

Работа с текстовыми редакторами в Linux

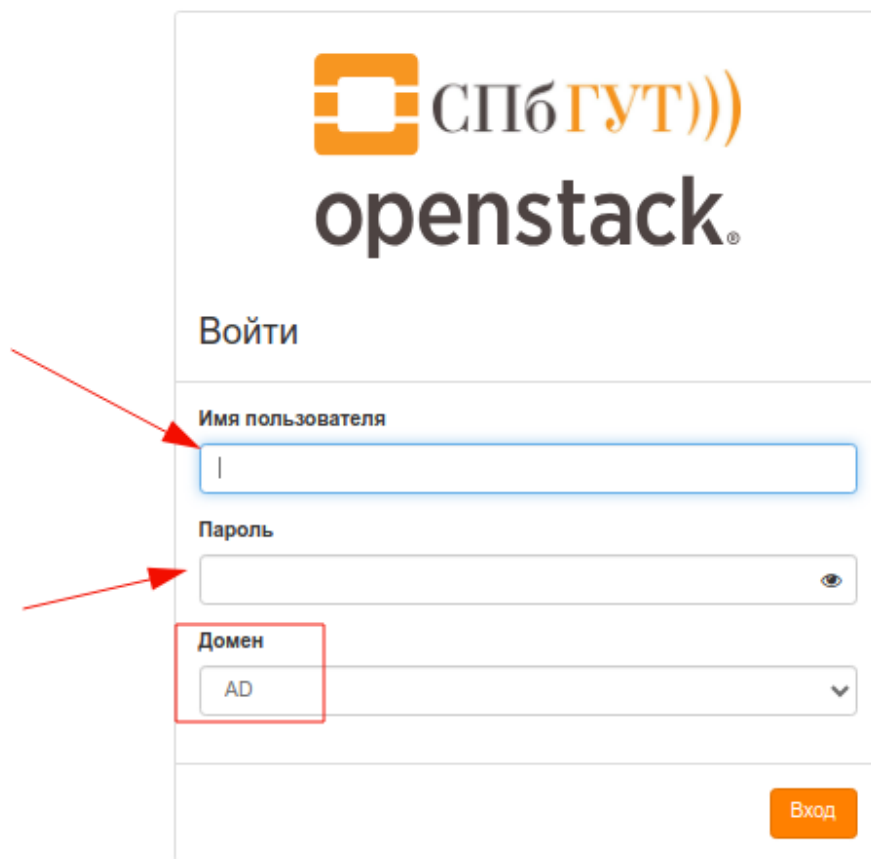
Задачи:

1. Подключиться к облачной инфраструктуре.
2. Убедиться в наличии доступных сетей.
3. Создать виртуальную машину.
4. Настройка правил безопасности.
5. Узнать адрес виртуальной машины.
6. Подключиться к VM по ssh
7. Установить пакет vim
8. Пройти обучение работе с vim.
9. Выполнить базовые действия с файлами и папками.
10. Установить пакет wget.
11. Научиться работать с переменными окружения.
12. Создать нового пользователя и подключиться к нему. Поменять shell у пользователя.
13. Изучить работу с текстом в Bash.
14. Тренировка работы с grep
15. Тренировка работы с sed
16. Тренировка работы с awk

Задание 1. Подключиться к облачной инфраструктуре.

Необходимо перейти по ссылке <https://cloud.resds.ru>. Для подключения использовать домен AD, а также учётную запись пользователя, используемую для подключения к WiFi СПбГУТ

Для выполнения практических занятий необходимо переключиться на проект {username}.dev.

