Москва 2018

**Содержание**

Введение…………………………………………………………….5

1. Оболочкa bash…………………………………………………...7
2. Команды категории «Print File Contents»………………………7

2.1. Команда head …………………………………………….…7

2.2. Команда tail…………………………………………………8

2.3. Команда less………………………………………………...10

2.4. Команда more……………………………………………....14

2.5. Команда cat…………………………………………………16

2.6. Команда tee…………………………………………………20

Заключение…………………………………………………………22

Список литературы………………………………………………...23

**Введение**

Linux - это современная POSIX-совместимая и Unix-подобная операционная система для персональных компьютеров и рабочих станций.

Это многопользовательская сетевая операционная система с сетевой оконной графической системой X Window System. ОС Linux поддерживает стандарты открытых систем и протоколы сети Internet и совместима с системами Unix, DOS, MS Windows. Все компоненты системы, включая исходные тексты, распространяются с лицензией на свободное копирование и установку для неограниченного числа пользователей.

Операционная система Linux широко распространена на платформах Intel PC 386/486/Pentium/Pentium Pro и завоевывает позиции на ряде других платформ.

**Возможности, которые предоставляет ОС Linux**

* дает возможность бесплатно и легально иметь современную ОС для использования как на работе, так и дома;
* обладает высоким быстродействием;
* работает надежно, устойчиво, совершенно без зависаний;
* не подвержена вирусам;
* позволяет использовать полностью возможности современных ПК, снимая ограничения, присущие DOS и MS Windows по использованию памяти машины и ресурсов процессора(ов);
* эффективно управляет многозадачностью и приоритетами, фоновые задачи (длительный расчет, передача электронной почты по модему, форматирование дискеты и т.д. и т.п.) не мешают интерактивной работе;
* позволяет легко интегрировать компьютер в локальные и глобальные сети, в т.ч. в Internet; работает с сетями на базе Novell и MS Windows;
* позволяет выполнять представленные в формате загрузки прикладные программы других ОС - различных версий Unix, DOS и MS Windows;
* обеспечивает использование огромного числа разнообразных программных пакетов, накопленных в мире Unix и свободно распространяемых вместе с исходными текстами;
* предоставляет богатый набор инструментальных средств для разработки прикладных программ любой степени сложности, включая системы класса клиент-сервер, объектно-ориентированные, с многооконным текстовым и/или графическим интерфейсом, пригодных для работы как в Linux, так и в других ОС;
* дает пользователю и особенно разработчику замечательную учебную базу в виде богатой документации и исходных текстов всех компонент, включая ядро самой ОС;
* дает всем желающим попробовать свои силы в разработке, организовать общение и совместную работу через Internet с любыми из разработчиков ОС Linux и сделать свой вклад, став соавтором системы.

Цель курсовой работы: познакомиться с оболочкой bash и утилитами, используемыми it – специалистом в повседневной жизни.

1. **Оболочкa bash**

Оболочка, или шелл (shell) — это программа, в нашем случае названная «bash», что является сокращением от Bourne Again Shell. Оболочка принимает ваши команды и передаёт их операционной системе. Для взаимодействия с системой используются терминалы.

1. **Команды категории «Print File Contents»**

В данной работе мы рассмотрим подробно команды категории «Print File Contents». Команды из этого раздела предназначены для создания и удаления файлов и директорий и навигации между ними. Более подробно мы рассмотрим head, tail, less, more, cat, tree.

**2.1 Команда head**

Команда head по умолчанию выводит первые 10 строк файла.

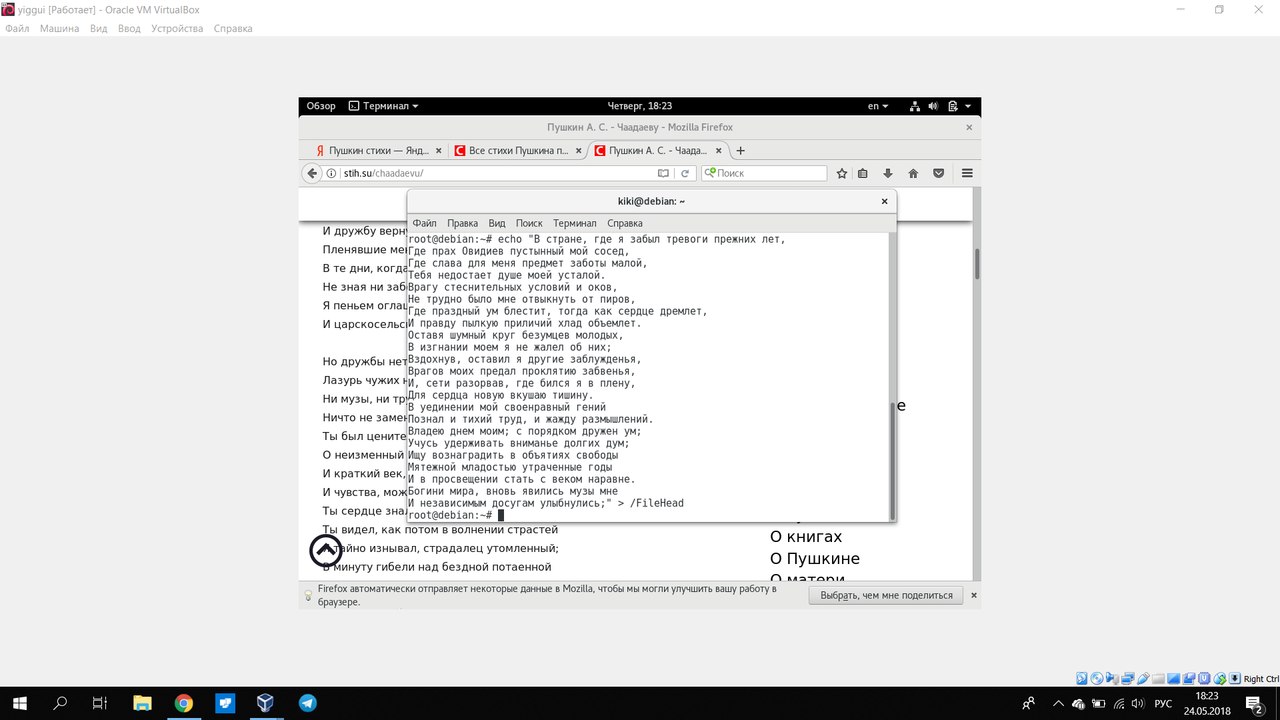
Ключи:

* -с– позволяет указать точное количество байт, которое необходимо вывести на экран;
* -n – позволяет указать точное количество строк, которое необходимо вывести на экран.

Создание файла с которым будем работать:

используем **команду** echo

|  |
| --- |
| $ echo “текст” > наш\_файл |

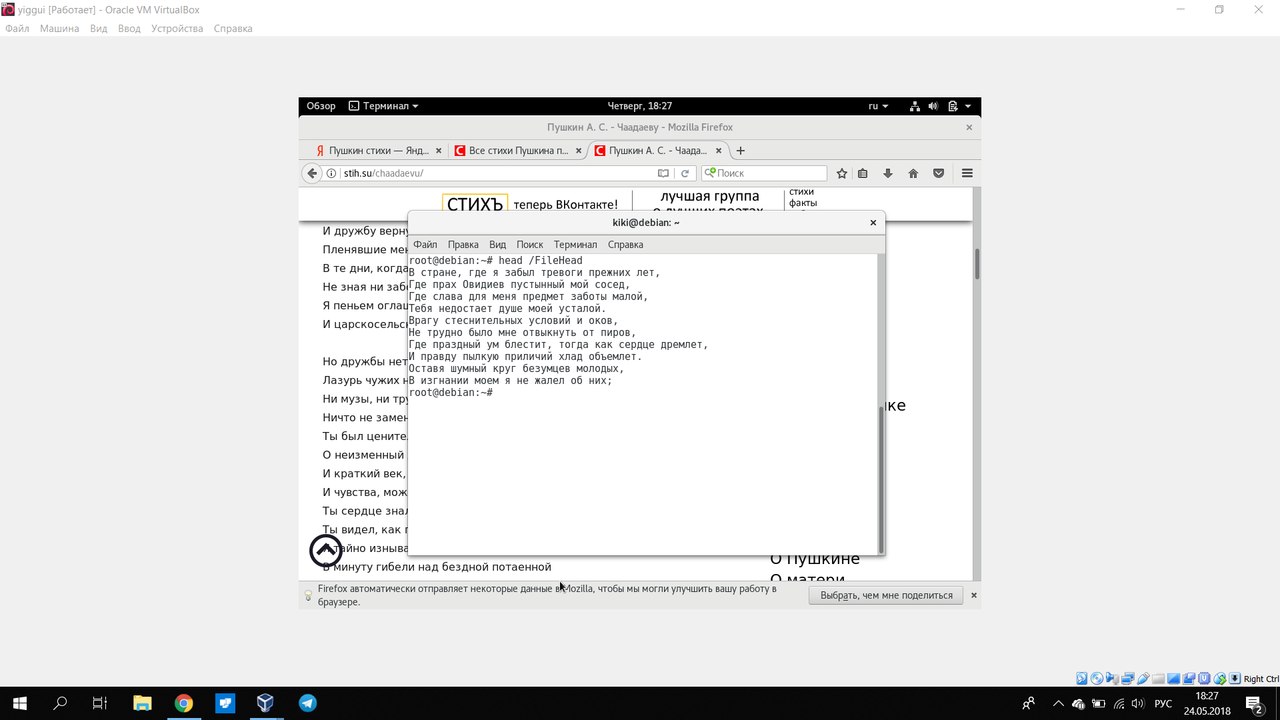


**Задача:** вывести на экран первые 10 строк файла.

**Команда:**

|  |
| --- |
| $ head /FileHead |

**Результат:**

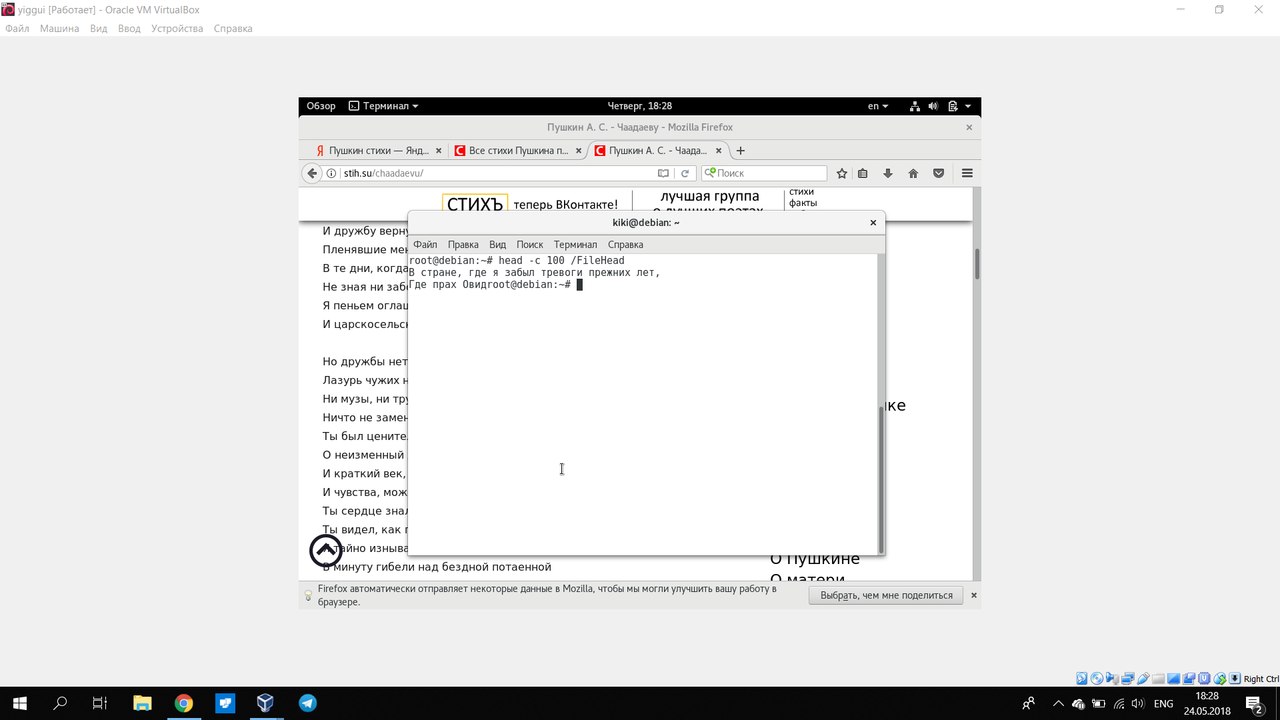


**Задача:** вывести на экран первые 100 байт файла.

**Команда:**

|  |
| --- |
| $ head –c 100 /FileHead |

**Результат:**

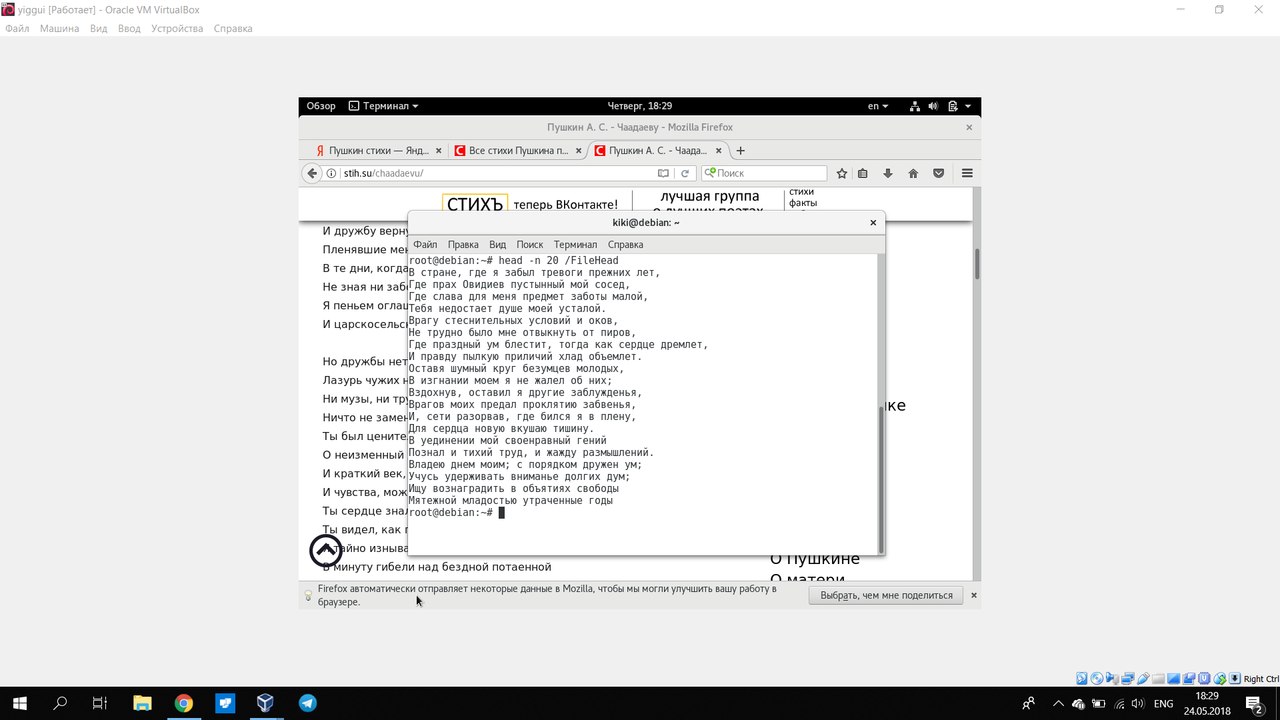


**Задача:** вывести на экран первые 20 строк файла.

**Команда:**

|  |
| --- |
| $ head –n 20 /FileHead |

**Результат:**



**2.2 Команда tail**

Команда, обратная по действию команде head — tail. Используя команду tail, вы можете увидеть последние десять строк файла. Это может оказаться полезным, если, например, вы хотите просмотреть последние строки файла журнала системных сообщений. Также при помощи программы tail можно наблюдать за изменением журнала.

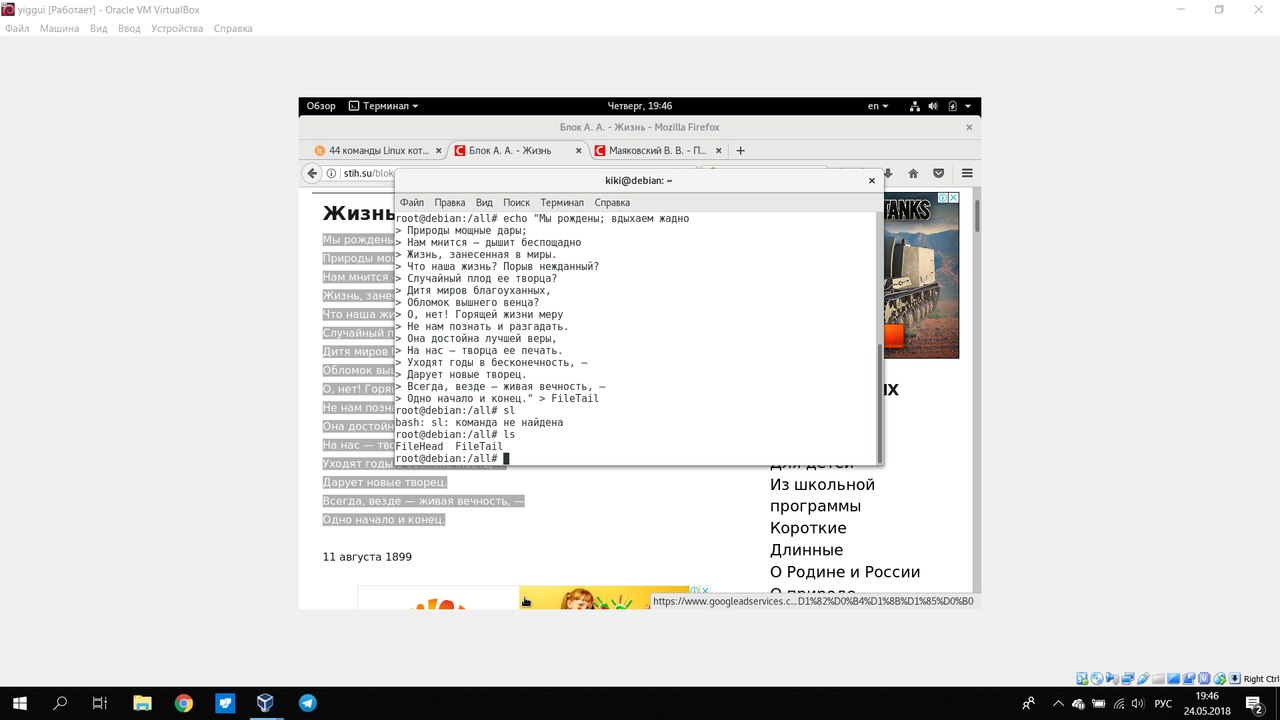
Ключи:

* -n N – позволяет указать точное количество строк, которое необходимо вывести на экран (N – число);
* -f - показывает последние 10 строк файла. Команда продолжает показывать новые строки по мере их добавления к выбранному файлу. Для окончания просмотра нажмите Ctrl + C. Для просмотра системных файлов необходимо быть суперпользователем.

Создание файла с которым будем работать:

используем **команду** echo

|  |
| --- |
| $ echo “текст” > наш\_файл |

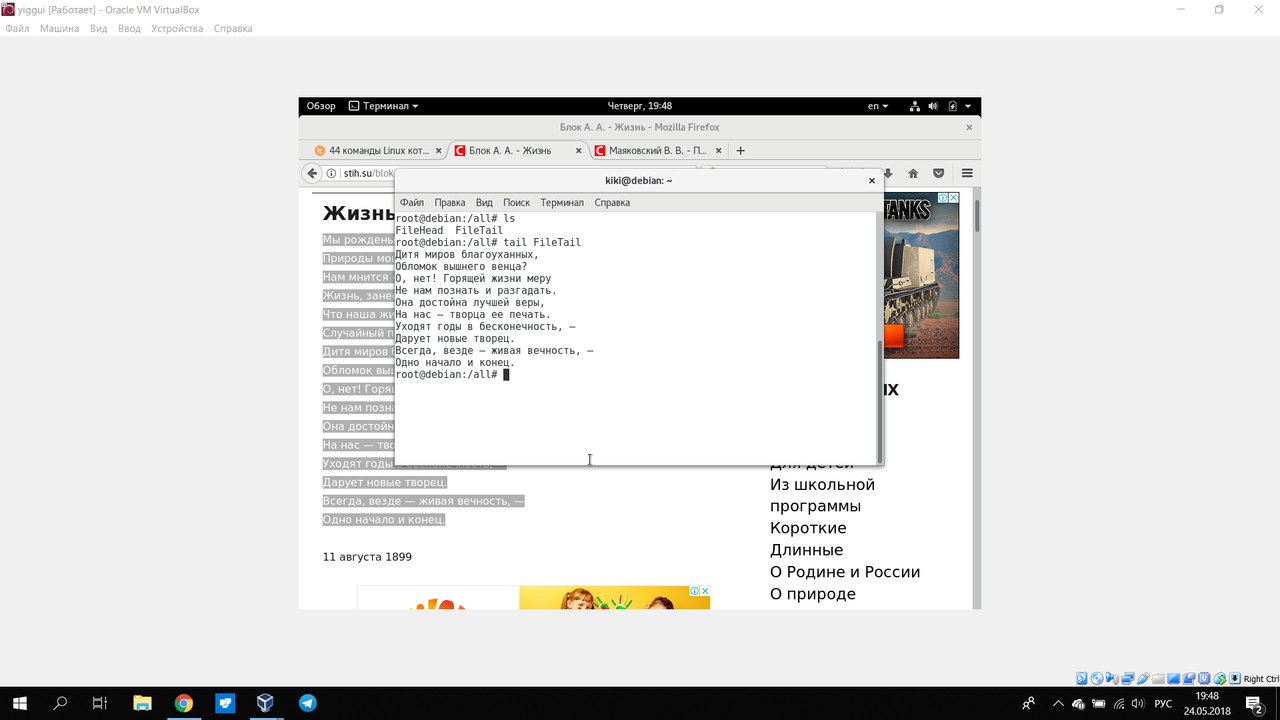


**Задача:** вывести последние 10 строк файла.

**Команда:**

|  |
| --- |
| $ tail FileTail |

**Результат:**

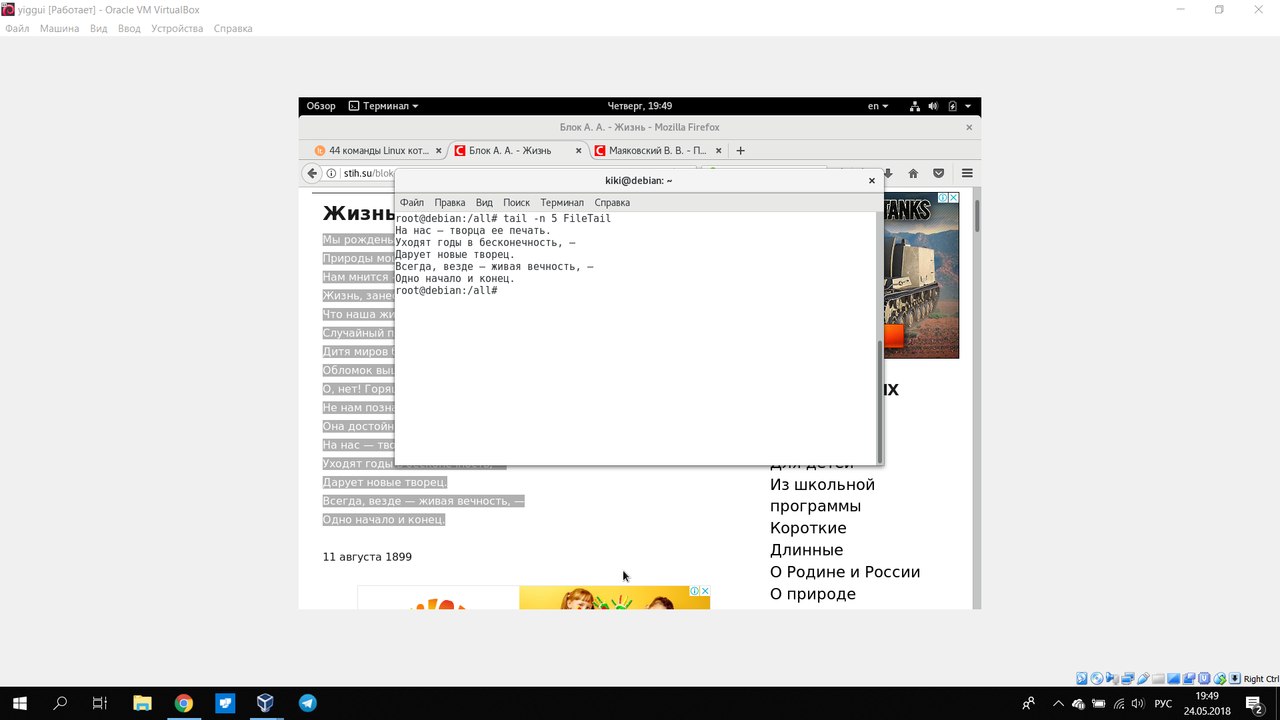


**Задача:** вывести на экран последние 5 строк файла.

**Команда:**

|  |
| --- |
| $ tail –n 5 FileTail |

**Результат:**

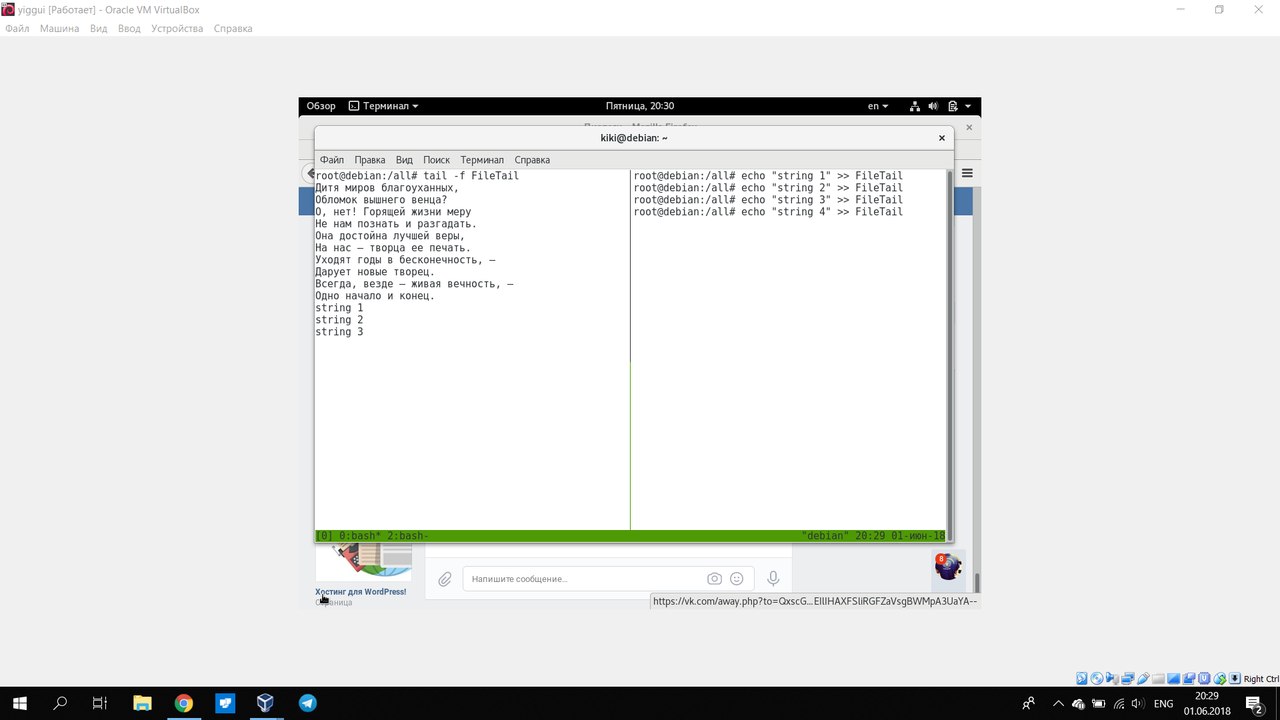


**Задача:** обновлять информацию по мере появления новых строк в файле.

**Команда:**

|  |
| --- |
| $ tail –f FileTail |

**Результат:**



Отличительной чертой команды является ее возможность не завершаться по достижении конца вывода, а ожидать появления новых данных.

**2.3 Команда less**

Команда less осуществляет поиск по данным в файле.

Ключи:

* -р текст - указывает первую строку, в которой найден текст;
* -N - показывать номера строк;
* -s - сжимает несколько пустых строк в одну пустую строку.

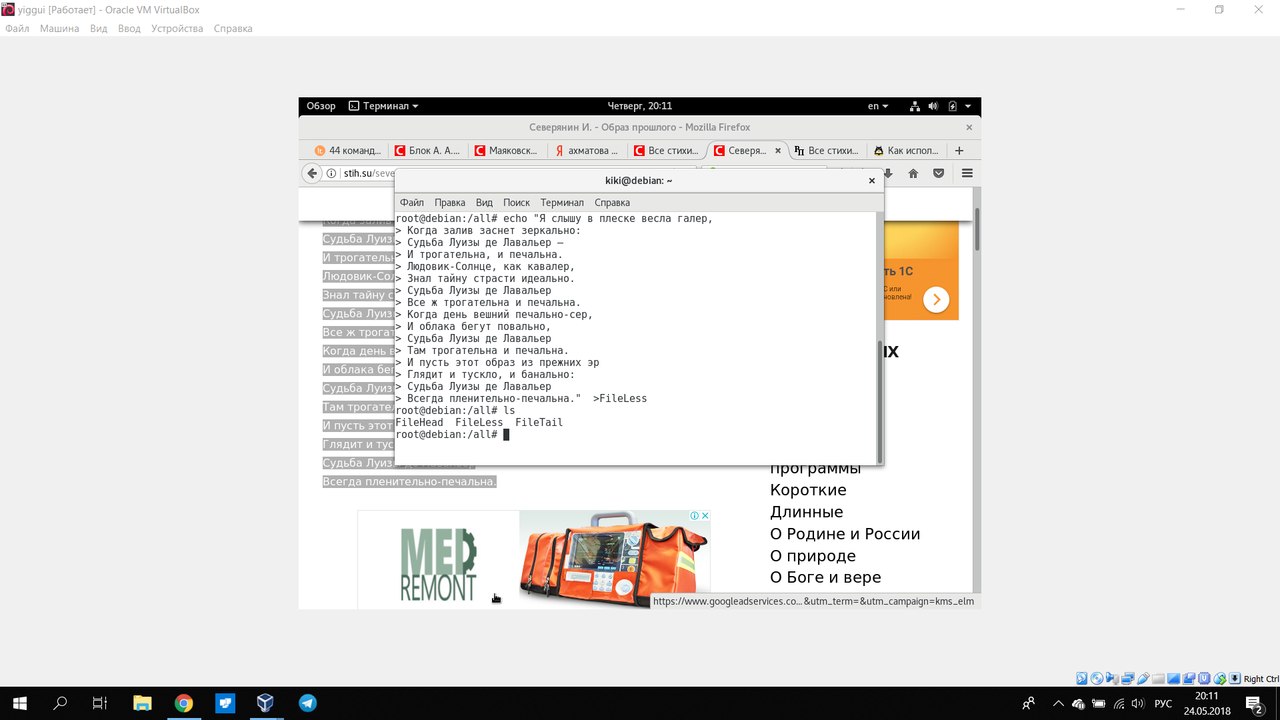
Работа с файлом осуществляется с помощью команд:

|  |  |
| --- | --- |
| **Команда** | **Назначение** |
| $ \uparrow$ | Прокрутка текста на строку назад. |
| $ \downarrow$ | Прокрутка текста на строку вперед. |
| $ \leftarrow$ или$ \rightarrow$ | Прокрутка текста влево или вправо на позицию табуляции. |
| PgUp или SPC | Прокрутка текста вперед на страницу. |
| PgDn | Прокрутка текста назад на страницу. |
| Ctrl-l | Перерисовка экрана. |
| /pattern | Поиск строк, содержащих образец, вперед по тексту внутри файла |
| < | Перемещение к началу файла. |
| > | Перемещение к концу файла. |
| G | Перейти к концу документа. |
| q | Прекращение просмотра и выход из less. |

Создание файла с которым будем работать:

используем **команду** echo

|  |
| --- |
| $ echo “текст” > наш\_файл |

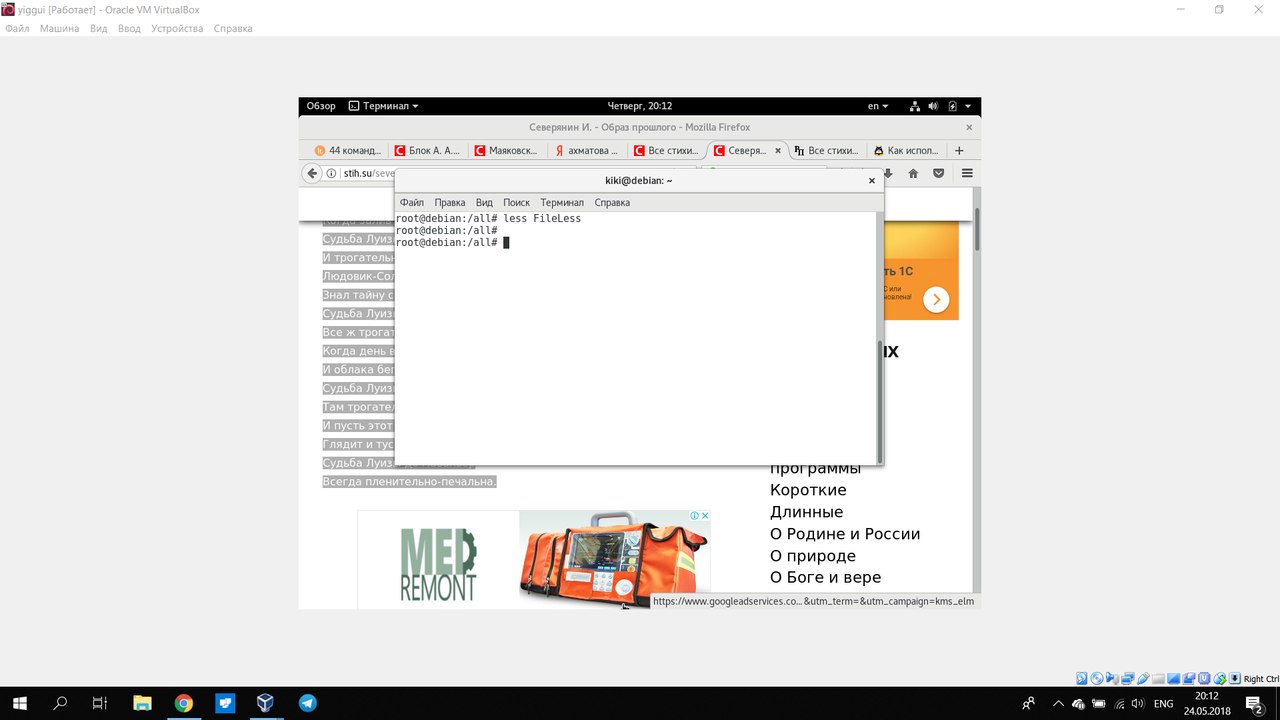


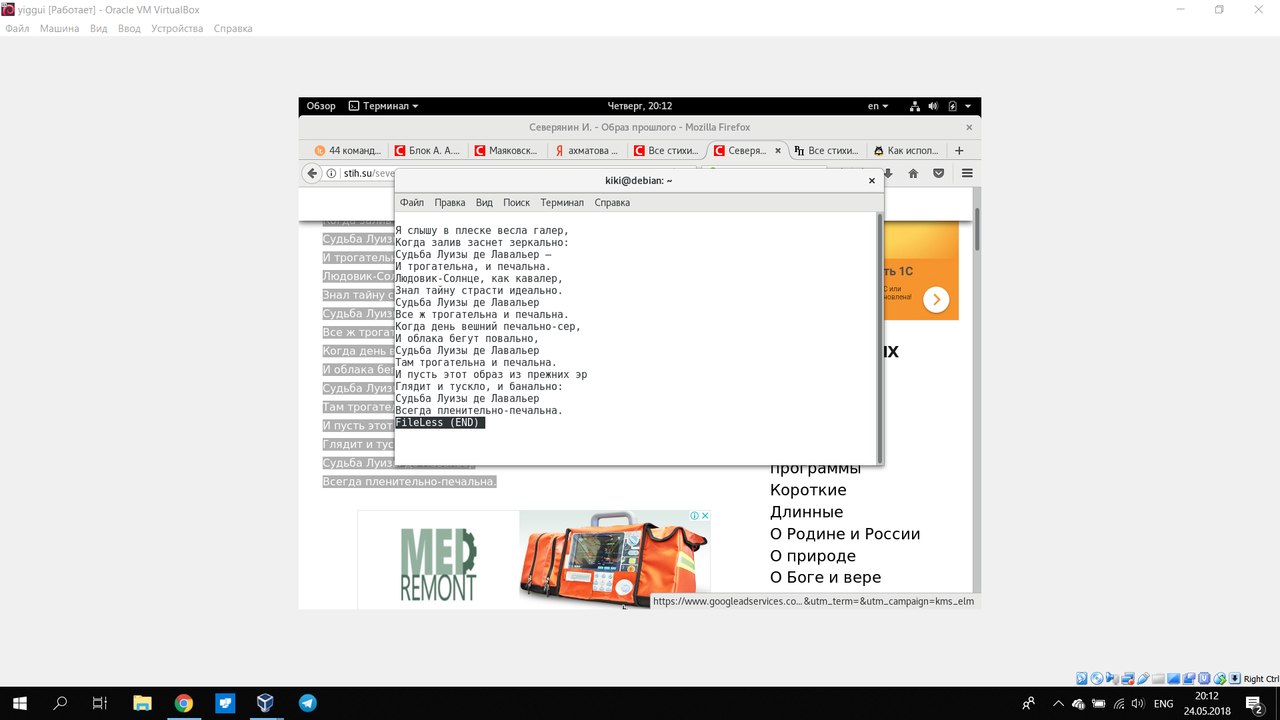
**Задача:** вывести на экран текстовый файл.

**Команда:**

|  |
| --- |
| $ less FileLess |

**Результат:**



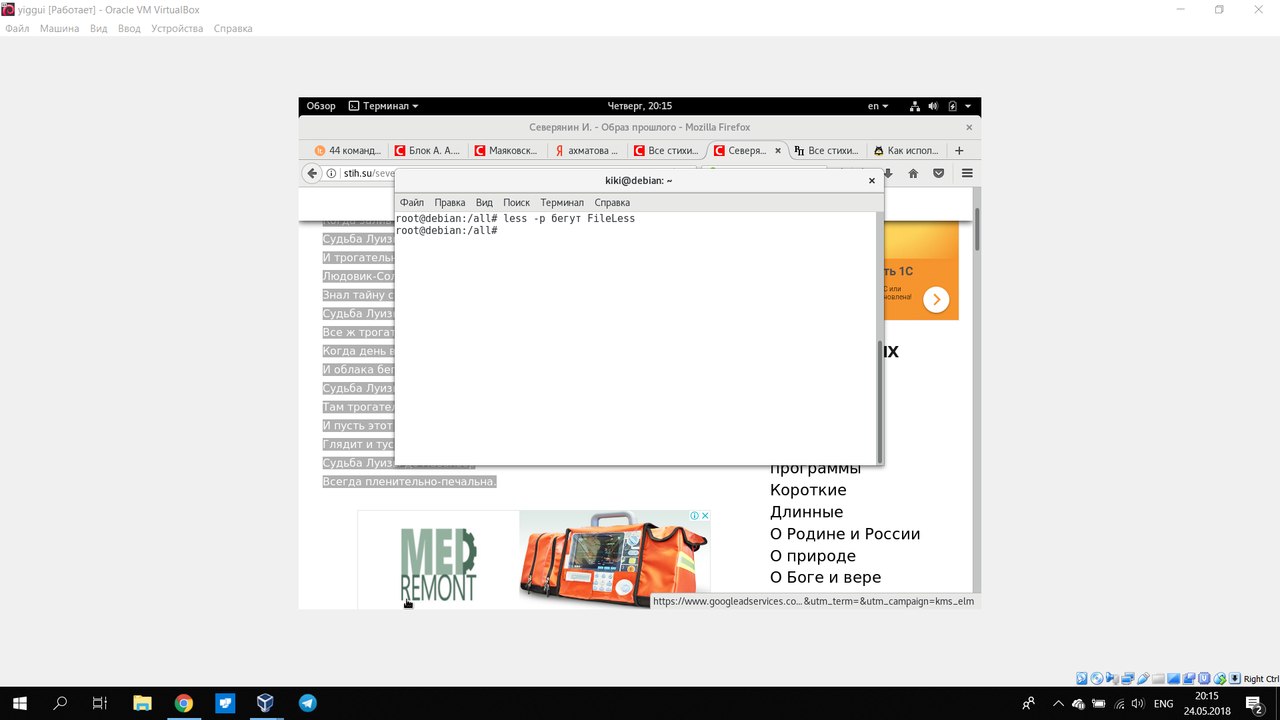


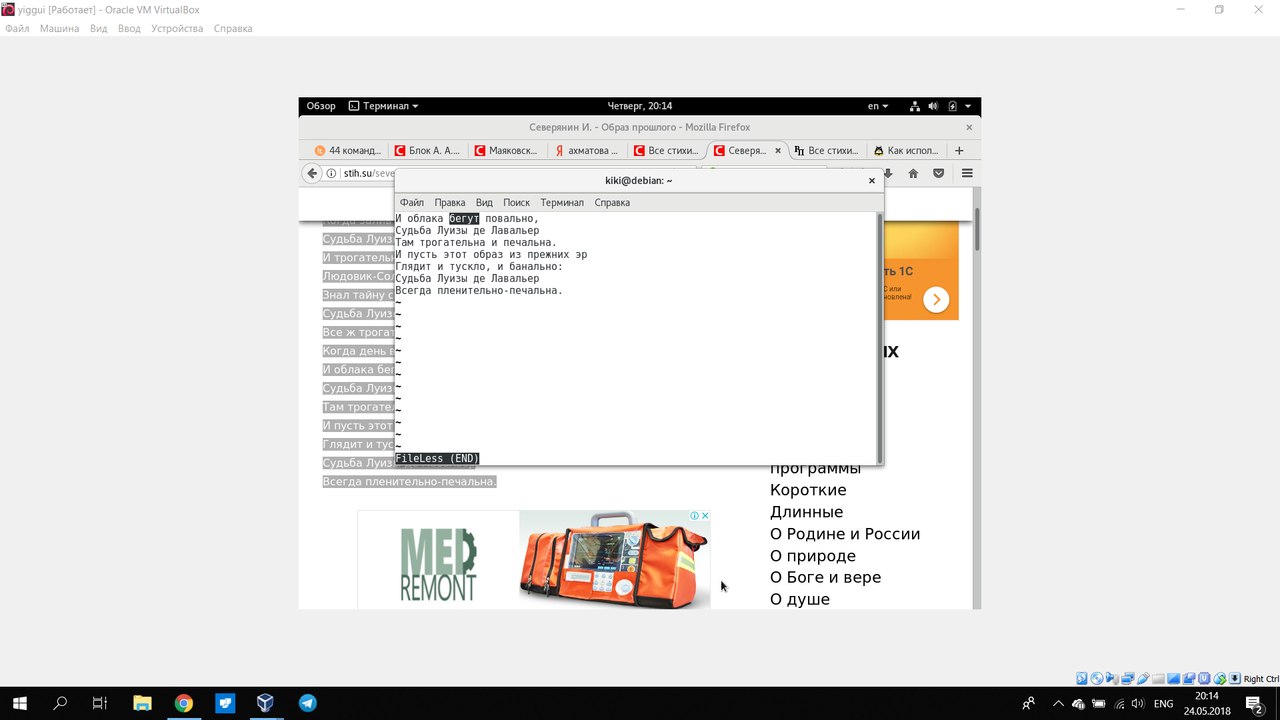
**Задача:** найти в файле текст «бегут».

