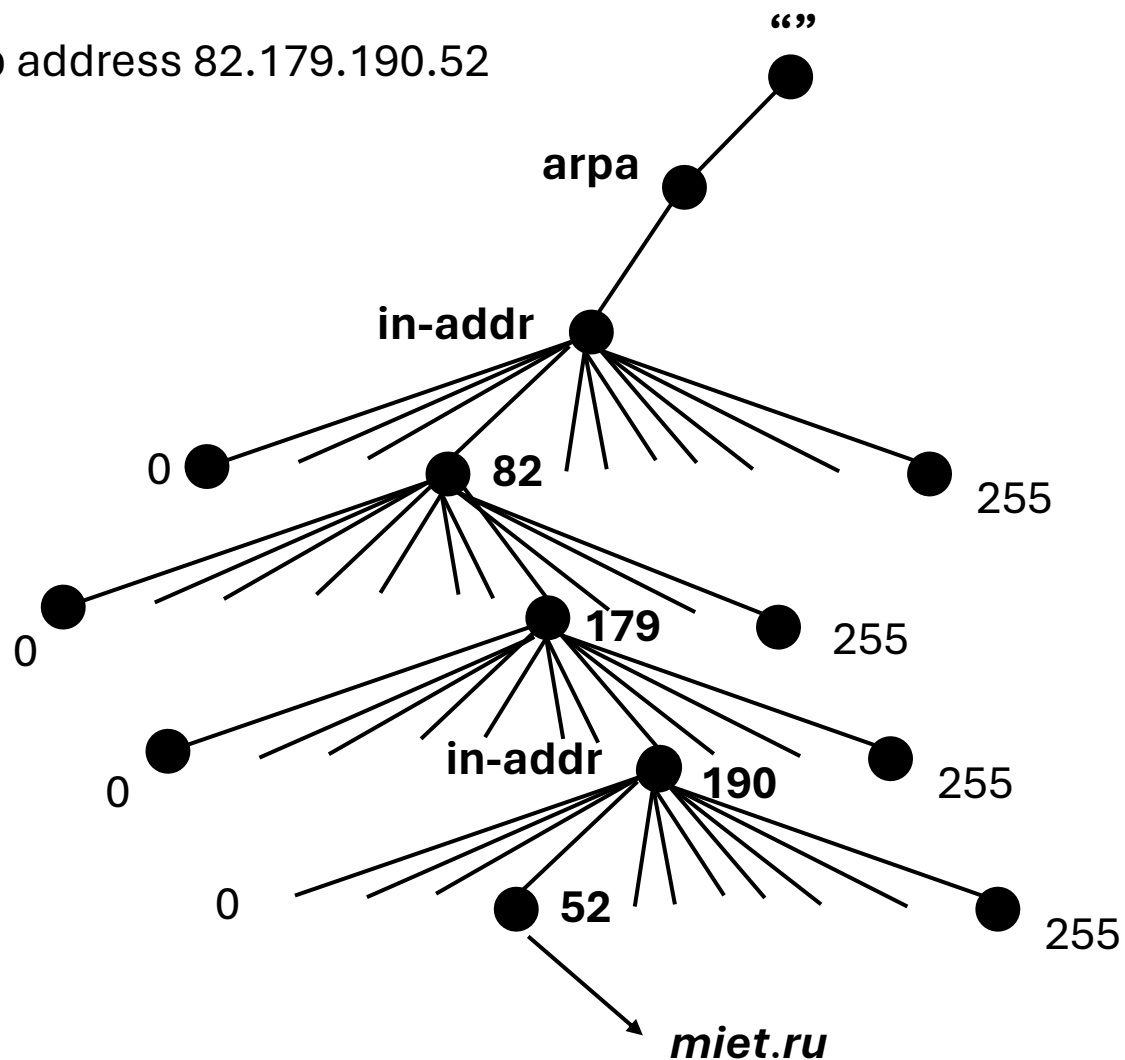


Географическое распределение корневых серверов DNS (2006)



# Отображение адресов в имена

Ip address 82.179.190.52



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 IP-адреса. 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@mpsु.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	PTR	srv.mpsu.stu.	
3	IN	PTR	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  
Для указания домена используйте ключевое слово domain