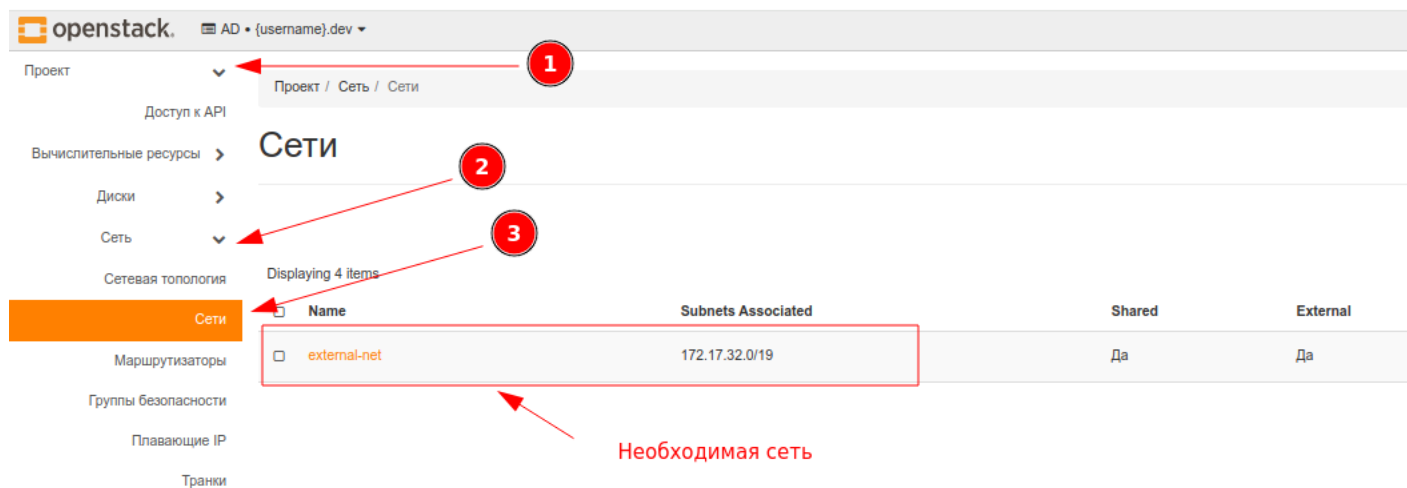


The image shows the OpenStack login page. At the top is the OpenStack logo with the text "СПбГУТ)))" above it. Below the logo is the heading "Войти". There are three input fields: "Имя пользователя" (Username), "Пароль" (Password), and "Домен" (Domain). The "Домен" field is a dropdown menu currently showing "AD". A red box highlights the "Домен" field. Below the input fields is an orange button labeled "Вход" (Login). Two red arrows point to the "Имя пользователя" and "Пароль" fields.

Рис. 1

Задание 2. Убедиться в наличии доступных сетей.

Открыть: проект -> сеть -> сети, и убедиться, что там есть сеть external-net (рис. 2)



The image shows the OpenStack dashboard interface. The top bar includes the OpenStack logo and the text "AD • {username}.dev". The left sidebar contains a navigation menu with items like "Проект", "Доступ к API", "Вычислительные ресурсы", "Диски", "Сеть", "Сетевая топология", "Сети", "Маршрутизаторы", "Группы безопасности", "Плавающие IP", and "Транки". The "Сети" item is highlighted in orange. The main content area shows the "Сети" (Networks) page. At the top, there is a breadcrumb "Проект / Сеть / Сети" with a red circle and arrow labeled "1" pointing to it. Below the breadcrumb, there is a section titled "Сети" with a red circle and arrow labeled "2" pointing to it. Under "Сети", there is a sub-section "Displaying 4 items" with a red circle and arrow labeled "3" pointing to it. Below this, there is a table with the following columns: "Name", "Subnets Associated", "Shared", and "External". The table contains one row with the name "external-net", which is highlighted with a red box and a red arrow labeled "Необходимая сеть" (Required network). The "Subnets Associated" column for "external-net" shows "172.17.32.0/19". The "Shared" and "External" columns both show "Да" (Yes).

Name	Subnets Associated	Shared	External
external-net	172.17.32.0/19	Да	Да

Рис. 2

Задание 3. Сгенерировать ключевую пару.