**Команда:**

|  |
| --- |
| $ less –p бегут FileLess |

**Результат:**



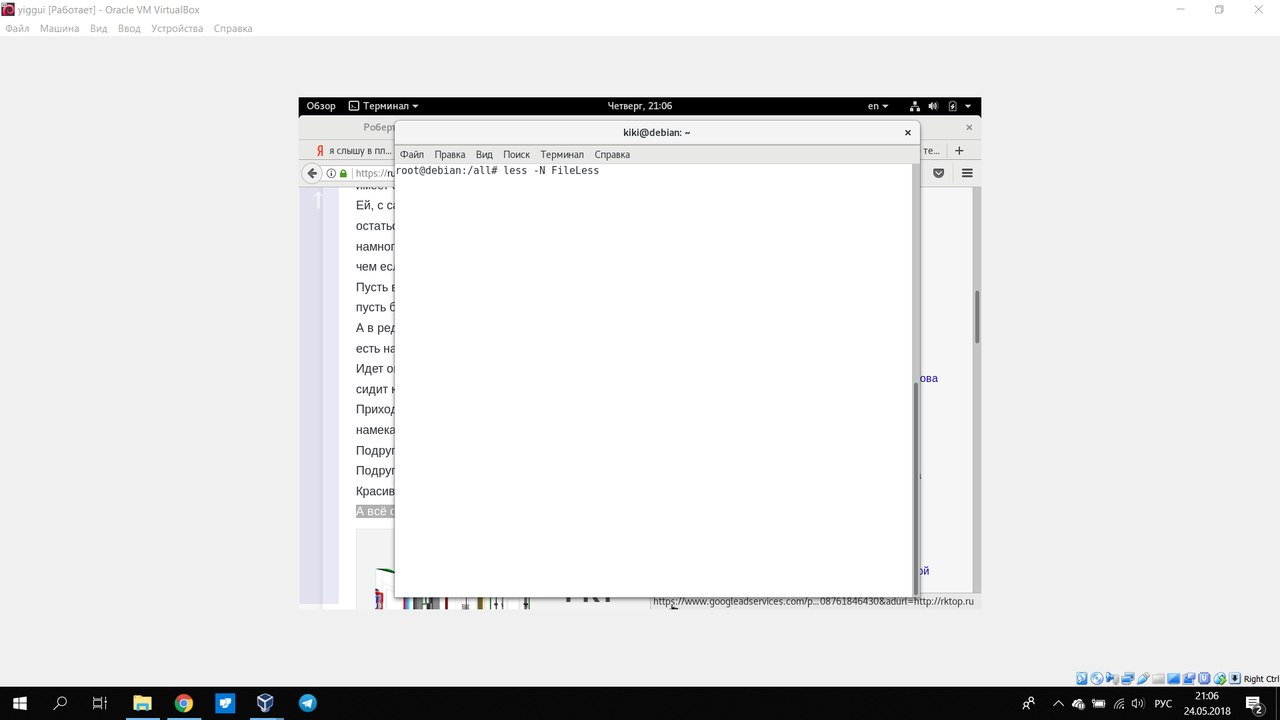


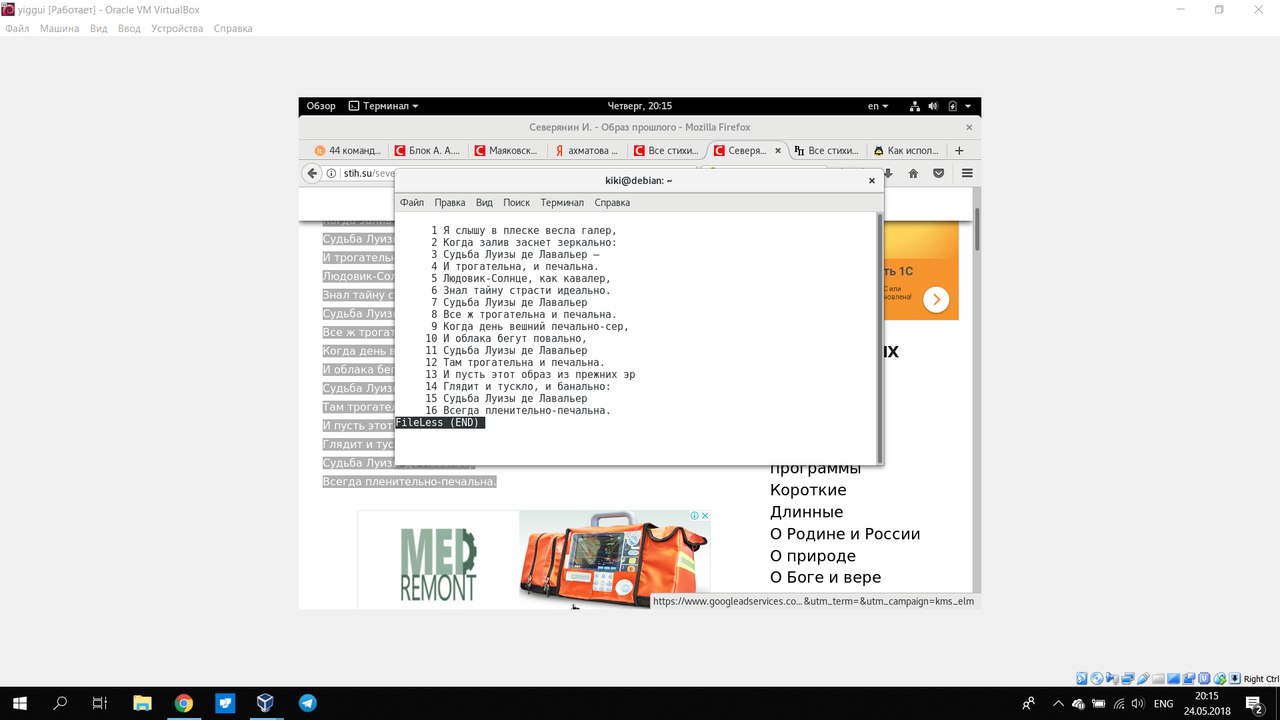
**Задача:** вывести текст с пронумерованными строками.

**Команда:**

|  |
| --- |
| $ less –N FileLess |

**Результат:**



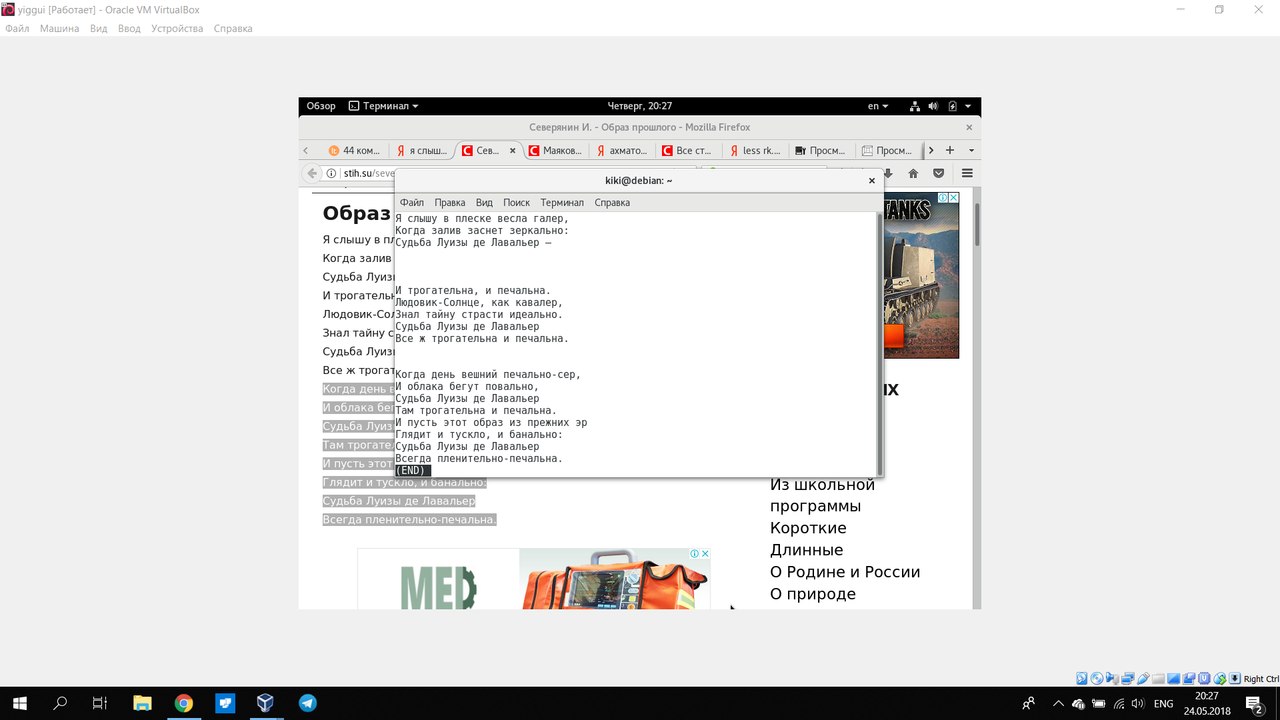


**Задача:** сжать несколько пустых строк в одну.

Добавим пустые строки в наш файл:

**команда** echo

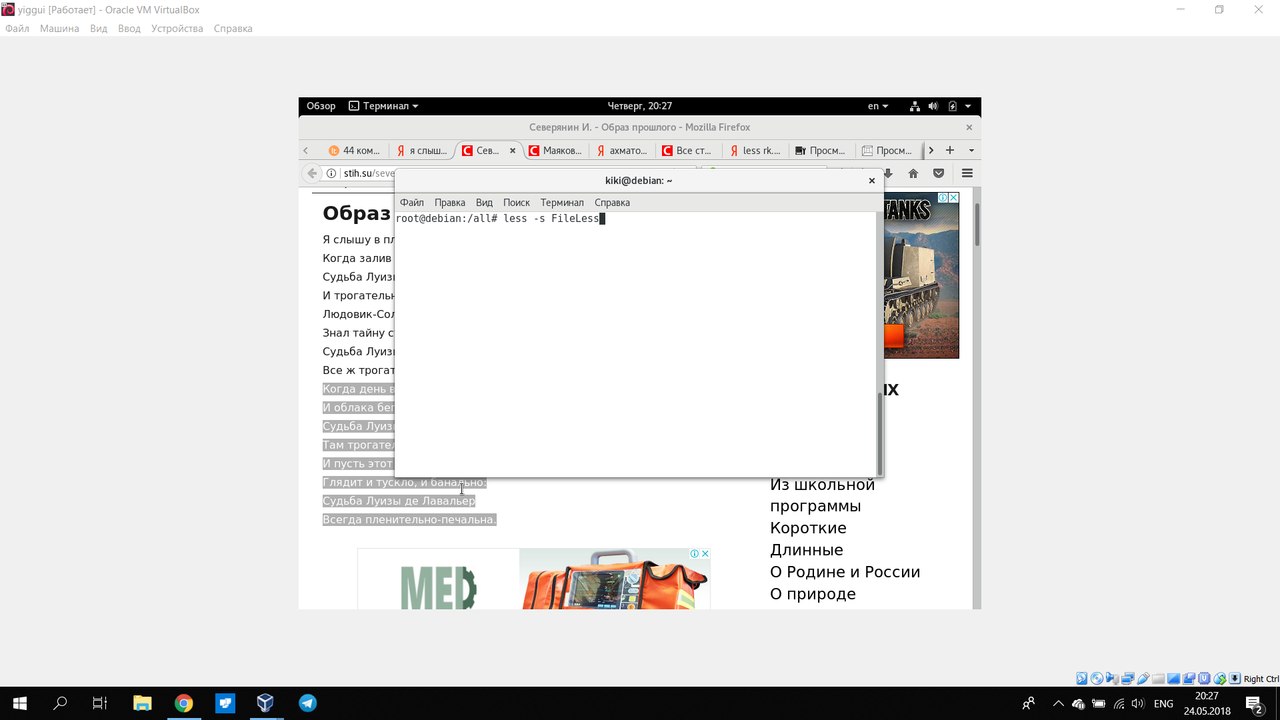
|  |
| --- |
| $ echo “текст” > наш\_файл |