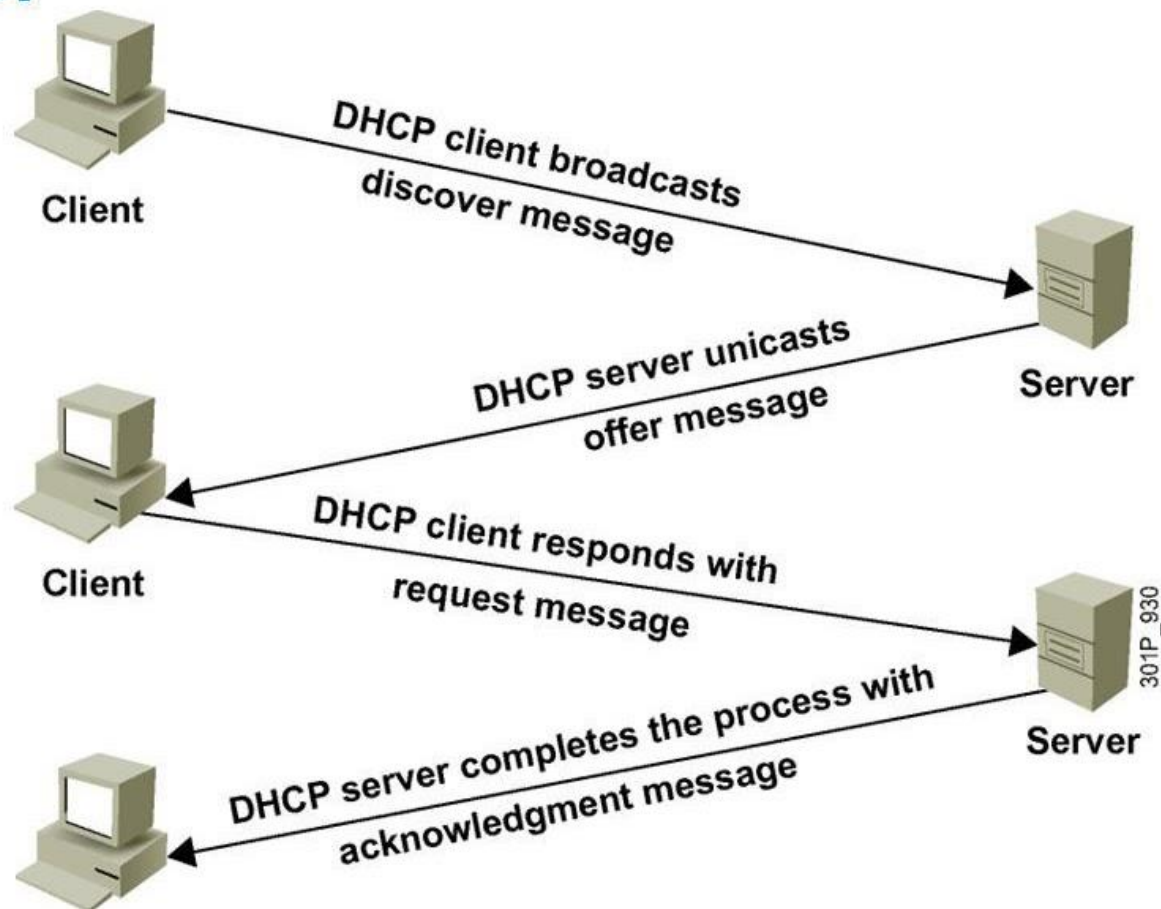
# **Администрирование локальных вычислительных сетей**

**Dynamic Host Configuration Protocol  
(DHCP)**



# Оснoвы DHCP

## DHCP



DHCP (Dynamic Host Configuration Protocol) — протокол, позволяющий хостам автоматически получать IP-адреса и другие сетевые настройки



# Предварительная настройка DHCP

Необходимо отключить DHCP на виртуальном мосте KVM

```
root@ru01wks070:/home/adminmsc# virsh net-edit default
```

```
Select an editor. To change later, run 'select-editor'.
```

1. /bin/nano <---- easiest
2. /usr/bin/vim.basic
3. /usr/bin/mcedit
4. /usr/bin/vim.tiny
5. /usr/bin/code

```
Choose 1-5 [1]: 1
```

```
XML-конфигурация сети default была изм
```

```
<!--
```

```
WARNING: THIS IS AN AUTO-GENERATED FILE. CHANGES TO IT ARE LIKELY TO BE  
OVERWRITTEN AND LOST. Changes to this xml configuration should be made using:
```

```
    virsh net-edit default  
or other application using the libvirt API.
```

```
-->
```

```
<network>
```

```
  <name>default</name>
```

```
  <uuid>2efaf743-b0bf-4b03-9ff2-4c1bc48c8644</uuid>
```

```
  <forward mode='nat' />
```

```
  <bridge name='virbr0' stp='on' delay='0' />
```

```
  <mac address='52:54:00:34:e1:7a' />
```

```
  <ip address='192.168.122.1' netmask='255.255.255.0'>
```

```
  </ip>
```

```
</network>
```



# Настройка DHCP

```
sab@server: /$ apt-get install isc-dhcp-server
```

Основные конфигурационные файлы:

/etc/default/isc-dhcp-server  
/etc/dhcp/dhcpd.conf

*установка значений по умолчанию  
настройка сервера dhcp*

В файле со значениями по умолчанию необходимо выбрать интерфейс, на котором будет работать сервер  
Например, INTERFACESv4="eth0 eth1"



# /etc/dhcp/dhcpd.conf

- default-lease-time задает время лизинга по умолчанию (в секундах)
- max-lease-time задает максимальное время лизинга
- Директива option определяет, какие TCP/IP настройки будут передаваться клиенту:
  - option domain-name имя\_домена; – задает имя домена
  - option domain-name-servers список\_DNS\_серверов; - определяет используемые DNS серверы
  - option routers IP\_адрес; – определяет маршрут по умолчанию
- Для описании топологии используются секции:
  - subnet адрес\_сети netmask сетевая\_маска {...} - описание сети
  - host имя\_хоста {...}- описание хоста
- Директива range внутри секции subnet определяет, какой диапазон адресов будет использоваться для назначения динамических адресов клиентам
- Директивы hardware и fixed-address внутри секции host используются для задания статических адресов. MAC адрес сетевого интерфейса сопоставляется получаемому IP адресу





# Пример настройки DHCP

*Пример задания динамических адресов:*

```
subnet 192.168.1.0 netmask 255.255.255.0
{
    range 192.168.1.100 192.168.1.150;
}
```

/etc/dhcp/dhcpd.conf

*Пример задания статических адресов:*

```
host comp1.example.ru
{
    hardware ethernet 00:DE:AA:10:35:BE;
    fixed-address 192.168.1.151;
}
```