При первом входе сгенерировать ключевую пару, для доступа к Linux виртуальным машинам. Открыть: Проект -> ключевая пара -> создать ключевую пару (рис. 3)

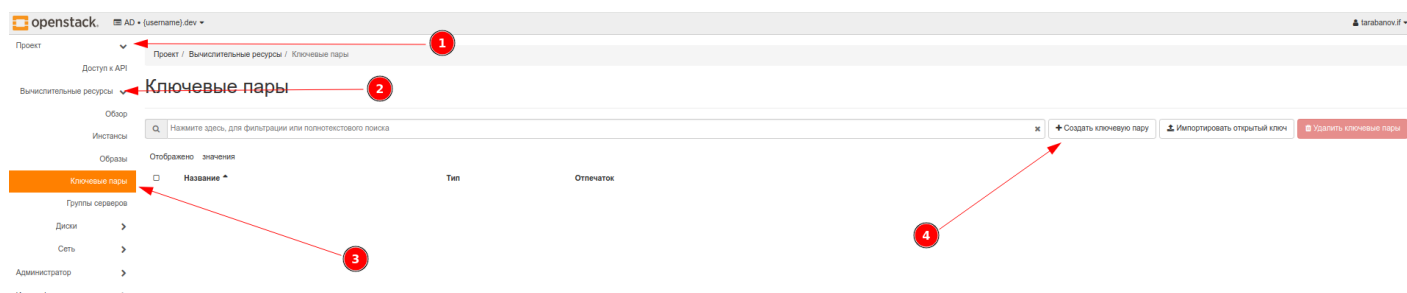


Рис. 3

В открывшемся окне (рис. 4) ввести имя ключевой пары и тип ключа(ssh-key)

Создать ключевую пару

Имя ключевой пары *

my-keypair

Key Type *

SSH Key

Отмена

Создать ключевую пару

Рис. 4

Ключ будет сохранен на ваш компьютер, он понадобится в дальнейшем.

Задание 4. Создать виртуальную машину.

Открыть меню Проект > вычислительные ресурсы > инстансы > запустить инстанс (рис. 5)

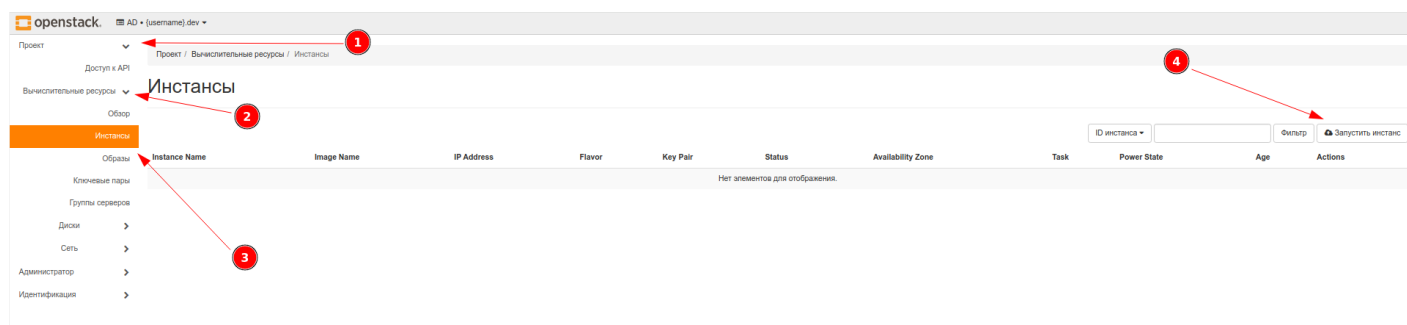


Рис. 5

В открывшемся окне (рис. 6), во вкладке подробности ввести имя инстанса и нажать Следующая > внизу страницы.

Запустить инстанс

Подобности

Имя инстанса *

my-instance

Описание

Зона доступности

Любая зона доступности

Количество *

1

Всего инстансов (100 Max)

21%

20 Использовано на текущий момент

1 Добавлено

79 Свободно

Отмена

< Назад

Следующая >

Запустить инстанс

Рис. 6

В следующем меню (Источник) выбрать источник – образ, указать размер тома данных, выбрать удаление диска при удалении инстанса, выбрать необходимый вам образ из доступных (например Ubuntu-server-20.04:docker), и нажать справа от него стрелку вверх (рис. 7)

Рис. 7

В следующем меню (тип инстанса) определить объем выделяемых виртуальной машине вычислительных ресурсов. Для этого нужно выбрать **один из предопределённых типов инстансов** (например small), и нажать справа от него стрелку вверх (рис. 8).