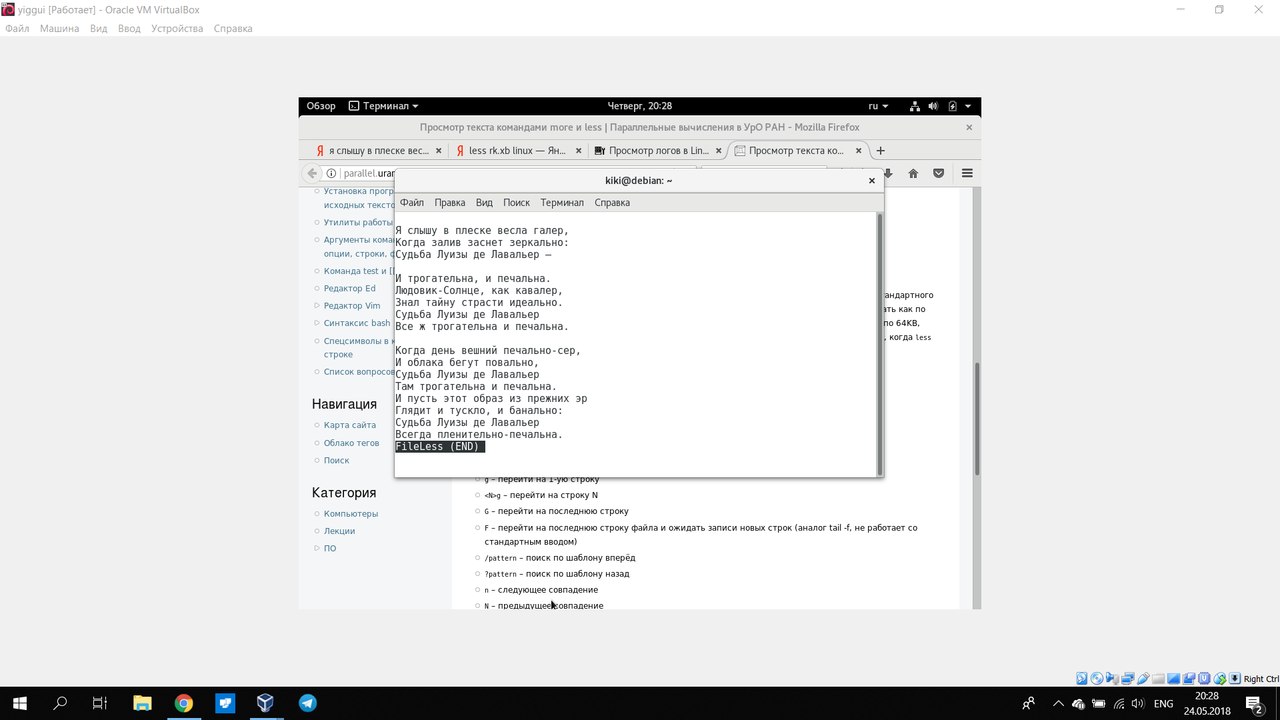


**Команда:**

|  |
| --- |
| $ less –s FileLess |

**Результат:**

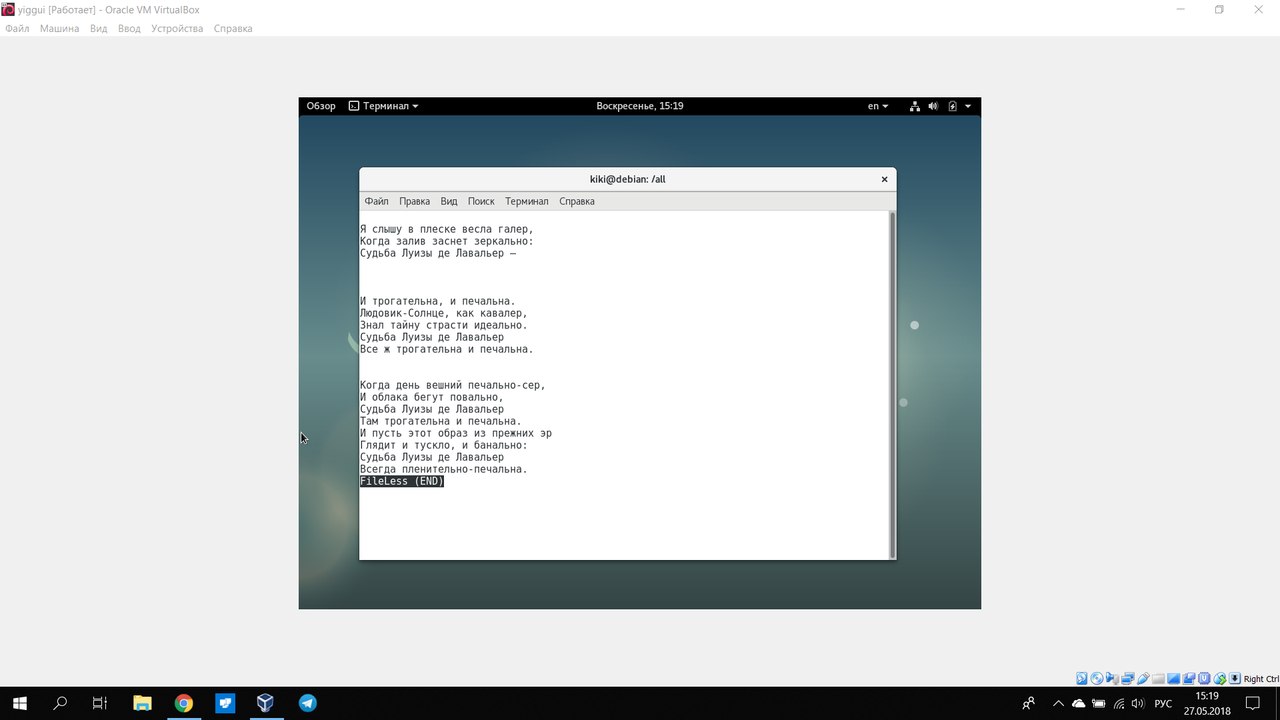




**Задача:** найти подстроку «пусть» внутри нашего файла.

Откроем наш файл **командой**

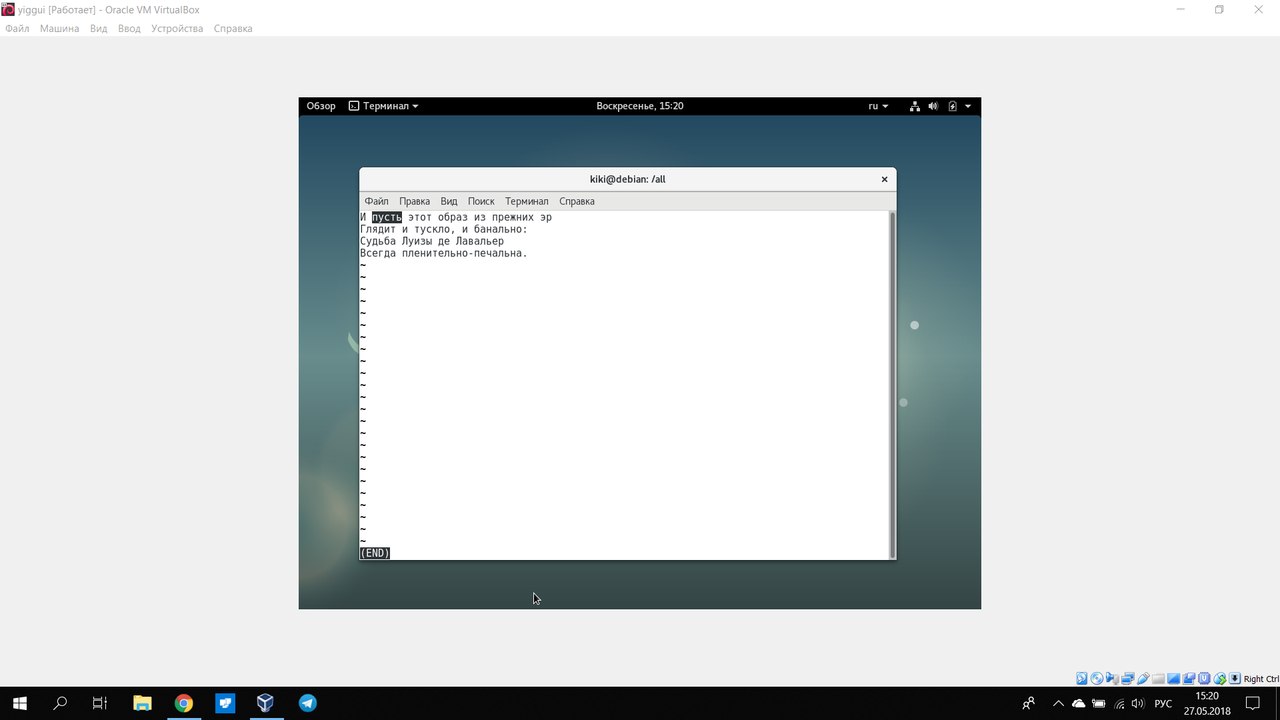
|  |
| --- |
| $ less FileLess |



Введем в открывшемся файле **команду**

|  |
| --- |
| /пусть |

**Результат:**



**2.4 Команда more**

Команды more и less тесно связаны и функционально подобны. Основное назначение этих команд - вывод на экран или в окно текста большого объема по частям, чтобы его можно было листать в обоих направлениях, а также поиск в этом тексте.

Команда more – команда постраничной или построчной прокрутки большого текста на экране и поиска в этом тексте.

Ключи:

* -uN – вывод первых N строк;
* -s - сжать каждый пустой блок пустых строк в одну строку.

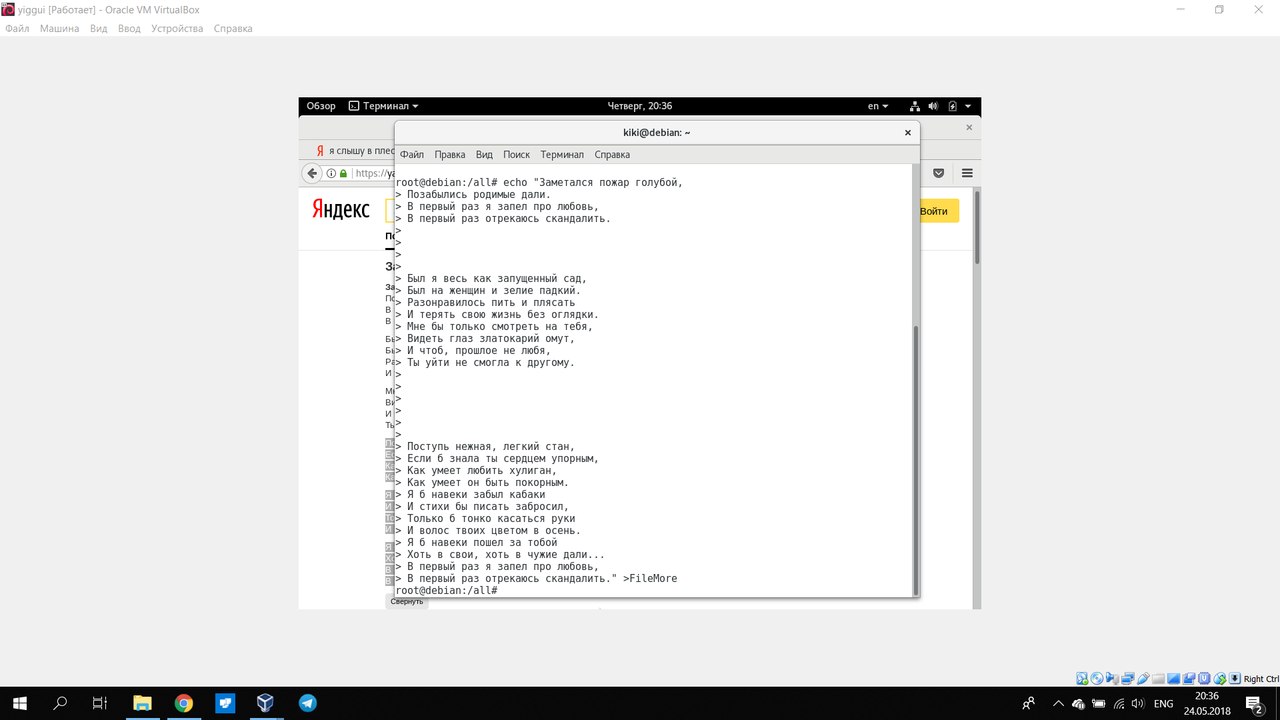
Работа с файлом осуществляется с помощью команд:

|  |  |
| --- | --- |
| SPACE | пролистать один экран |
| d | пролистать пол экрана |
| /pattern | пролистать до строки содержащей подстроку pattern |
| n | повторить поиск подстроки |
| = | показать номер текущей строки |
| :f | напечатать имя текущего файла и номер строки в нем |

Создание файла с которым будем работать:

используем **команду** echo

|  |
| --- |
| $ echo “текст” > наш\_файл |

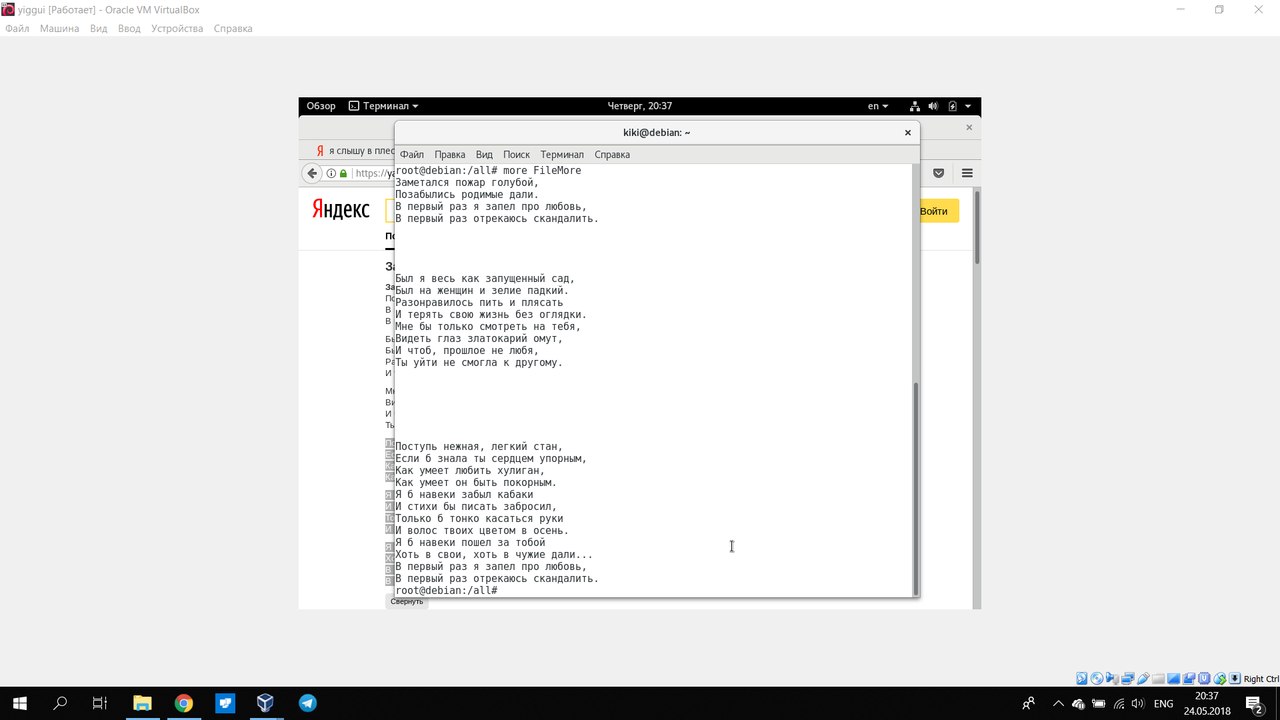


**Задача:** вывести на экран текстовый файл.

**Команда:**

|  |
| --- |
| $ more FileMore |

**Результат:**

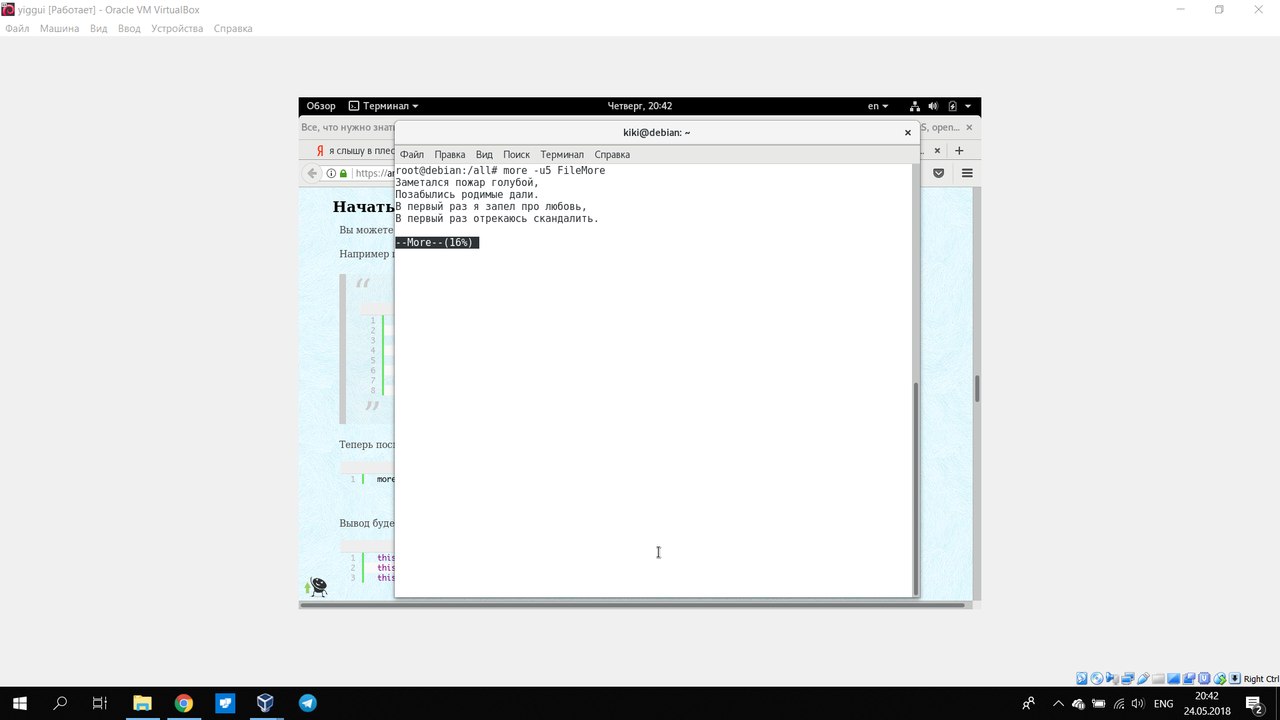


**Задача:** вывести первые 5 строк файла.

**Команда:**

|  |
| --- |
| $ more –u5 FileMore |

**Результат:**

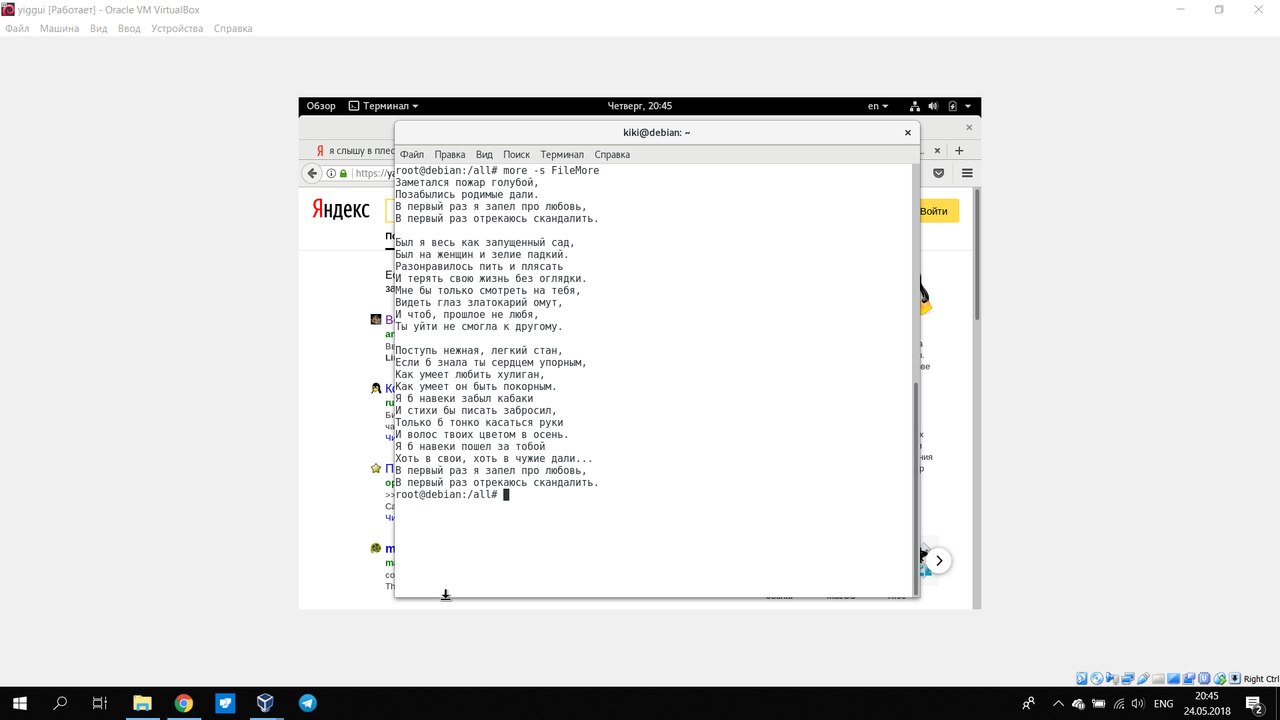


**Задача:** сжать несколько пустых строк в одну.

**Команда:**

|  |
| --- |
| $ more –s FileMore |

**Результат:**



**2.5 Команда cat**

Команда cat – вывести файл на stdout, также существует возможность перенаправить потоки ввода - вывода с помощью >, <.

Ключи:

* -b – нумеровать только непустые строки;
* -E - показывать символ $ в конце каждой строки;
* -s - удалять пустые повторяющиеся строки.

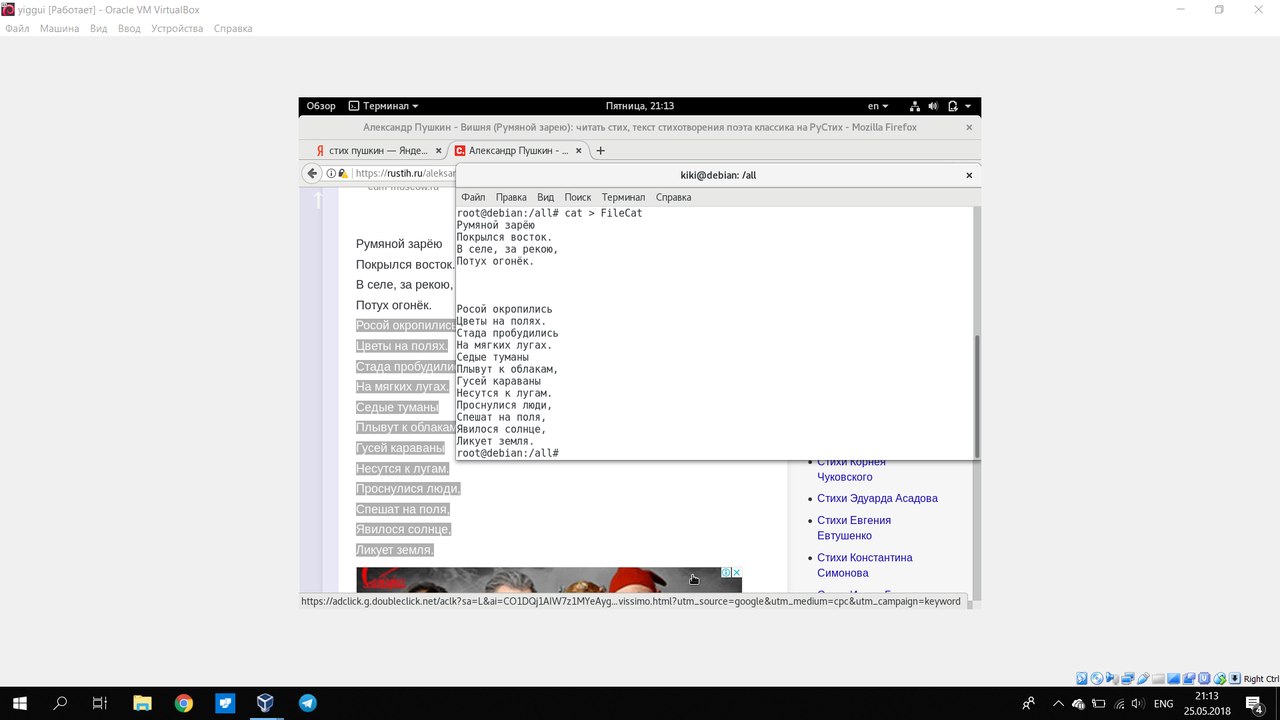
**Задача:** создать файл с помощью перенаправления потока вывода.

