Редактирование файлов

Работа с WinSCP. Обзор команд для просмотра текста и текстовых редакторов (nano, vim). Работа с текстовыми файлами в командной строке. Конкатенация файлов (cat). Объединение команд.

**Оглавление**

[Работа с WinSCP](#_ydnulzldcona)

[Просмотр текстовых файлов](#_oyfgidt210wy)

[Команда less](#_vznvk3k67174)

[Текстовые редакторы](#_ct4knme9z8jb)

[nano](#_oum5maihhs8)

[Vim](#_gets86djjn4x)

[Конкатенация файлов (cat)](#_xd6x3kylxylm)

[Объединение команд](#_2u54j3rgnf1g)

[Практическое задание](#_111kx3o)

[Дополнительные материалы](#_3l18frh)

[Используемая литература](#_206ipza)

# 

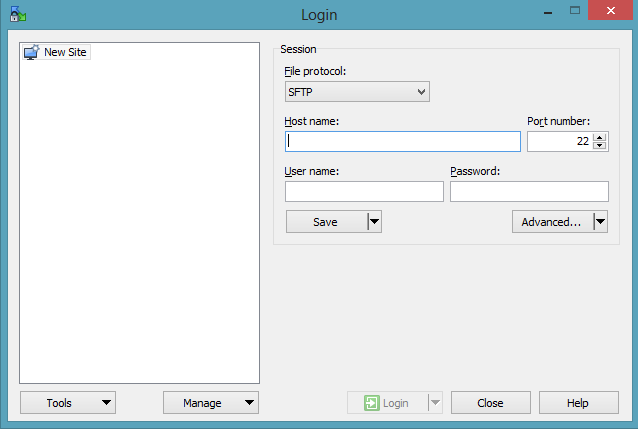
# 

# Работа с WinSCP

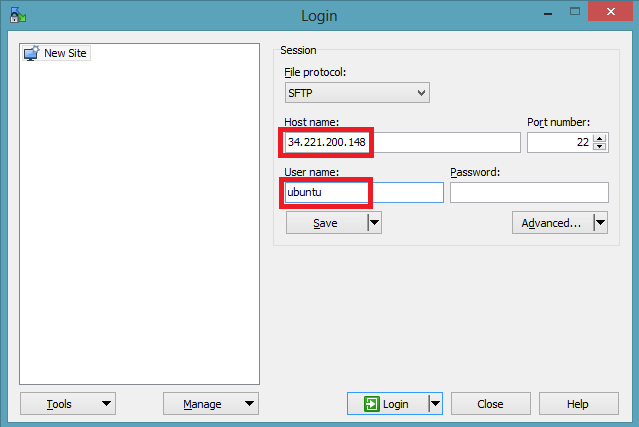
В начале этого урока мы скопируем текстовые файлы с локального компьютера на удаленный сервер, который мы запустили в облаке AWS. Если для отправки команд на удаленный сервер мы использовали протокол **ssh** (Secure Shell — «безопасная оболочка»), то для передачи файлов в облако и обратно применим протокол **SFTP** (SSH File Transfer Protocol).

Для передачи файлов с Windows на Linux будем использовать программу **WinSCP**. Мы скачали ее в одном из предыдущих уроков. Для соединения с сервером потребуется логин, IP-адрес и приватный ключ (файл с расширением **.ppk**).

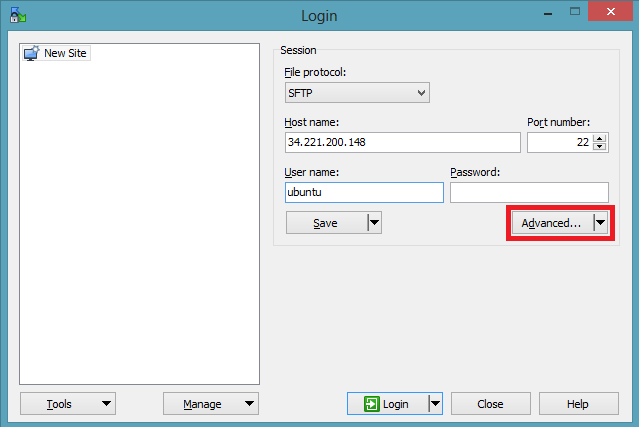
Запустим **WinSCP**, и появится окно ввода логина:



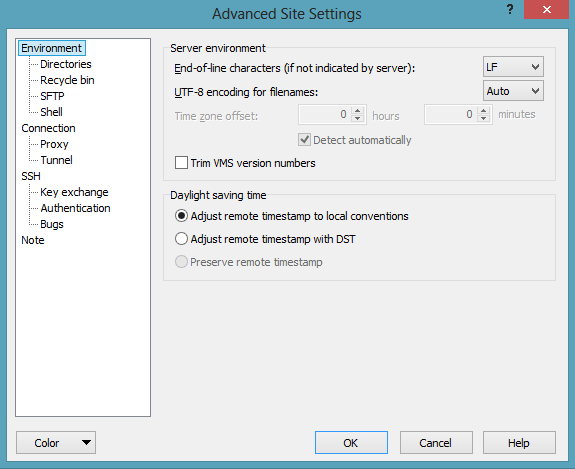
Введем **hostname** (можно ввести IP удаленного сервера) и логин:



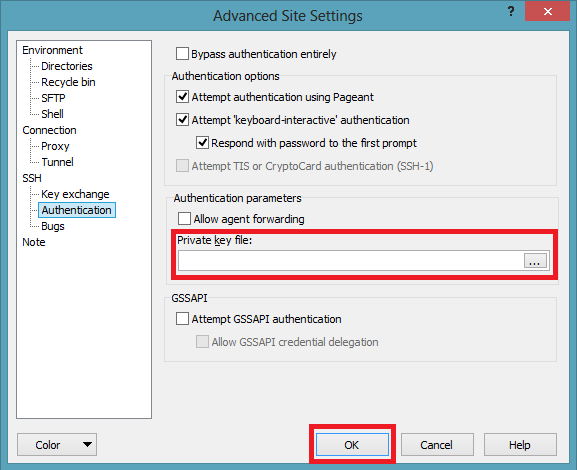
Затем нажмем на кнопку **Advanced**:



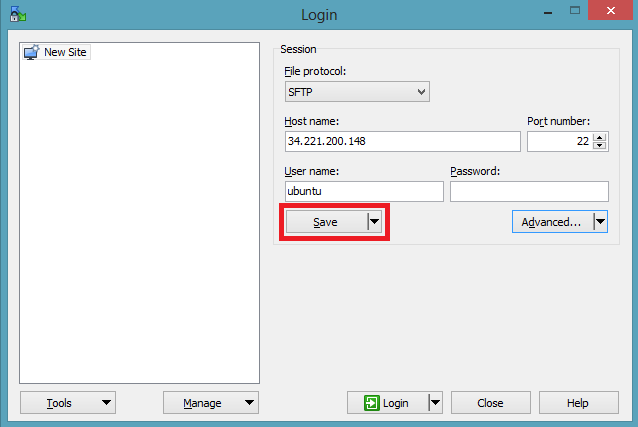
Появится окно с продвинутыми настройками:



Зайдем в раздел **Authentication**, выберем файл приватного ключа и нажмем **OK**:

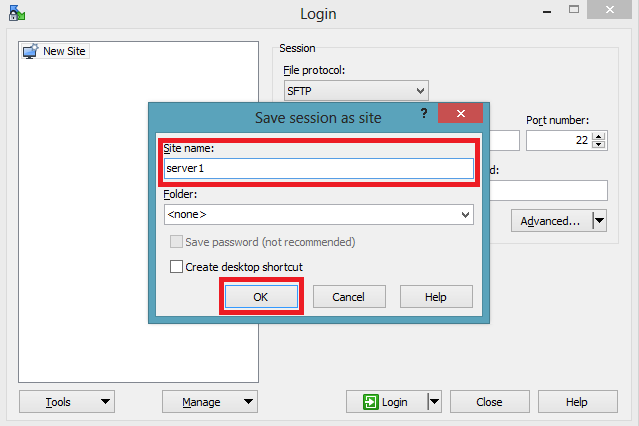


Сохраним настройки, нажав кнопку **Save**:

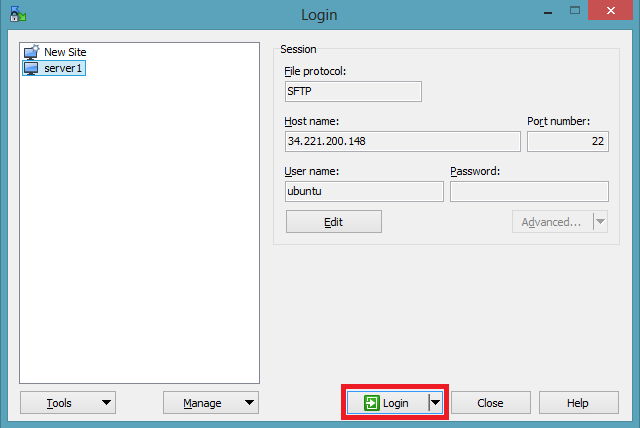


Появится всплывающее окно. В поле **Site name** введем имя соединения.

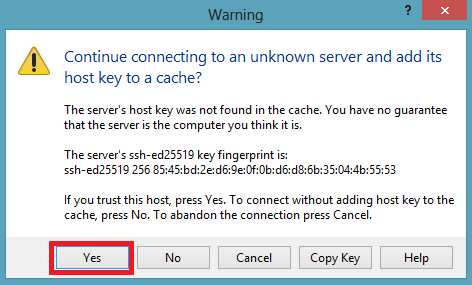
Для него можно выбрать любое название:



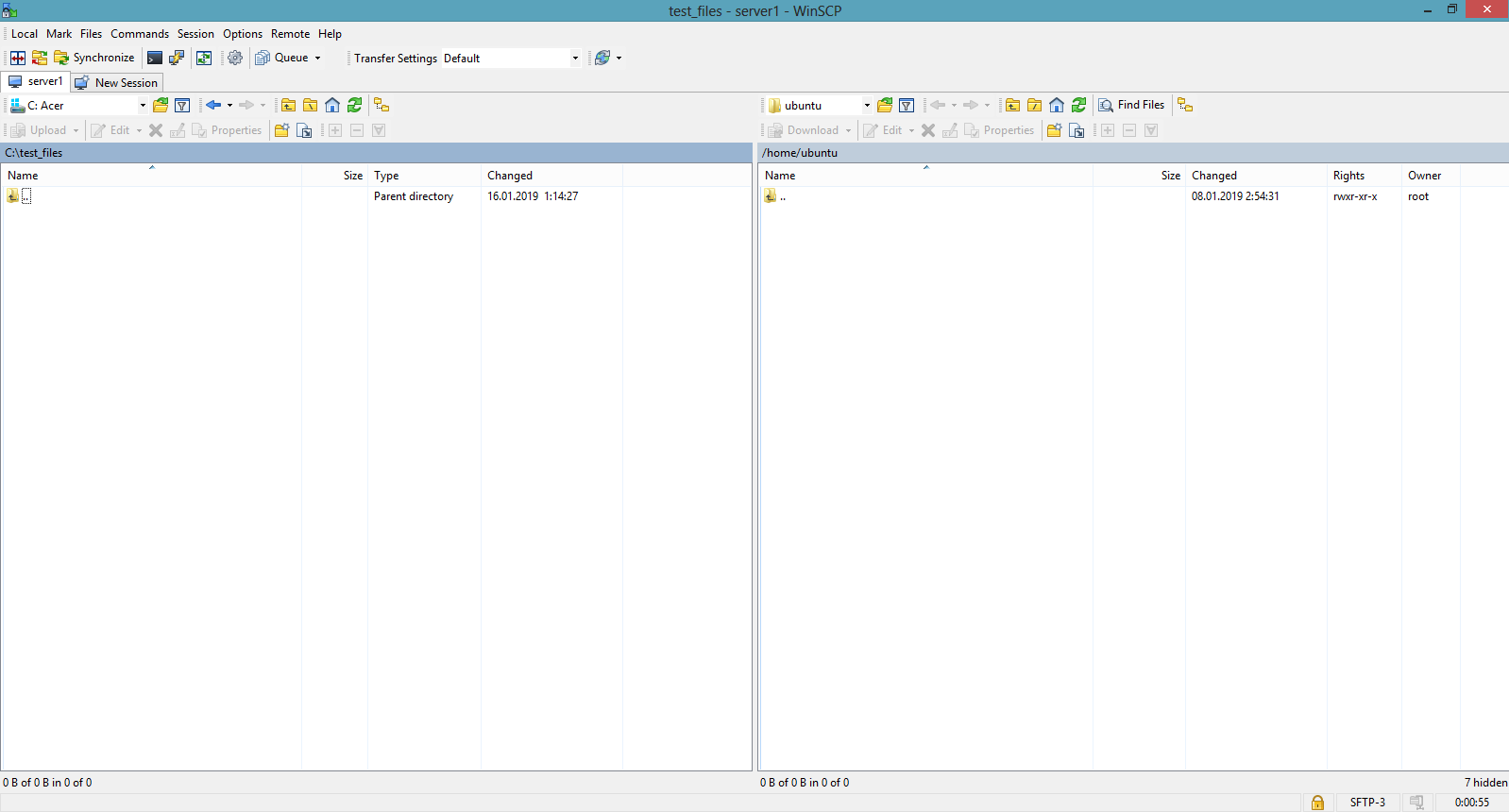
Теперь можно залогиниться:



Далее нажимаем **Yes**:



Появится окно, в котором можно управлять файлами как локального компьютера (окно справа), так и удаленного сервера (окно слева), в том числе передавать файлы:

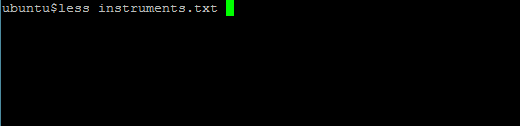


Передадим файл **instruments.txt** с локального компьютера на удаленный сервер в домашнюю директорию пользователя **ubuntu**. Этот файл мы сначала добавим на локальный компьютер, а затем перенесем мышкой из левой половины окна в правую. Таким образом файл начнет копироваться на удаленный сервер.

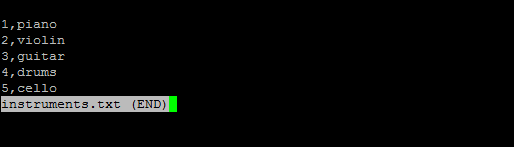
# Просмотр текстовых файлов

## Команда less

Откроем **instruments.txt** командой **less**. С ее помощью удобно знакомиться с большими текстовыми файлами, так как она не грузит сразу весь текст, а позволяет просматривать его небольшими порциями. Запустим команду **less ruments.txt**:

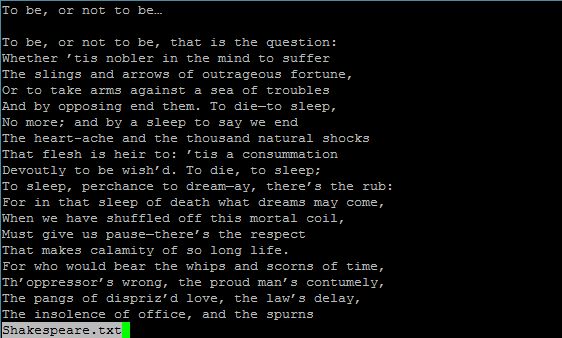


На экран будет выведено содержимое файла **instruments.txt**:



У файла есть два поля, разделенных запятой. В первом — id музыкального инструмента, во втором — его название. Для выхода из файла нужно нажать клавишу **q**.