Рис. 8

В меню сети выбрать нужную вам сеть, к которой будет подключена виртуальная машина (наличие сети было проверено в п.1). Если в инфраструктуре доступна только одна сеть, она будет выбрана автоматически, и выбирать ничего не нужно. (рис. 9)

Рис. 9

Затем перейти к меню Ключевая пара, выбрать созданную ключевую пару, и нажать справа от неё стрелку вверх. (рис. 10)

Рис. 10

После выполнения всех действий - нажать справа снизу кнопку «запустить инстанс» для создания и запуска виртуальной машины.

Задание 5. Настройка правил безопасности.

Для работы с инстансом необходимо разрешить ему сетевое взаимодействие :80/TCP – HTTP, 22/TCP – SSH, 51820/UDP - other. Для этого нужно открыть Проект > Сеть > Группы безопасности > выбрать группу безопасности default и нажать – управление правилами (рис. 11)

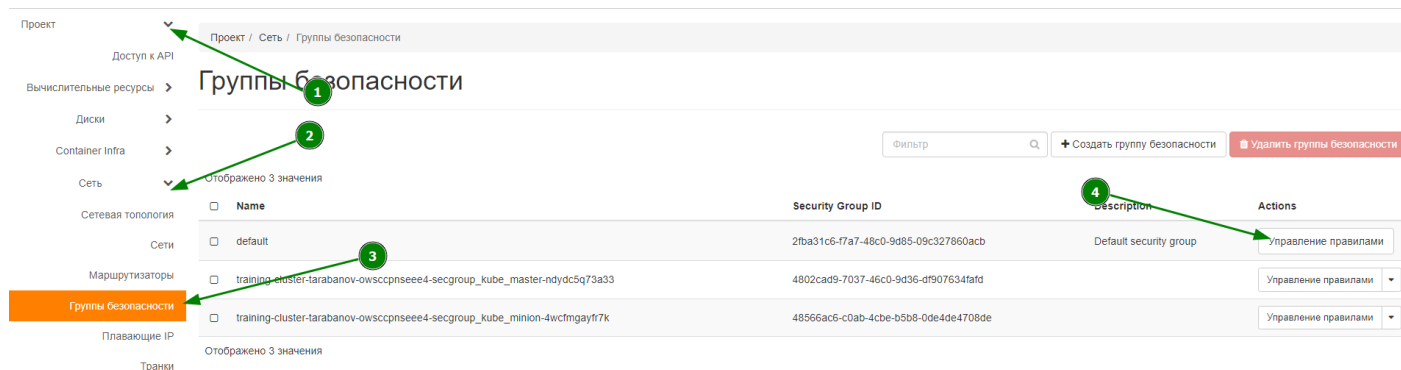


Рис. 11

В открывшемся меню добавить правило для входящего трафика (рис. 12)

Рис. 12

В открывшемся меню добавления правил (рис. 13), добавить правило для порта 80(tcp) Для этого выбрать: Правило: «Настраиваемое правило TCP» Направление: Входящий трафик Порт: 80 Формат записи подключаемого диапазона адресов: CIDR Сам подключаемый диапазон адресов: 0.0.0.0/0 Последняя запись означает разрешение подключения с любого адреса

После заполнения всех полей нажать кнопку «Добавить» в правом нижнем углу.

Рис. 13

Тоже самое необходимо сделать для всех остальных портов.

Задание 6. Узнать адрес виртуальной машины.

Для этого вернуться во вкладку инстансы и в поле ip адрес будет ip адрес вашего виртуального инстанса (рис 14). Этот адрес понадобится в дальнейшем, для подключения к нему и его настройки.

Рис. 14

Задание 7. Подключиться к ВМ по ssh

Linux

Для подключения в большинстве дистрибутивов уже установлены SSH-агенты и для подключения используя ключ достаточно добавить его в агент.

