

Linux и облачные вычисления

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](#)

[Просмотр текстовых файлов](#)

[Команда less](#)

[Текстовые редакторы](#)

[nano](#)

[Vim](#)

[Конкатенация файлов \(cat\)](#)

[Объединение команд](#)

[Практическое задание](#)

[Дополнительные материалы](#)

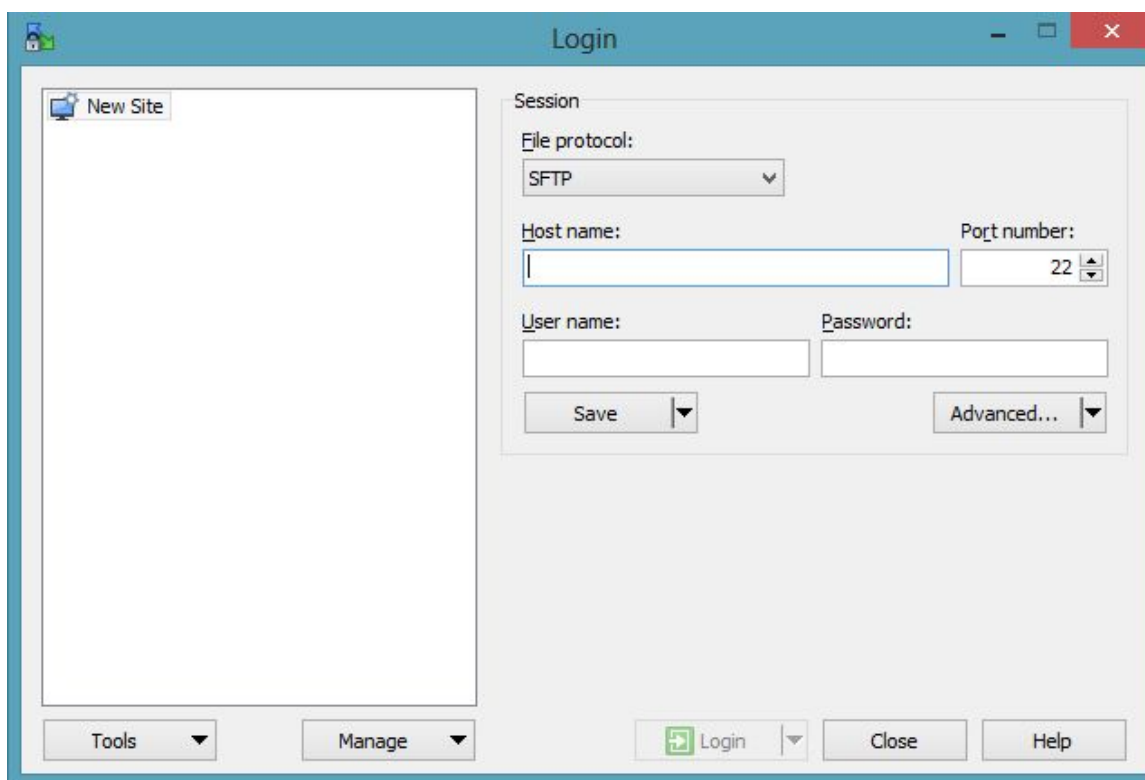
[Используемая литература](#)