Бывает нужно просмотреть файл, все содержимое которого не помещается на один экран. Чтобы увидеть следующий экран с текстом, нужно нажать клавишу пробела, а чтобы вернуться к тексту на экран назад — клавишу **b**. Попробуем освоиться с этими командами, и для этого откроем файл, в котором содержится больше строк — **Shakespeare.txt**:



**less** унаследовала часть команд от **more**, но является более новой и имеет больше возможностей.

# Текстовые редакторы

## nano

Команда **nano** дает возможность вносить изменения в текстовый файл. Открыть файл **instruments.txt** можно с помощью команды **nano instruments.txt**, а редактировать — в окне терминала:



Переместим курсор с помощью клавиши «Вниз» и впишем две новые строки в файл:



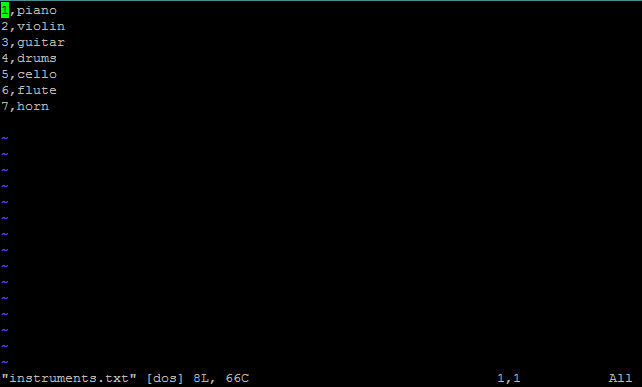
В правом верхнем углу появится надпись ***Modified***. Чтобы сохранить изменения, нужно нажать **Ctrl+O**, а после этого — либо нажать **Enter**, тогда файл сохранится под этим же именем, либо сначала отредактировать название файла, чтобы создать новый с обновленным содержимым). Для закрытия файла нужно нажать **Ctrl+X**.

## Vim

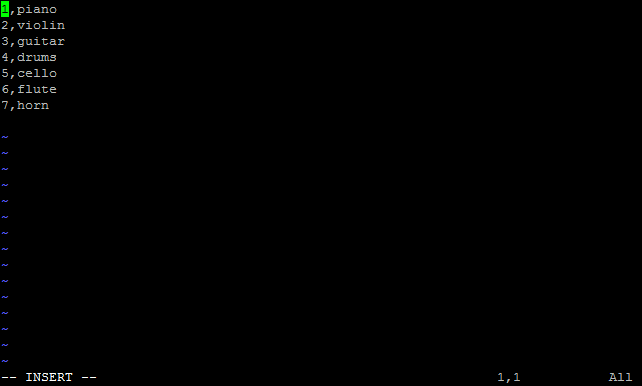
Текстовый редактор **Vim** произошел от более старого редактора **Vi**, и их команды слегка разнятся. В отличие от других текстовых редакторов, **Vim** не прост: чтобы управлять им, требуется подготовка. У него два основных режима: командный (нормальный) и текстовый (режим вставки).

Сначала запустим редактор **Vim** — откроем в нем файл **instruments.txt**, который мы до этого редактировали. Для этого запустим команду **vim instruments.txt**:

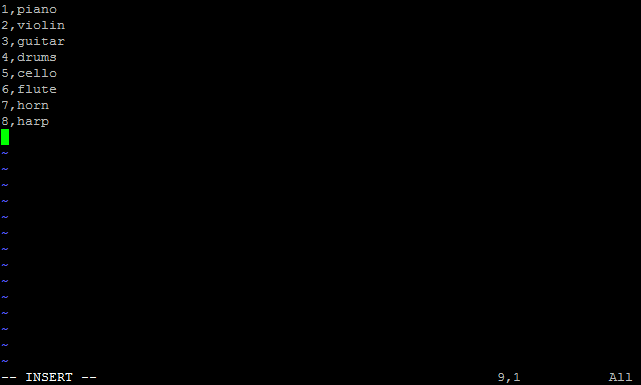


После того как мы открыли файл, редактор работает в командном режиме: 

Чтобы перейти в текстовый режим, более привычный для работы, нужно нажать клавишу **i** (от **insert** — вставить). Внизу экрана появится надпись **INSERT**:



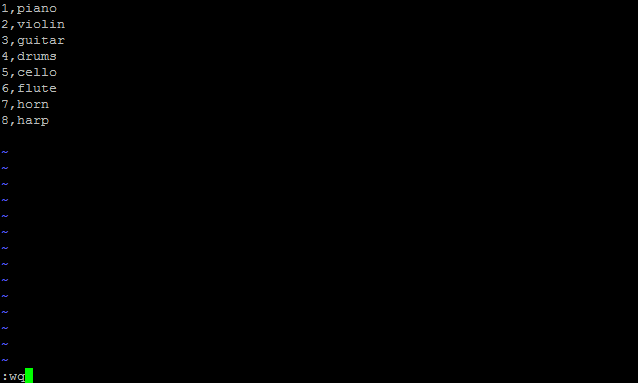
Переместим курсор вниз и добавим строку с текстом ***8,harp*** в файл **instruments.txt**, находясь в текстовом режиме:



Чтобы вернуться в командный режим, нажмем **Esc** (иногда требуется нажать два раза). После этого для сохранения файла нужно ввести команду **:w**, а для выхода без сохранения — **:q**. Если файл был изменен, то в конце команды нужно добавить **!**, чтобы редактор позволил закрыть файл без сохранения. Чтобы выйти из файла с сохранением, нужно ввести команду **:wq**.

После ввода двоеточия курсор переместится вниз экрана, и можно будет продолжать писать команду.

Например, сейчас нам нужно выйти из редактора с сохранением файла. Для этого нажимаем **Esc**, чтобы перейти в командный режим, и вводим двоеточие. Затем — символы **wq**, завершая ввод клавишей **Enter** (на картинке показано состояние экрана перед нажатием **Enter**):

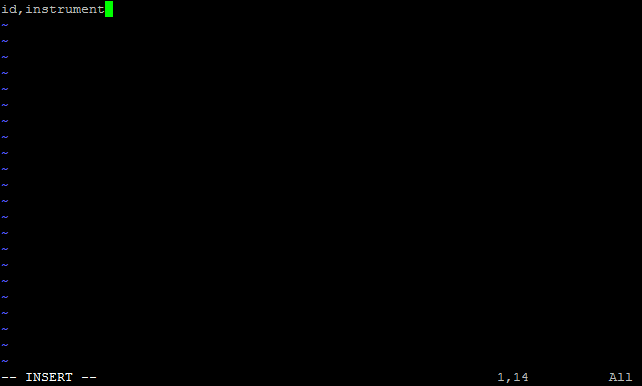


После нажатия **Enter** файл сохранится и редактор закроется.

Редактор **Vim** удобен для программирования, так у него множество команд для быстрого редактирования текста, их можно объединять и автоматизировать. Есть подсветка синтаксиса и поддержка около двухсот языков программирования. С помощью **Vim** можно сравнивать два файла и переносить отдельные изменения из одного в другой. Этот редактор можно гибко настроить под задачи конкретного пользователя.

Создать файл с помощью **Vim** просто: если ввести после команды **vim** название файла, который еще не создан в текущей папке, то создастся пустой файл с таким именем и он откроется в редакторе **Vim**.

С помощью команды **vim header.txt** создадим новый файл (предварительно с помощью команды **ls** можно убедиться, что такого файла нет в текущей папке). Перейдем в текстовый режим и в первой строке файла введем текст **id,instrument**:



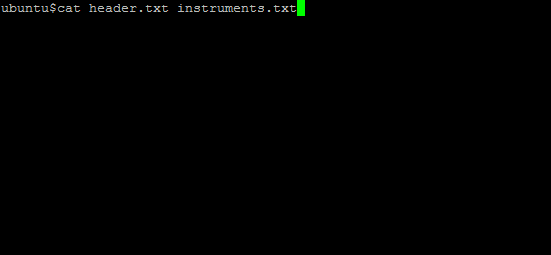
После этого перейдем в нормальный режим и закроем файл с сохранением.

Помимо командного и текстового режимов в **Vim** есть визуальный, позволяющий обрабатывать сразу несколько строк. Такой режим более нагляден, чем командный.