Для этого нужно выполнить команду, где pemkey.pem, это файл полученный вами на 3 пункте данной инструкции

```
ssh-add pemkey.pem
```

Для подключения в данном случае в терминале достаточно ввести команду:

```
ssh cloudadmin@172.17.5.1
```

“ адрес 172.17.5.1 необходимо заменить на ваш адрес полученный из пункта 6 инструкции

Windows OpenSSH

В Windows 10 с версии 1809 включен пакет OpenSSH, проверить это можно с помощью команды (выполняется с правами администратора):

“ Дальнейшие действия выполняются в PowerShell

```
Get-WindowsCapability -Online | ? Name -like 'OpenSSH.Client*'
```

```
PS C:\Users\tarab> Get-WindowsCapability -Online | ? Name -like 'OpenSSH.Client*'

Name : OpenSSH.Client~~~~0.0.1.0
State : Installed
```

Если SSH клиент отсутствует (State: Not Present), его можно установить:

```
Add-WindowsCapability -Online -Name OpenSSH.Client*
```

```
Get-Service ssh-agent | Set-Service -StartupType Automatic -PassThru | Start-Service
```

Далее необходимо включить SSH-агент:

```
Start-Service ssh-agent
```

Добавить ключ можно с помощью команды:

```
ssh-add "C:\Users\{username}\.ssh\id_rsa"
```

“ username - имя вашего пользователя в системе

Теперь вы можете подключиться используя команду:

```
ssh cloudadmin@172.17.5.1
```

“ адрес 172.17.5.1 необходимо заменить на ваш адрес полученный из пункта 6 инструкции

Задание 8. Установка пакета VIM

Для установки, удаления, обновления пакетов в Ubuntu используется утилита `apt`. В процессе установки/обновления/удаления пакетов нужно будет подтвердить установку новых версий, нажав `y`. Или же использовать флаг `-y`. Необходимо установить пакет `vim`, в котором содержится необходимое обучающее руководство.

```
sudo apt install -y vim
```

Задание 9. Пройти обучение работе с vim

Ввести в консоли следующую команду, для запуска интерактивного обучающего курса по работе с VIM:

```
vimtutor ru
```

Пройти интерактивный курс до конца.

Задание 10. Работа с файловой системой.

1. Создать директорию task01. Перейти в неё.

```
mkdir task01  
cd task01
```

2. Создать в домашнем каталоге текстовый файл user при помощи текстового редактора `vi`, и заполнить его произвольными символами.

```
vi user
```

3. Скопировать файл `user` в новый файл с именем `root`.

```
cp user root
```

4. Посмотреть права доступа на файлы можно с помощью команды:

```
ls -l
```

5. Задать владельца `root` и группу `root` на файл `root`. (`sudo` позволяет выполнять команды от `root`; ввести пароль в диалоговом окне, при этом символы отображаться не будут; пароль - `labpass1!`)

```
sudo chown root:root root
```

6. Переименовать (т.е. переместить с новым именем) файл `user` в файл `lock`.

```
mv user lock
```

7. На файл `root` поставить доступ на чтение и запись группе и владельцу остальным только на чтение.

```
sudo chmod 664 root
```

8. На файл `lock` поставить доступ на чтение владельцу, группе и остальным пользователям убрать доступ на чтение запись и исполнение.

```
chmod 600 lock
```

9. Вывести содержимое файла `root` в терминал.

```
cat root
```


10. Отредактировать файл root случайным образом и вывести его содержимое в консоль.

```
sudo vi root  
cat root
```

11. Удалить каталог task01.

```
cd  
sudo rm -rf task01
```

Задание 11. Установка пакетов.

Для установки, удаления, обновления пакетов в Ubuntu используется утилита `apt`. В процессе установки/обновления/удаления пакетов нужно будет подтвердить установку новых версий, нажав `y`. Или же использовать флаг `-y`.

Для установки/удаления пакетов существуют команды:

```
sudo apt install -y package_name  
sudo apt remove -y package_name
```

В качестве примера необходимо установить программу для скачивания файлов по web-протоколам - `wget`.

```
sudo apt install wget
```

Можно проверить версию установленного пакета:

```
wget --version
```

Задание 12. Создание пользователей.