Работа с WinSCP

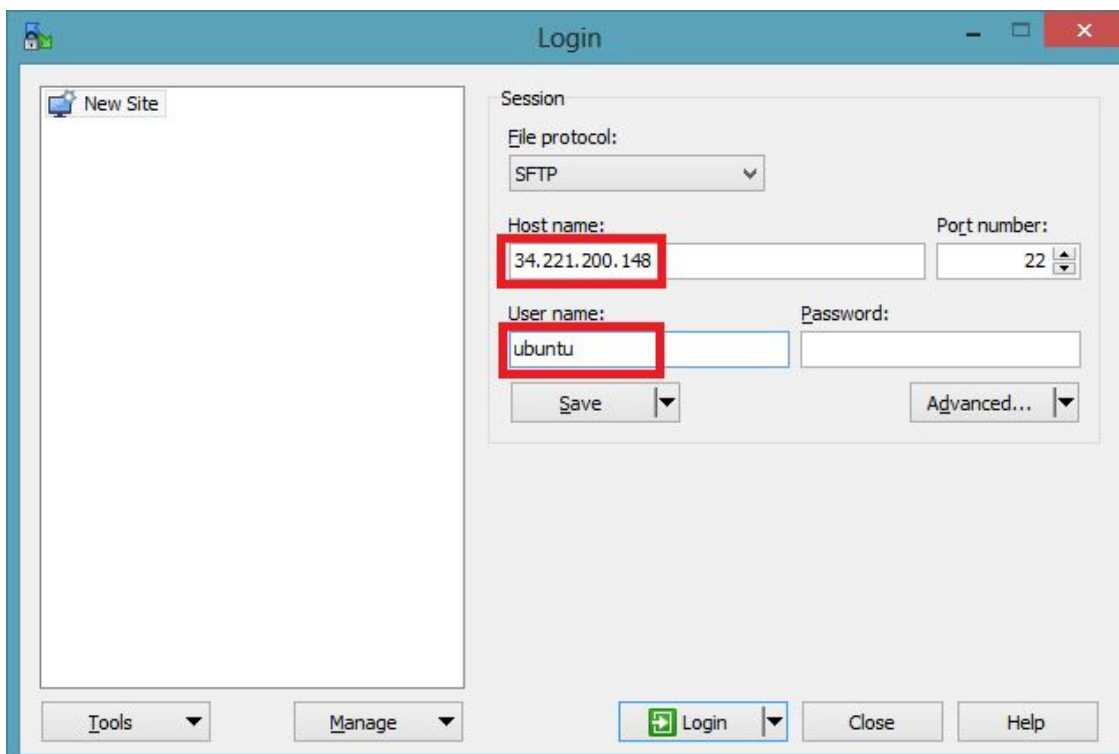
В начале этого урока мы скопируем текстовые файлы с локального компьютера на удаленный сервер, который мы запустили в облаке AWS. Если для отправки команд на удаленный сервер мы использовали протокол **ssh** (Secure Shell — «безопасная оболочка»), то для передачи файлов в облако и обратно применим протокол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

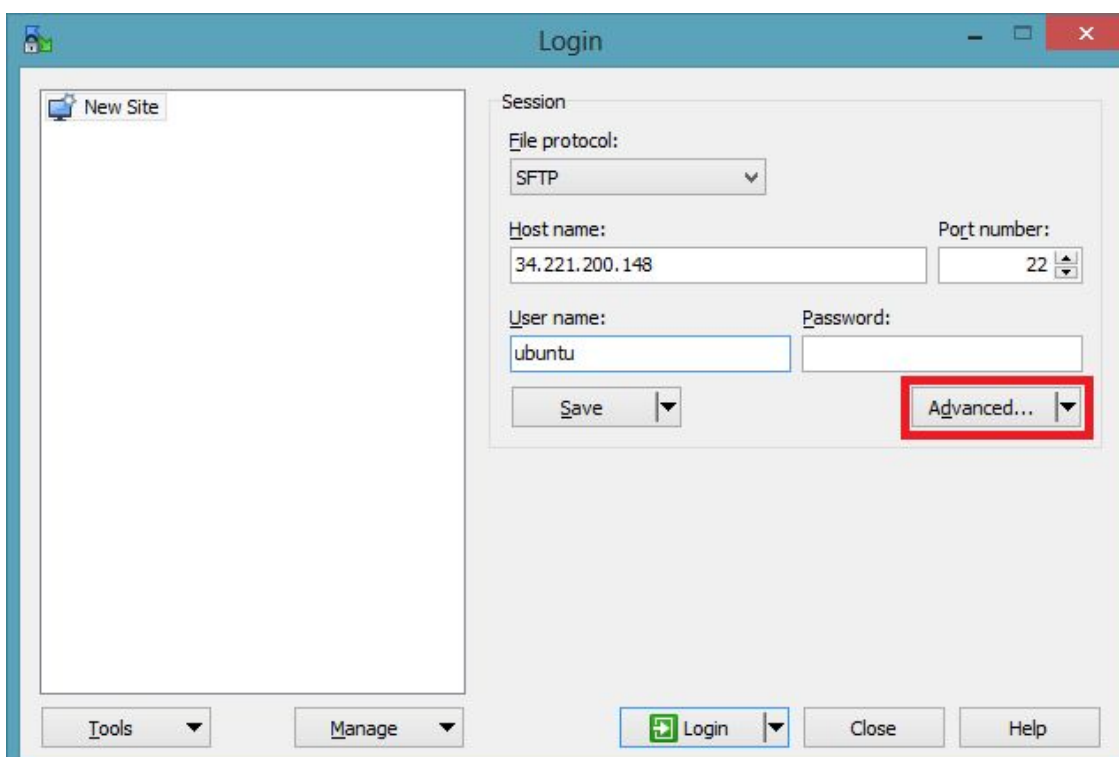


Введем **hostname** (можно ввести IP удаленного сервера) и логин:



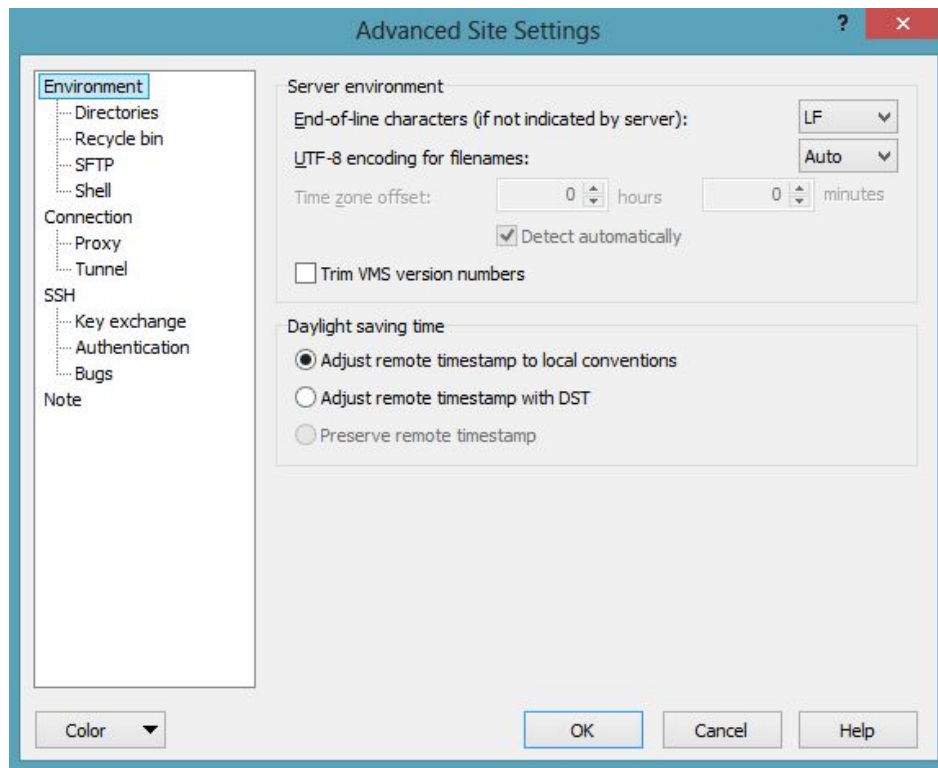
The screenshot shows a 'Login' window with a 'Session' section. The 'File protocol' is set to 'SFTP'. The 'Host name' field contains '34.221.200.148' and is highlighted with a red box. The 'Port number' is set to '22'. The 'User name' field contains 'ubuntu' and is also highlighted with a red box. The 'Password' field is empty. There are 'Save' and 'Advanced...' buttons below the fields. At the bottom of the window, there are 'Tools', 'Manage', 'Login', 'Close', and 'Help' buttons.

Затем нажмем на кнопку **Advanced**:

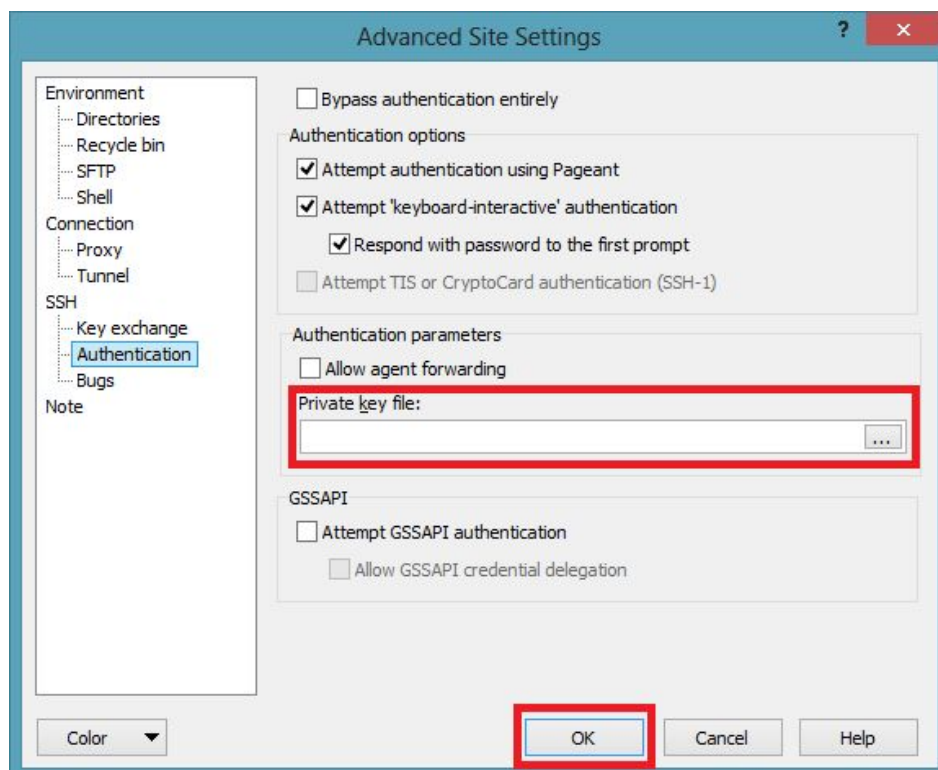


This screenshot is identical to the previous one, but the 'Advanced...' button in the 'Session' section is now highlighted with a red box, indicating the next step in the process.

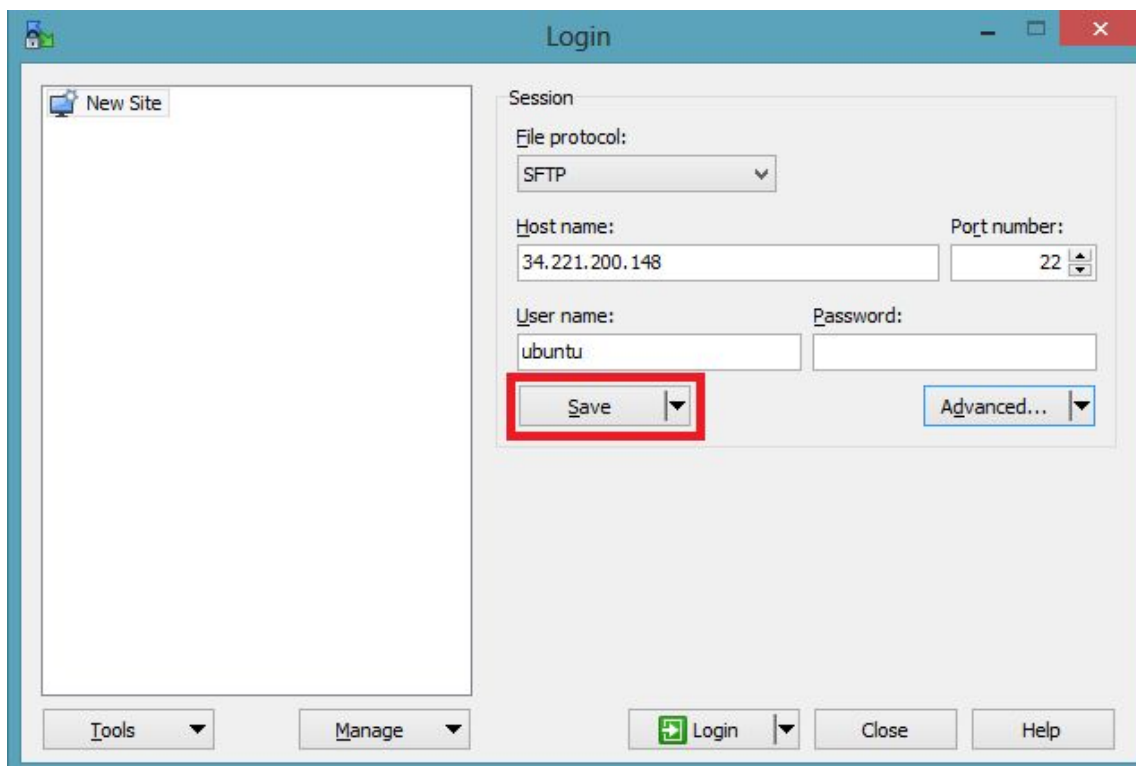
Появится окно с продвинутыми настройками:



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

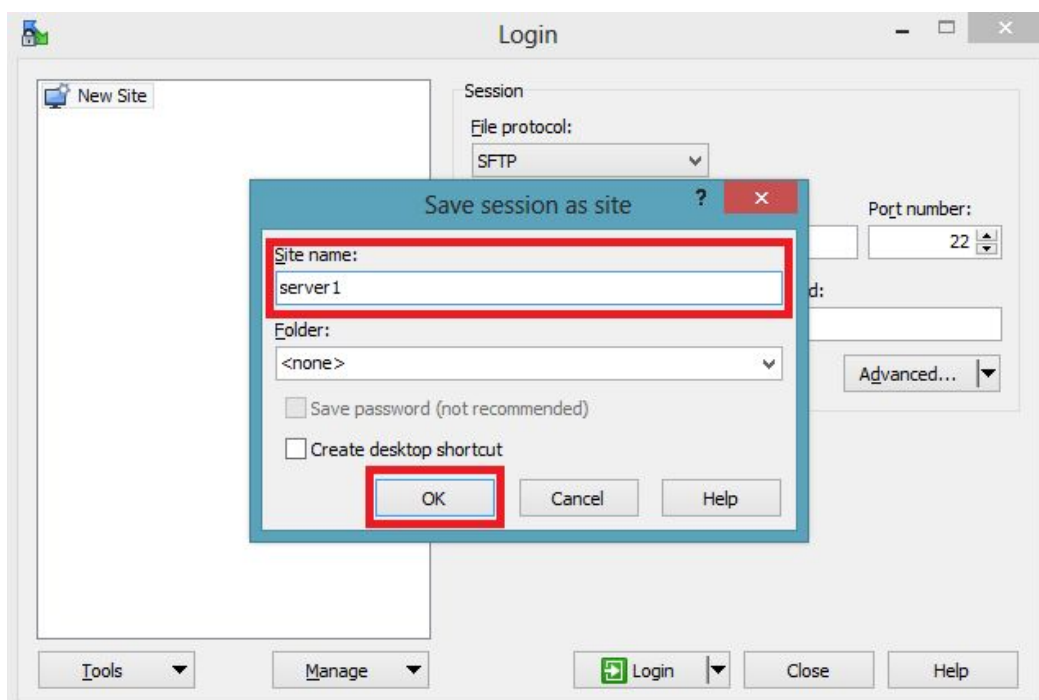


Сохраним настройки, нажав кнопку **Save**:

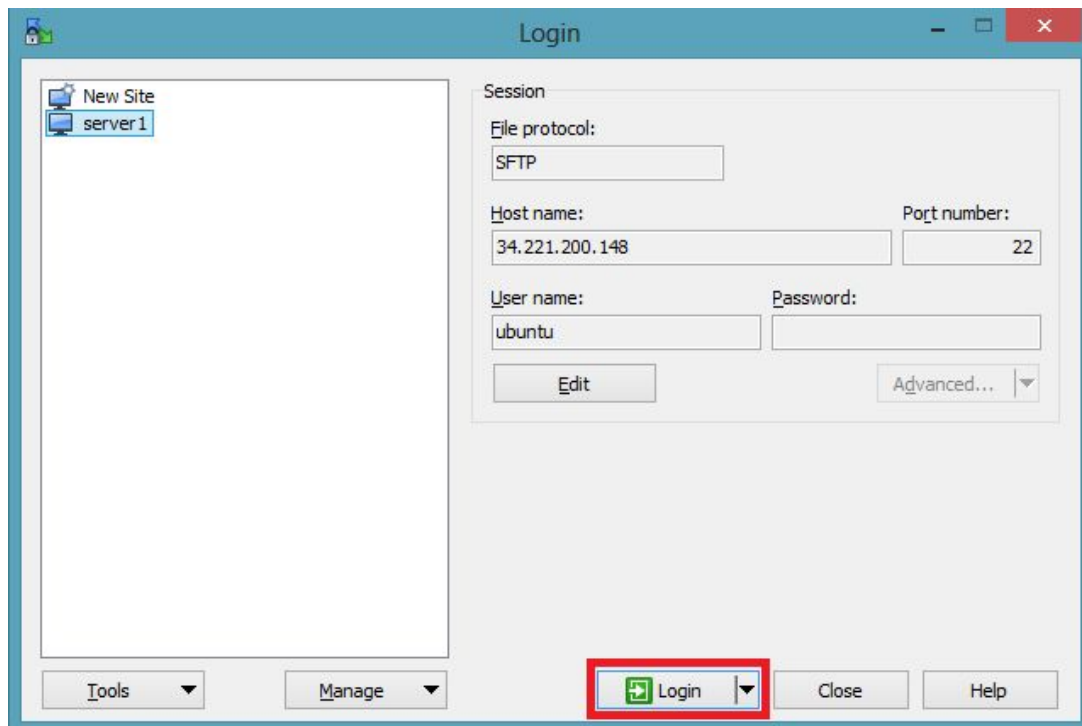


Появится всплывающее окно. В поле **Site name** введем имя соединения.

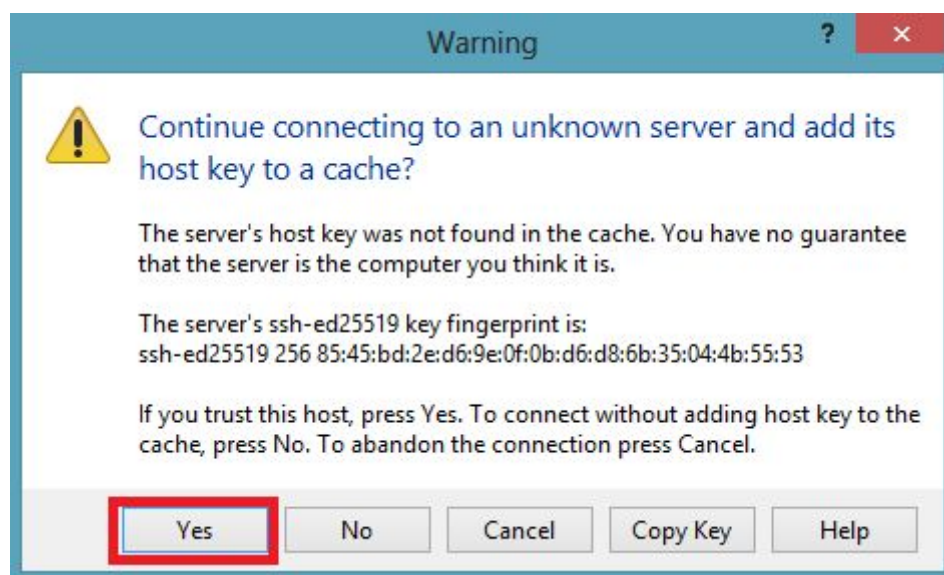
Для него можно выбрать любое название:



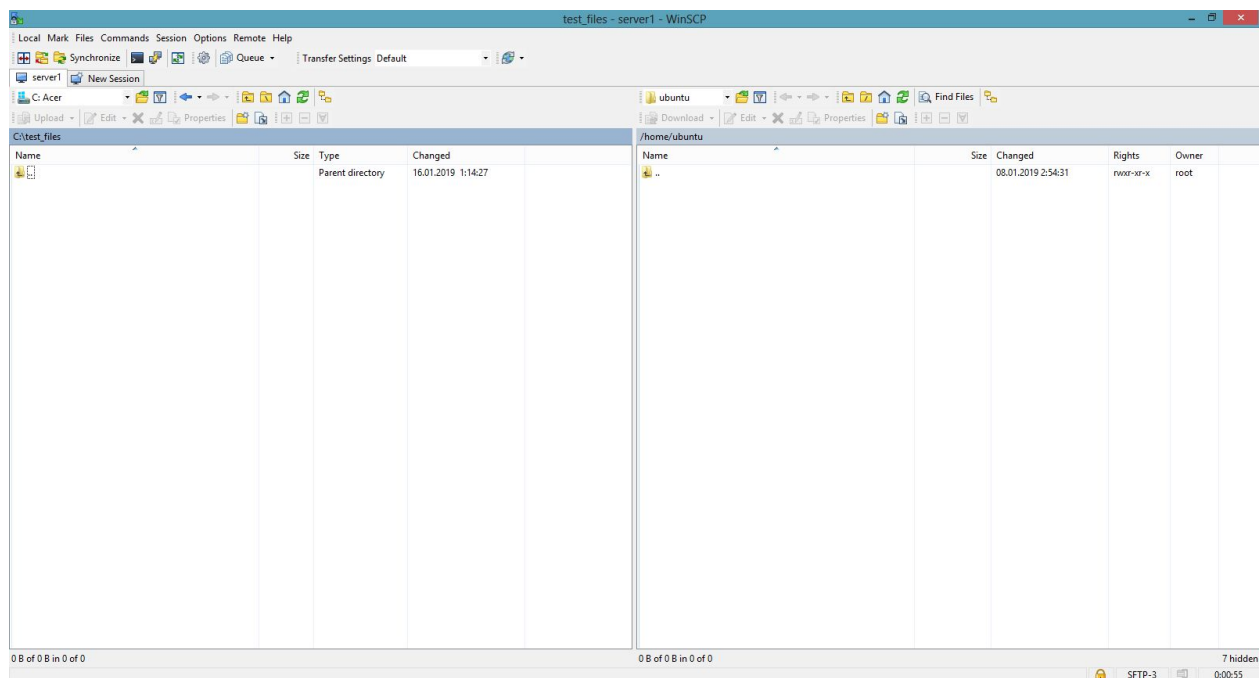
Теперь можно залогиниться:



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:

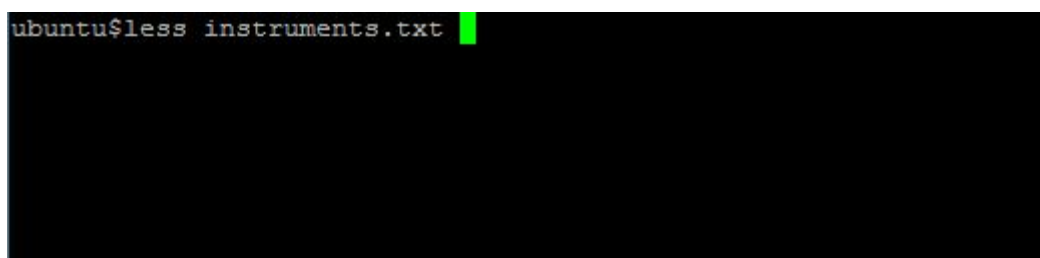


Передадим файл **instruments.txt** с локального компьютера на удаленный сервер в домашнюю директорию пользователя **ubuntu**. Этот файл мы сначала добавим на локальный компьютер, а затем перенесем мышкой из левой половины окна в правую. Таким образом файл начнет копироваться на удаленный сервер.

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**:

```
1,piano
2,violin
3,guitar
4,drums
5,cello
instruments.txt (END)
```

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**:

```
To be, or not to be...

To be, or not to be, that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take arms against a sea of troubles
And by opposing end them. To die—to sleep,
No more; and by a sleep to say we end
The heart-ache and the thousand natural shocks
That flesh is heir to: 'tis a consummation
Devoutly to be wish'd. To die, to sleep;
To sleep, perchance to dream—ay, there's the rub:
For in that sleep of death what dreams may come,
When we have shuffled off this mortal coil,
Must give us pause—there's the respect
That makes calamity of so long life.
For who would bear the whips and scorns of time,
Th'oppressor's wrong, the proud man's contumely,
The pangs of dispriz'd love, the law's delay,
The insolence of office, and the spurns
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:


```
GNU nano 2.9.3 instruments.txt

1,piano
2,violin
3,guitar
4,drums
5,cello

[ Read 5 lines (Converted from DOS format) ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

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

```
GNU nano 2.9.3 instruments.txt Modified

1,piano
2,violin
3,guitar
4,drums
5,cello
6,flute
7,horn
^

[ Read 5 lines (Converted from DOS format) ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

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**:

```
ubuntu$vim instruments.txt
```

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