# Настройка на клиенте

*Сбросить динамический адрес на клиенте*

```
sab@server: /$ dhclient -r
```

*Запросить новый динамический адрес*

```
sab@server: /$ dhclient
```

*Настройка сети*

*/etc/network/interfaces/*

*auto eth0*

*iface eth0 inet dhcp*



# **Администрирование локальных вычислительных сетей**

**Веб-сервер на основе Apache**



# Протокол HTTP

- Прикладной протокол HyperText Transfer Protocol – протокол передачи гипертекста
- Находится на **прикладном** уровне OSI
- Не имеет встроенных средств шифрования
- Использует протокол TCP и порт 80/TCP

## Структура запроса

МЕТОД РЕСУРС HTTP/1.1

### МЕТОДЫ:

- GET
- POST
- PUT
- DELETE
- HEAD
- и т.д.



## Структура ответа

HTTP/1.1 КОД\_СОСТОЯНИЯ РАСШИФРОВКА

### КОДЫ СОСТОЯНИЯ:

- 1XX – информационные
- 2XX – успешное завершение
- 3XX – перенаправление
- 4XX – ошибки клиента
- 5XX – ошибки сервера



# Протокол HTTP

```
sab@server: /$ telnet www.miet.ru 80
```

```
GET / HTTP/1.1  
Host: www.miet.ru
```

```
HTTP/1.1 301 Moved Permanently  
Server: nginx  
Date: Fri, 05 Apr 2024 23:31:11 GMT  
Content-Type: text/html  
Content-Length: 162  
Connection: keep-alive  
Keep-Alive: timeout=20  
Location: https://www.miet.ru/
```

```
<html>  
<head><title>301 Moved Permanently</title></head>  
<body>  
<center><h1>301 Moved Permanently</h1></center>  
<hr><center>nginx</center>  
</body>  
</html>
```



# Протокол HTTP

```
sab@server: /$ telnet www.wikipedia.ru 80
```

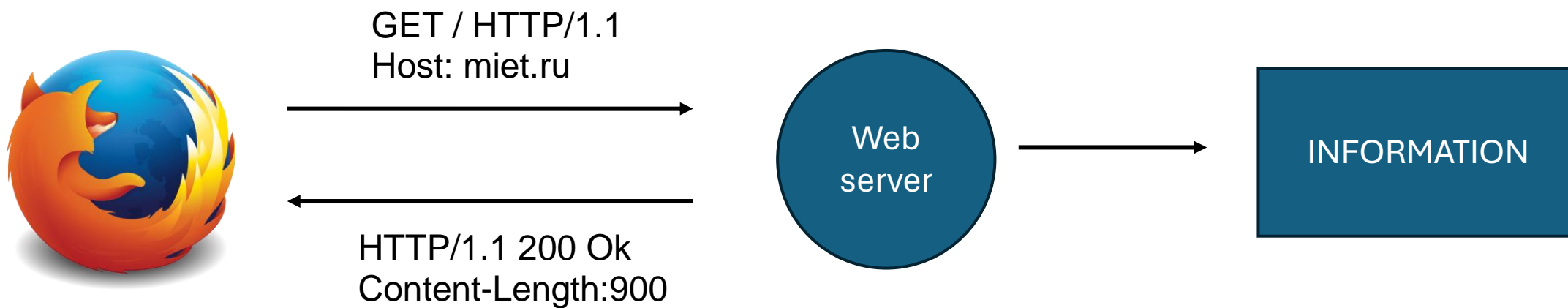
```
HEAD / HTTP/1.1  
Host: www.wikipedia.ru
```

```
HTTP/1.1 301 Moved Permanently  
Server: nginx  
Date: Fri, 05 Apr 2024 23:31:11 GMT  
Content-Type: text/html  
Content-Length: 162  
Connection: keep-alive  
Keep-Alive: timeout=20  
Location: https://www.miet.ru/
```

```
<html>  
<head><title>301 Moved Permanently</title></head>  
<body>  
<center><h1>301 Moved Permanently</h1></center>  
<hr><center>nginx</center>  
</body>  
</html>
```



# Работа веб-сервера





# Веб-сервер







# Первоначальная настройка

*Отключение режима безопасности*  
*AstraMode off*

`/etc/apache2/apache2.conf`

```
sab@server: /$ mkdir /var/www/mpsu.stu/  
sab@server: /$ touch /var/www/mpsu.stu/index.html
```

```
<!DOCTYPE html>  
<html>  
  <body>  
    <h1>Hello MIET</h1>  
    <p>I love MPSU!</p>  
  </body>  
</html>
```