Создать нового пользователя newuser с паролем `newpass1!`. Сделать его администратором.

```
sudo adduser newuser  
sudo passwd newuser # Ввести пароль newpass1!, при этом пароль отображаться не будет  
sudo usermod -aG sudo newuser # Добавить пользователя newuser в группу sudo, что даст ему права на  
исполнение команд с sudo
```

Для того чтобы выполнять команды от имени пользователя `newuser`, необходимо сменить текущее окружение на окружение пользователя newuser командой - `su`. При этом нужно будет ввести пароль пользователя `newuser`.

```
su - newuser
```

Имя пользователя изменилось с `labuser` на `newuser`. Также, имя пользователя хранится в переменной окружения `$USER`. Просмотреть значение переменной можно с помощью команды `echo`.

```
echo $USER
```

Чтобы узнать, в каких группах состоит текущий пользователь, используется команда `groups`. В данном случае это должны быть группы `newuser` и `sudo`.

```
groups
```

Для выхода из оболочки используется команда `exit`. При этом выход идет в оболочку, запущенную пользователем `labuser`.

```
exit
```

Теперь нужно сменить оболочку, запускаемую по умолчанию при входе пользователя `newuser`. Чтобы узнать список доступных в системе оболочек, используется следующая команда:

```
cat /etc/shells
```

Сейчас должна быть запущена оболочка `/bin/bash`. Чтобы узнать, какая оболочка запущена сейчас, можно вывести на экран значение переменной `$0`.

```
echo $0
```

Чтобы просто запустить оболочку, достаточно просто набрать путь к ней. Запустить оболочку - `sh`, и вывести на экран переменную - `$0`. После выйти обратно в `bash`.

```
/bin/sh  
echo $0  
exit
```

Чтобы сменить оболочку по умолчанию для пользователя, можно отредактировать файл `/etc/passwd`, либо можно использовать утилиту `usermod`. Необходимо посмотреть, какая оболочка сейчас является оболочкой по умолчанию для пользователя `newuser`, сменить её на `/bin/sh`, и проверить изменения.

```
sudo grep newuser /etc/passwd # grep - программа для поиска по тексту. В данном случае, выведет все строки, содержащие newuser.
```

```
sudo usermod --shell /bin/sh newuser  
sudo grep newuser /etc/passwd # Данная команда произведет поиск по файлу  
# /etc/passwd, и отобразит в консоли все  
# строки, в которых присутствует слово  
# newuser. В этой же строке будет указано,  
# какая оболочка используется данным  
# пользователем
```

Теперь необходимо переключиться в режим работы от имени пользователя `newuser`, и проверить, какая оболочка используется. После выйти из этого режима, и удалить пользователя.

```
su - newuser  
echo $0  
exit  
sudo userdel -r newuser # флаг r используется, когда вы хотите удалить также домашний каталог  
пользователя.
```

Задание 13. Работа с текстом в bash.

Для практики нужно воспользоваться текстовым файлом, специально созданным для выполнения задания - таблица подключений OpenVPN, состоящая из имени клиента, IP адреса, MAC адреса устройства и внешнего IP, с которого клиент подключился. Необходимо загрузить файл, воспользовавшись специальной утилитой curl:

```
sudo apt install curl -y  
cd ~  
curl -L -o clients.txt https://pastebin.com/raw/cEYa64Cz
```

Вывести содержимое файла в консоль (Тут для удобства можно пользоваться встроенными в консоль горячими клавишами: Ctrl+L - Очистить содержимое консоли, Shift+PgUp - Прокрутить консоль вверх, Shift+PgDn - Прокрутить консоль вниз).

```
cat ~/clients.txt
```

Для того чтобы вывести строки, содержащие подстроку, можно использовать grep. Необходимо вывести информацию о клиенте под номером 24 (grep - регистрозависимый, Client24 начинается с заглавной буквы).

```
grep Client24 ~/clients.txt
```

В результате исполнения команды должна быть выведена одна строка. Если было несколько строк, содержащих подстроку Client24, то в результате выведется несколько строк. Далее

необходимо ввести команду:

```
grep Client2 ~/clients.txt
```

Необходимо попробовать понять, сколько будет выведено строк и почему именно они.