```
1,piano
2,violin
3,guitar
4,drums
5,cello
6,flute
7,horn

"instruments.txt" [dos] 8L, 66C 1,1 All
```

```
1,piano  
2,violin  
3,guitar  
4,drums  
5,cello  
6,flute  
7,horn
```

-- INSERT --

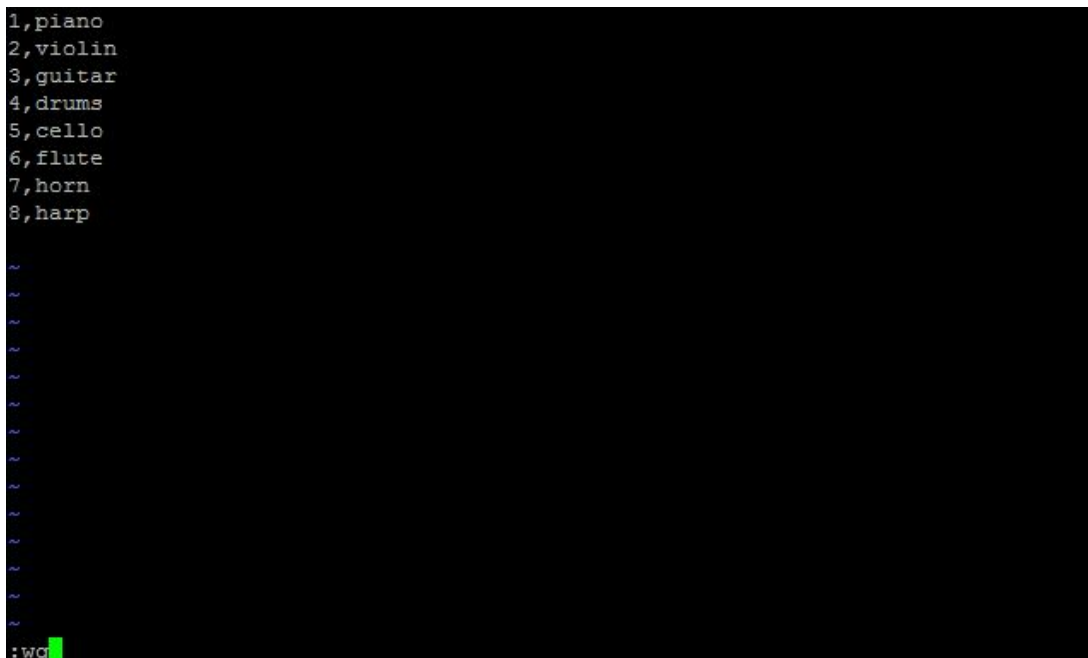
1,1 All

```
1,piano
2,violin
3,guitar
4,drums
5,cello
6,flute
7,horn
8,harp
9,
-- INSERT --
```

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**):

A screenshot of the Vim text editor interface. The background is black, and the text is white. The first eight lines of the file contain the following text: 1,piano, 2,violin, 3,guitar, 4,drums, 5,cello, 6,flute, 7,horn, 8,harp. Below these lines are several lines of tilde (~) characters, indicating empty lines in the file. At the bottom of the screen, the command prompt is visible, showing a colon followed by the text 'wq'. A green cursor is positioned at the end of the 'wq' text, indicating that the user is currently entering the command.

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**:

```
id,instrument
-- INSERT -- 1,14 All
```

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`:

```
ubuntu$cat header.txt instruments.txt
```

Объединенное текстовое содержимое этих двух файлов будет выведено на экран:

```
ubuntu$cat header.txt instruments.txt
id,instrument
1,piano
2,violin
3,guitar
4,drums
5,cello
6,flute
7,horn
8,harp
ubuntu$
```

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**:

```
ubuntu$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:

```
ubuntu$wc -l Shakespeare.txt
213 Shakespeare.txt
ubuntu$
```

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

```
ubuntu$cat *.txt | wc -l
233
ubuntu$
```

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](#).
2. [Текстовый редактор nano в Linux для новичков](#).
3. [Nano: И все-таки его придется выучить](#).
4. [Команда Cat в Linux. Основные и расширенные примеры](#).

Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. [Команда cat Linux](#).
2. [13 Basic Cat Command Examples in Linux](#).
3. [Редактор nano](#).