# Настройка сервера

```
sab@server: /$ touch /etc/apache2/sites-available/web.mpsu.stu.conf
```

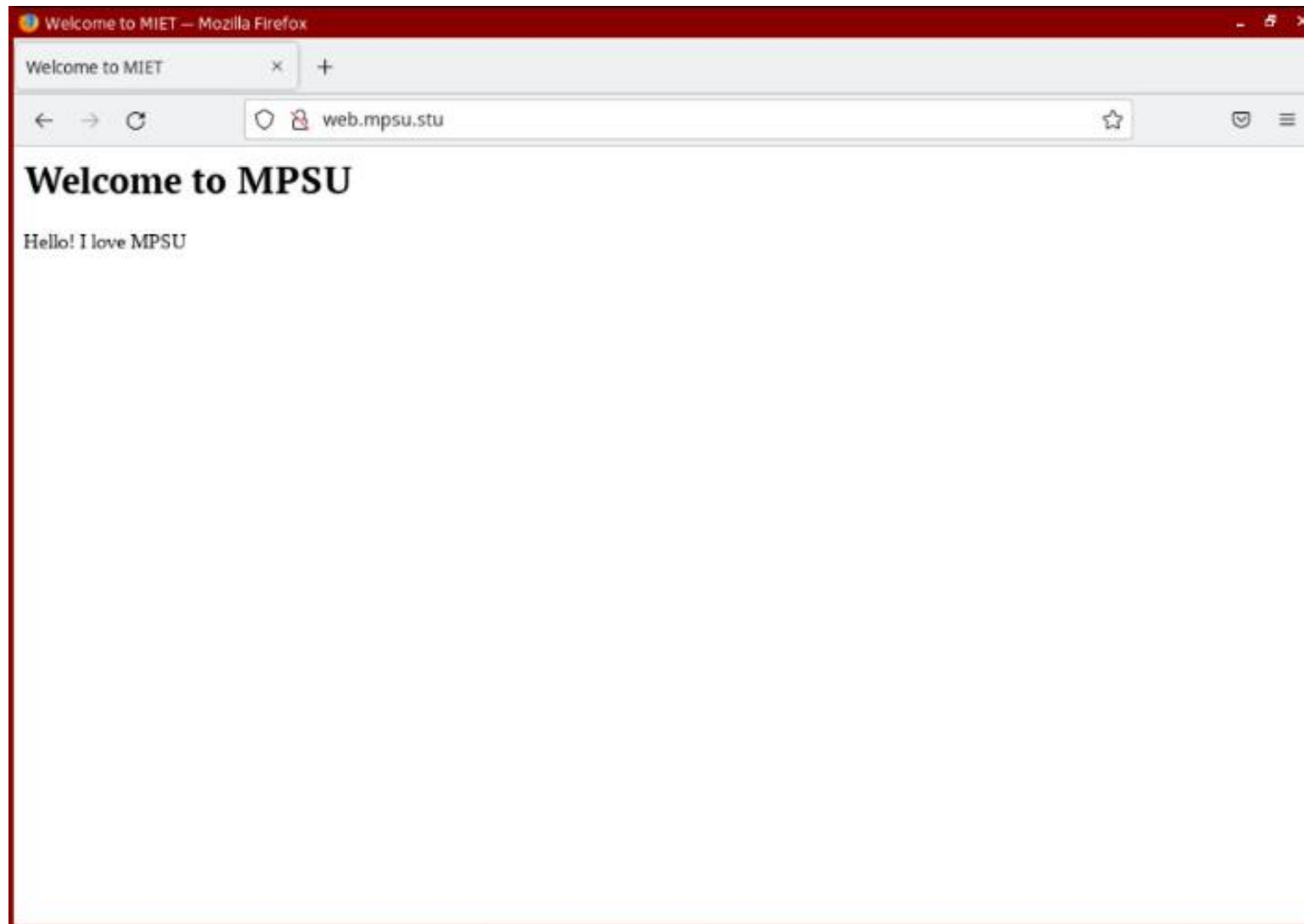
```
<VirtualHost *:80>  
    ServerAdmin admin@mps.u.stu  
    ServerName web.mpsu.stu  
    ServerAlias www.web.mpsu.stu  
    DocumentRoot /var/www/mpsu.stu  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
</VirtualHost>
```

```
sab@server: /$ a2ensite web.mpsu.stu
```

```
sab@server: /$ a2dissite 000-default.conf
```