Помимо обычных строк `grep` поддерживает также регулярные выражения. При этом регулярные выражения должны экранироваться символом `\`. С помощью регулярных выражений можно задать абсолютно любой паттерн. Относительно простой - вывести всех клиентов, имя которых заканчивается на 4. Точка в выражении означает любой символ.

```
grep Client\4 ~/clients.txt
```

`grep` также умеет работать с пайплайном (вертикальная черта). С помощью пайплайна можно передавать вывод от одной программы другой, по принципу конвейера. Например, можно вывести текст через `cat`, и передать этот вывод на вход команды `grep`, для его обработки этой командой.

```
cat ~/clients.txt | grep Client\4
```

Пайплайнов может быть несколько. Так, например, воспользовавшись программой `AWK` можно вывести только имя клиента и его внешний адрес - то есть первый и четвертый столбцы. (Внимание на пробел в двойных кавычках между номерами столбцов. Это разделитель, который будет между столбцами в конечном результате. Пробел можно заменить на любой другой символ, например, на дефис `-`, или символ табуляции `\t`. Можно попробовать это сделать)

“Awk - это полноценный язык обработки текстовой информации с синтаксисом, напоминающим синтаксис языка C. Он обладает довольно широким набором возможностей, однако, мы рассмотрим лишь некоторые из них -- наиболее употребимые в сценариях командной оболочки. Awk "разбивает" каждую строку на отдельные поля. По-умолчанию, поля -- это последовательности символов, отделенные друг от друга пробелами, однако имеется возможность назначения других символов, в качестве разделителя полей. Awk анализирует и обрабатывает каждое поле в отдельности. Это делает его идеальным инструментом для работы со структурированными текстовыми файлами, особенно с таблицами.

```
cat ~/clients.txt | grep Client\4 | awk '{print $1" "$4}'
```

`AWK` также поддерживает функции в своем синтаксисе. Например, чтобы вывести имя клиента и MAC, при этом чтобы MAC печатался заглавными буквами, нужно использовать следующую конструкцию.

```
cat ~/clients.txt | grep Client\4 | awk '{print $1" "toupper($3)}'
```

Далее необходимо вывести MAC адрес, чтобы он был разделён не двоеточиями, а дефисами, тогда можно воспользоваться `sed`. Нужно лишь задать параметры для замены. После этого результат можно вывести не в консоль, а в новый файл (набирать в одну строку):

“ Sed - это неинтерактивный строчный редактор. Он принимает текст либо с устройства `stdin`, либо из текстового файла, выполняет некоторые операции над строками и затем выводит результат на устройство `stdout` или в файл. Как правило, в сценариях, `sed` используется в конвейерной обработке данных, совместно с другими командами и утилитами.

```
cat ~/clients.txt | grep Client\4 | awk '{print $1" "toupper($3)}' | sed -r 's:/-/g' > newfile.txt
```

Задание 14. Тренировка работы с `grep`

Устанавливаем пакет, `git` для работы с `git` репозиториями:

```
sudo apt install -y git
```

Копируем репозиторий с GitHub и устанавливаем тренажер:

```
git clone https://github.com/learnbyexample/TUI-apps.git
cd TUI-apps/GrepExercises
sudo apt install -y python3 python3-pip python3-venv
python3 -m venv textual_apps
cd textual_apps
source bin/activate
pip install grepexercises
```

Запускаем тренажер

```
grepexercises
```

Задание 15. Тренировка работы с `sed`

Переходим :

```
cd ../sedexercises
python3 -m venv textual_apps
```

```
cd textual_apps
source bin/activate
pip install sedexercises
```

Запускаем тренажер

```
sedexercises
```

Задание 16. Тренировка работы с awk

Переходим :

```
cd ../awkexercises
python3 -m venv textual_apps
cd textual_apps
source bin/activate
pip install awkexercises
```

Запускаем тренажер

```
awkexercises
```

Версия #15

Тарабанов Илья Федорович создал 2 ноября 2023 11:03:57

Тарабанов Илья Федорович обновил 16 мая 2024 19:12:05