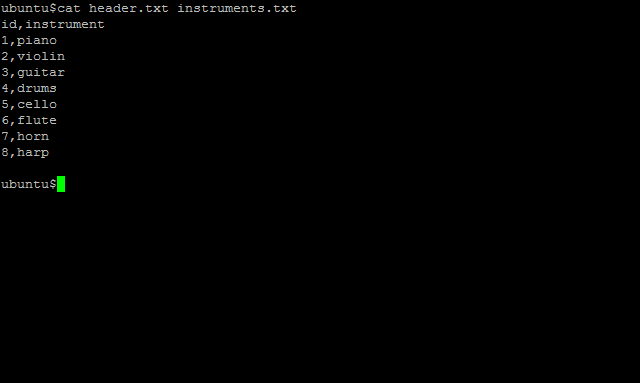
# Конкатенация файлов (cat)

Команда позволяет объединять файлы с текстовым содержимым.

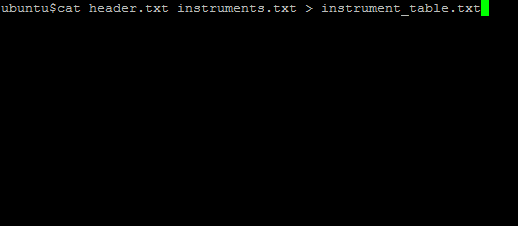
Запустим команду **cat header.txt instruments.txt**:



Объединенное текстовое содержимое этих двух файлов будет выведено на экран:

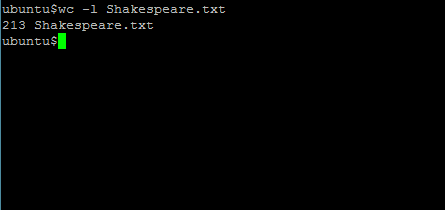


Немного усложним команду, чтобы объединенный текст сохранился в новый файл под названием **instrument\_table.txt**. Для этого запустим команду **cat header.txt instruments.txt > instrument\_table.txt**:

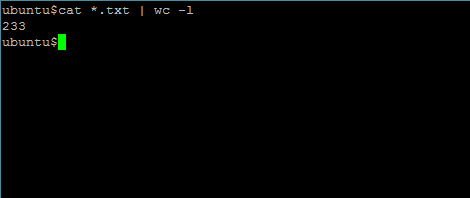


# Объединение команд

С помощью символа **|** можно передавать результат одной команды на вход следующей. Например, посредством команды **wc -l Shakespeare.txt** мы можем узнать количество строк одного файла:



А используя команду **cat \*.txt | wc -l**, можно узнать суммарное количество строк всех файлов из текущей папки, имеющих расширение **.txt**:



Таким образом, с помощью символа **|** можно составлять длинные цепочки команд. Их называют **pipe** — конвейер.

# Практическое задание

1. С помощью текстового редактора **Vim** создать файл с программой на Python, выводящей текст ***Hello, world!***
2. Запустить команду, определяющую число строк в файле.
3. Создать еще один файл с командой на Python, выводящей текст ***Linear regression***.
4. Объединить эти два файла с помощью команды **cat**.
5. Придумать три случая применения команды **cat** для работы с текстовыми файлами.

# Дополнительные материалы

1. [vim — Linux Command — Unix Command](https://www.lifewire.com/vim-linux-command-4095196).
2. [Текстовый редактор nano в Linux для новичков](https://losst.ru/tekstovyj-redaktor-nano-v-linux-dlya-novichkov).
3. [Nano: И все-таки его придется выучить](https://habr.com/ru/post/106471/).
4. [Команда Cat в Linux. Основные и расширенные примеры](https://andreyex.ru/operacionnaya-sistema-linux/komanda-cat-v-linux-osnovnye-i-rasshirennye-primery/).

# Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. [Команда cat Linux](https://losst.ru/komanda-cat-linux).
2. [13 Basic Cat Command Examples in Linux](https://www.tecmint.com/13-basic-cat-command-examples-in-linux/).
3. [Редактор nano](http://www.systemfree.ru/okl/rn/).