# Подключение к веб сайту



*Для обращения по доменному имени необходимо его указать в файле DNS*

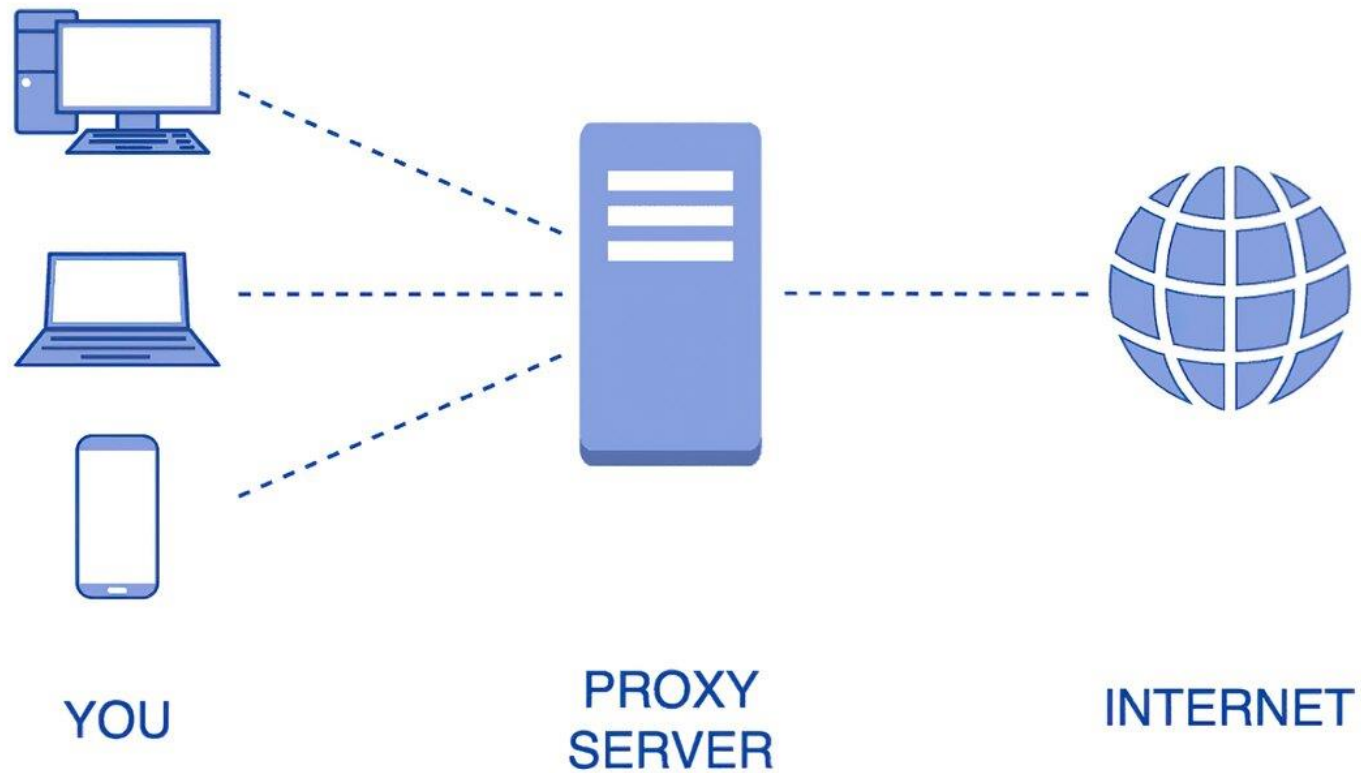


# **Администрирование локальных вычислительных сетей**

**Прокси сервер SQUID**



# Прокси сервер





# Файл конфигурации

## acl имя\_списка параметр содержимое

→	src/dst	->	IP адрес
→	port	->	порт
→	method	->	метод HTTP-запроса {GET, POST, CONNECT...}

## http\_access инструкция acl

→	allow	->	разрешить
→	deny	->	запретить

acl eth port 80	#Добавить в список номер порта
acl localnet src 172.102.0.100	#Добавить в список адрес
acl localnet src 172.1.0.1-172.1.0.255	#Добавить в список диапазон адресов
acl localnet src 172.16.0.0/12	#Добавить в список диапазон адресов
acl localnet srcdomain .mpsu.stu	#Добавить в список адрес домена от которого хотим подключиться
acl guestnet dst 172.16.0.11	#Добавить в список адрес машины к которой пытаемся осуществить доступ
acl guestnet dstdomain .temp.ru	#Добавить в список адрес домена к которому хотим подключиться
http_access allow localnet	#Разрешить доступ к списку с именем localnet
http_access deny all	#Разрешить доступ из остальной сети



# Настройка сервера

```
sab@server: /$ nano /etc/squid/squid.conf
```

```
http_port 3128
```

```
acl block-site dstdomain "/etc/squid/block.txt"  
http_access deny block-site
```

*block.txt*

```
.ya.ru
```

```
sab@server: /$ squid -k reconfigure  
sab@server: /$ systemctl restart squid
```

Параметры соединения

### Настройка прокси для доступа в Интернет

☐ Без прокси

☐ Автоматически определять настройки прокси для этой сети

☐ Использовать системные настройки прокси

☒ Ручная настройка прокси

HTTP прокси  Порт

☒ Также использовать этот прокси для HTTPS

HTTPS прокси  Порт

Узел SOCKS  Порт

☐ SOCKS 4 ☒ SOCKS 5

☐ URL автоматической настройки прокси

Не использовать прокси для



# **Администрирование локальных вычислительных сетей**

**Ansible**





# Устранение ошибки “broken pipe”

Удаляем 8 версию SSH

```
sab@server: /$ sudo apt-get remove openssh-server  
sab@server: /$ sudo apt-get remove openssh-client
```

Устанавливаем 7 версию SSH

```
sab@server: /$ sudo apt-cache policy openssh-client  
sab@server: /$ sudo apt-get install openssh-client=1:7.4p1-10+deb9u6astra6se5  
sab@server: /$ sudo apt-get install openssh-server=1:7.4p1-10+deb9u6astra6se5  
sab@server: /$ sudo apt-get install openssh-sftp-server=1:7.4p1-10+deb9u6astra6se5
```

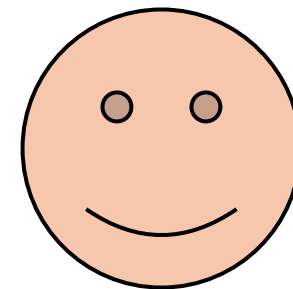
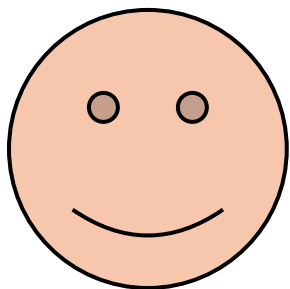


# **Администрирование локальных вычислительных сетей**

**Почтовый сервер**

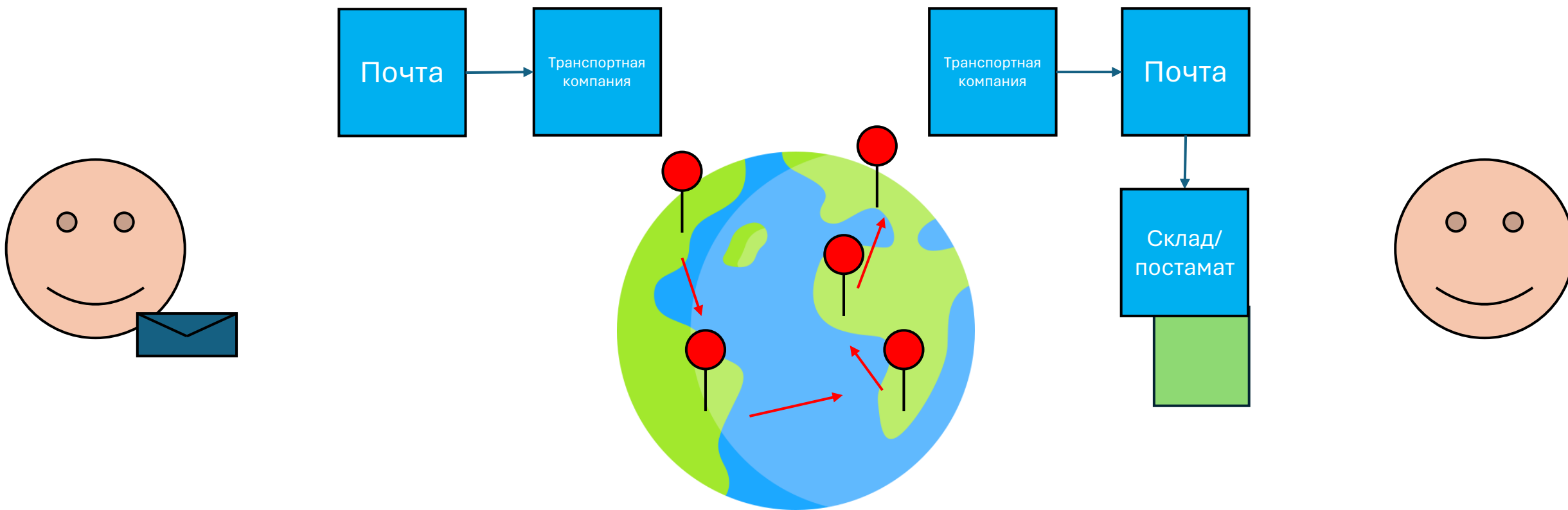


# Передача почты



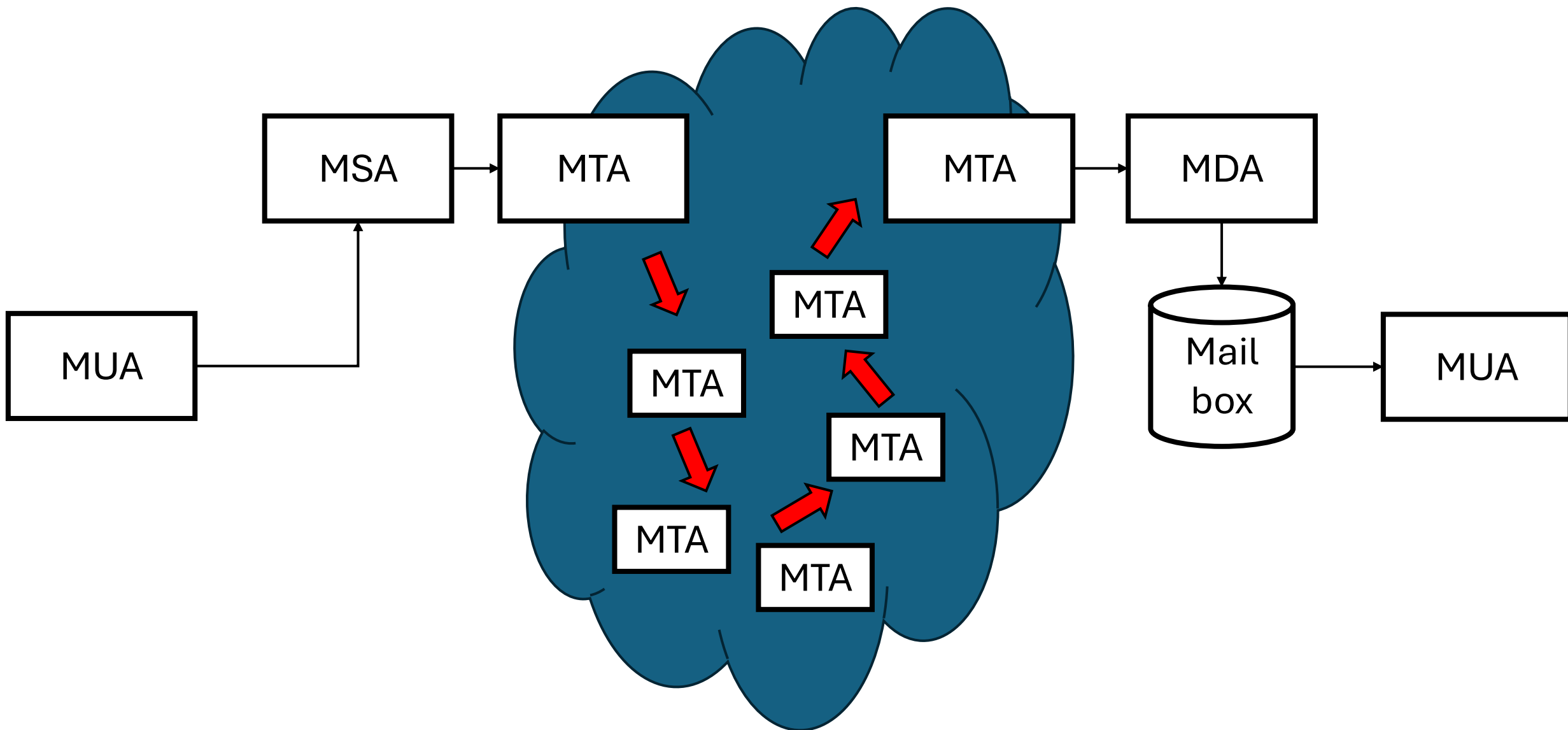


# Почтовое путешествие



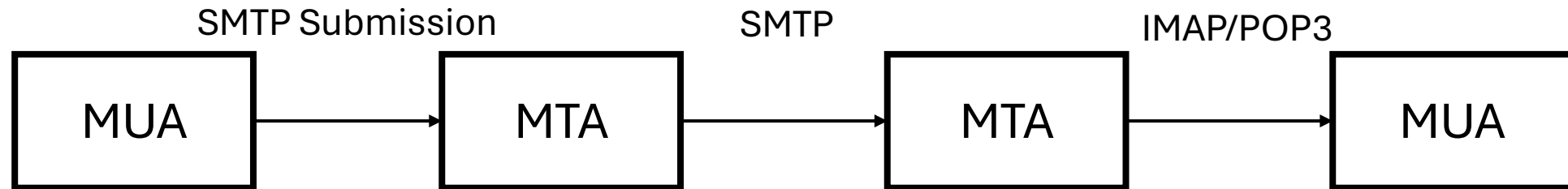


# Передача почты





# Сетевые протоколы



**SMTP** (*Simple Mail Transfer Protocol*) — это протокол, работающий поверх TCP/IP. Предназначен для передачи исходящей почты с использованием порта TCP 25.

*Прикладные протоколы доставки почты:*

**POP3** (*Post Office Protocol Version 3*)

- Порты - 110/TCP, 995/TCP
- Достает и удаляет (по требованию) почту с сервера

**IMAP** (*Internet Message Access Protocol*)

- Порты - 143/TCP, 993/TCP
- Клиент может отслеживать состояние сообщения
- Допускает одновременный доступ нескольких клиентов к почтовому ящику



# Настройка почтовых сервисов

*На сервере*

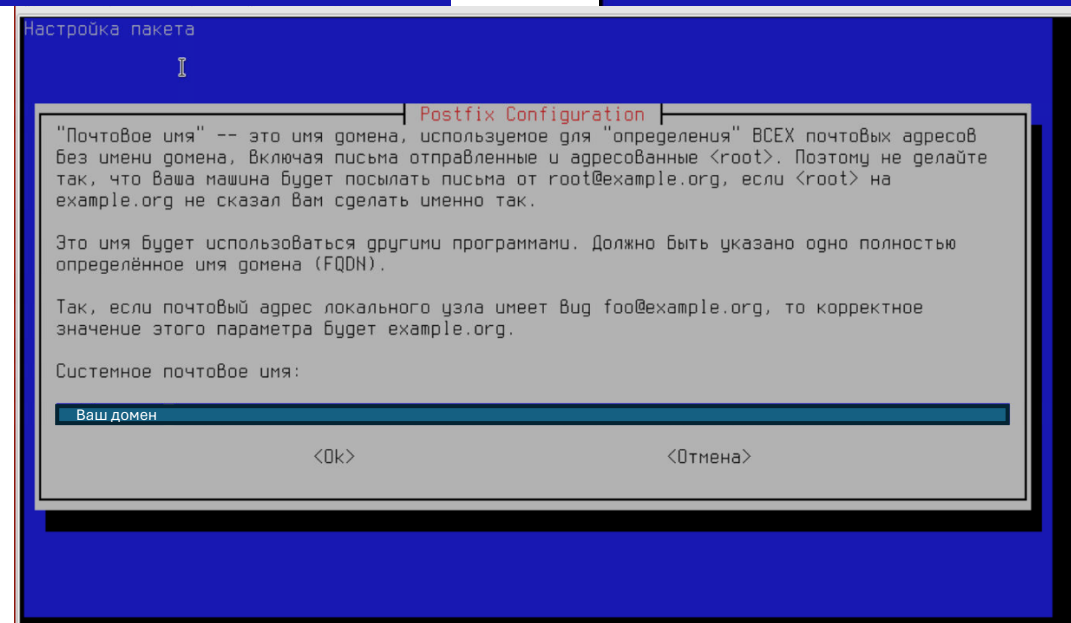
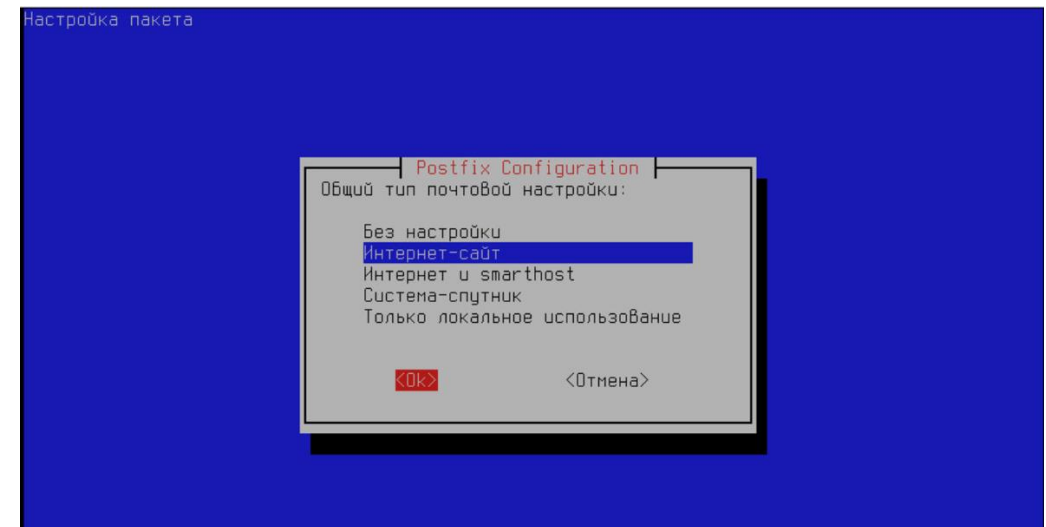
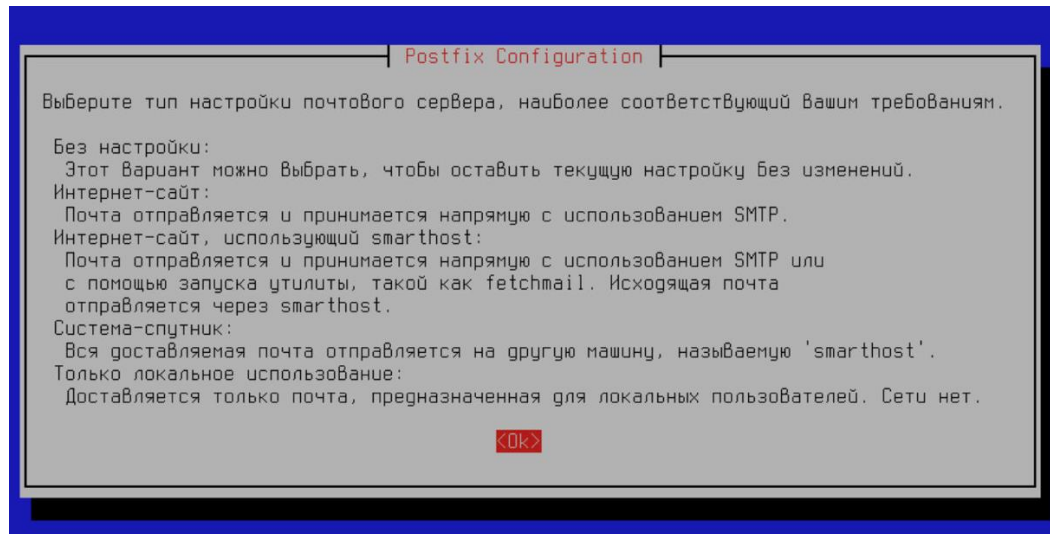
```
sab@server: /$ sudo apt-get install apt-get install postfix dovecot-imapd dovecot-core astrase-fix-maildir
```

*Создание пользователей*

```
adduser mailserver  
usermac mailserver -l 2  
usermac mailserver -c 1  
sudo -u mailserver mkdir -p /home/mailserver/Maildir  
sudo -u mailserver chmod -R 700 /home/mailserver/Maildir  
  
adduser mailclient  
usermac mailclient -l 2  
usermac mailclient -c 1  
sudo -u mailclient mkdir -p /home/mailclient/Maildir  
sudo -u mailclient chmod -R 700 /home/mailclient/Maildir
```



# Настройка postfix







# Настройка конфигураций postfix

*/etc/postfix/main.cf*

```
mynetworks = 192.168.122.0/24  
mydestinations = $myhostname, localhost, ваш домен  
home_mailbox = Maildir/
```

```
sudo postconf  
sudo systemctl enable postfix  
sudo service postfix reload  
sudo systemctl restart postfix  
sudo systemctl status postfix
```



# Настройка конфигураций dovecot

*/etc/dovecot/conf.d/10-auth.conf*

```
disable_plaintext_auth = no  
auth_mechanisms = plain login
```

*/etc/dovecot/conf.d/10-mail.conf*

```
mail_location = maildir:~/Maildir
```

```
sab@server: /$ sudo systemctl enable dovecot  
sab@server: /$ sudo systemctl restart dovecot
```



# Настройка почтового клиента Thunderbird

## *Сервер*

Полное имя: mailserver

Адрес электронной почты: mailserver@mps.u.stu

Сервер входящей почты:

Протокол IMAP

Имя сервера: доменное имя сервера

Порт: 143

Защита соединения: нет

Метод аутентификации: Обычный пароль

Имя пользователя mailserver

Сервер исходящей почты:

Имя сервера: доменное имя сервера

Порт: 25

Защита соединения: нет

Метод аутентификации: Обычный пароль

Имя пользователя mailserver

## *Клиент*

Полное имя: mailclient

Адрес электронной почты: mailclient@mps.u.stu

Сервер входящей почты:

Протокол IMAP

Имя сервера: доменное имя сервера

Порт: 143

Защита соединения: нет

Метод аутентификации: Обычный пароль

Имя пользователя mailserver

Сервер исходящей почты:

Имя сервера: доменное имя сервера

Порт: 25

Защита соединения: нет

Метод аутентификации: Обычный пароль

Имя пользователя mailserver

*Полное имя должно совпадать с созданными выше пользователями*