**Команда:**

|  |
| --- |
| $ cat > FileCat |

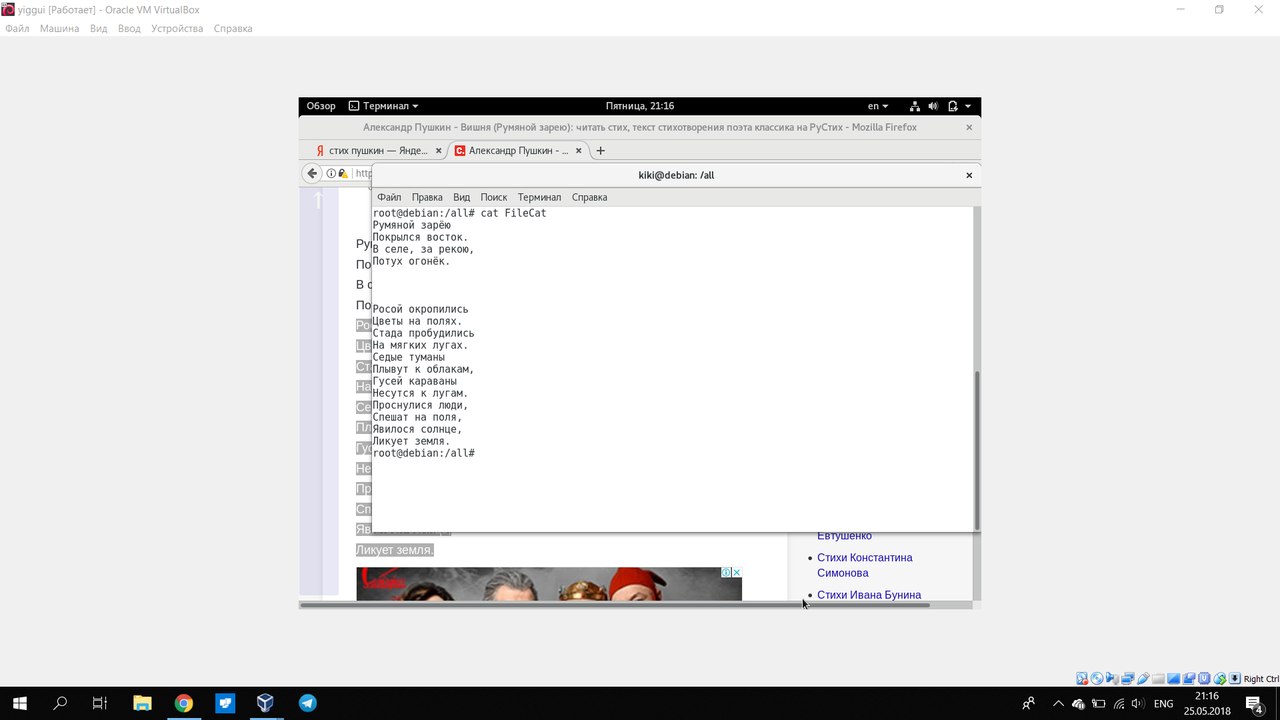
Ожидая ввод от пользователя, наберите необходимый текст и нажмите **Ctrl + D,** чтобы выйти. Текст будет записан в указанный файл.

**Результат:**



Посмотрим содержимое файла, набрав **команду**

|  |
| --- |
| **$ cat** FileCat |



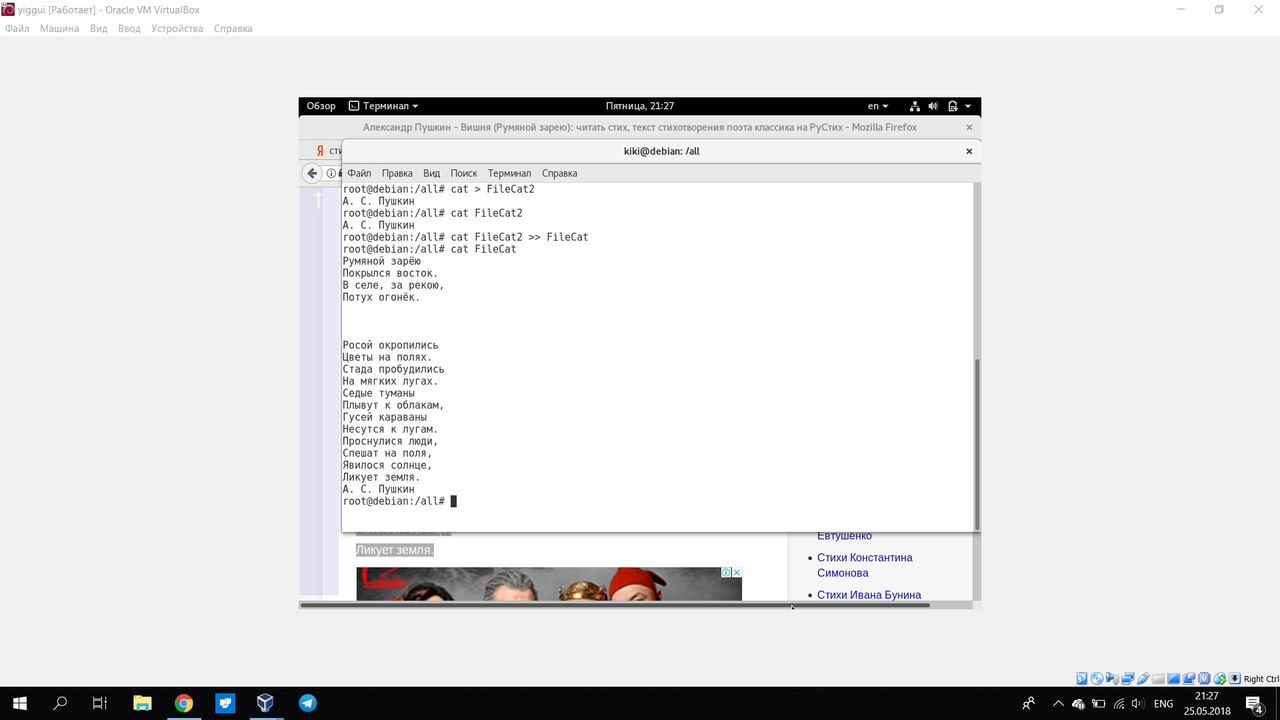
**Задача:** добавить в наш файл содержимое другого файла

Для начала создадим новый файл тем же способом что и первый. Затем, с помощью **команды**

|  |
| --- |
| $ cat FileCat2 >> FileCat |

добавим содержимое второго файла к первому.

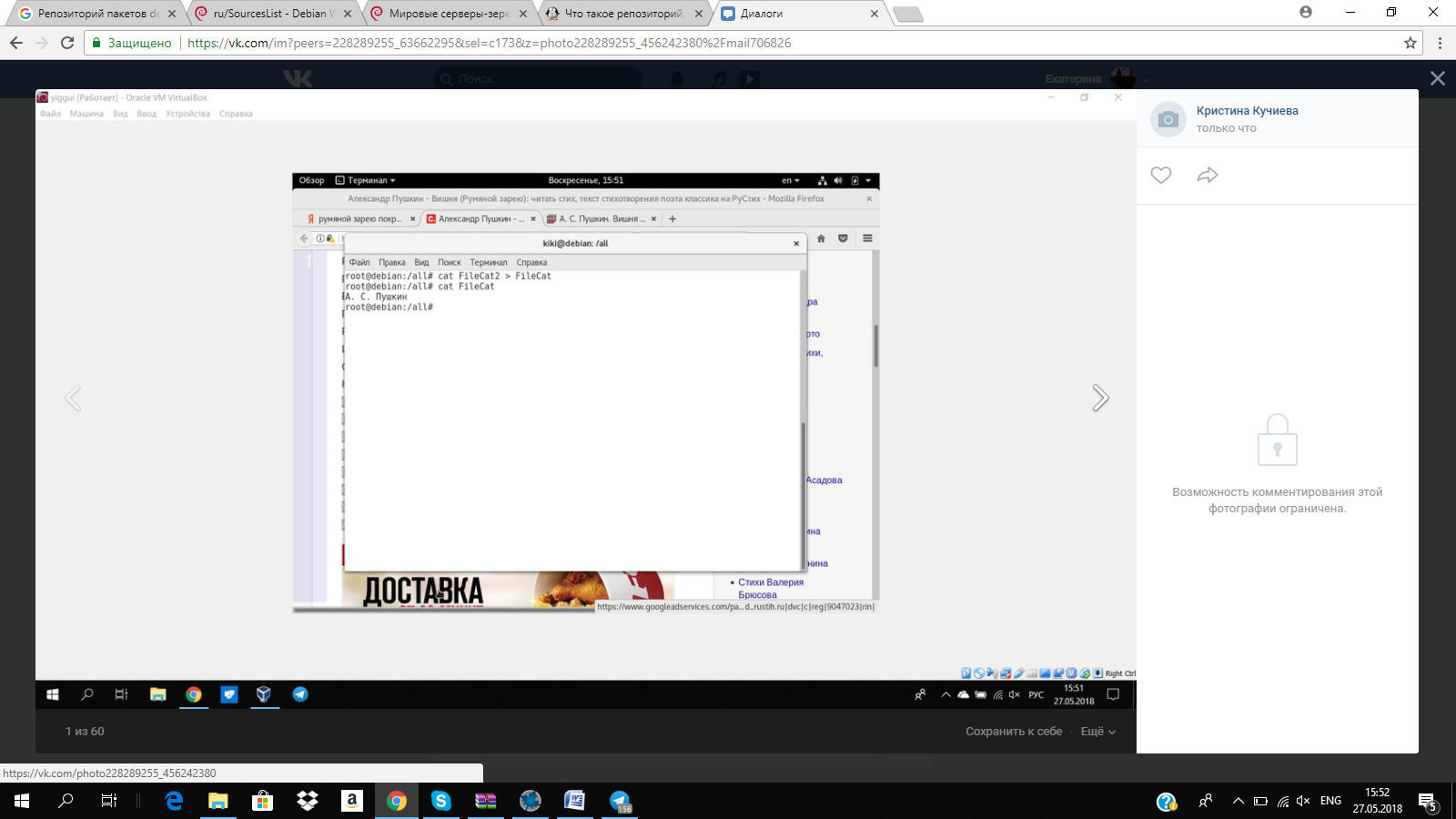
**Результат:**



Также можно перезаписать файл **командой**

|  |
| --- |
| $ cat FileCat2 > FileCat |

**Результат:**

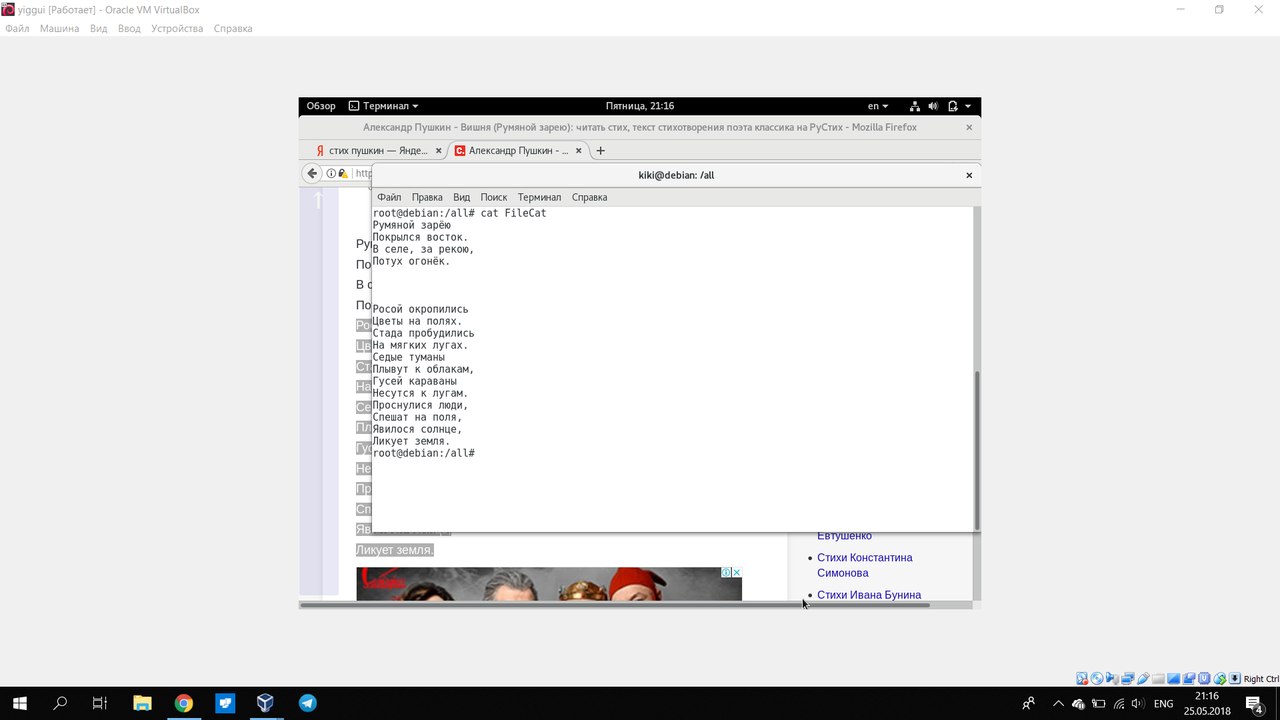


**Задача:** вывести на экран файл.

**Команда:**

|  |
| --- |
| $ cat FileCat |

**Результат:**

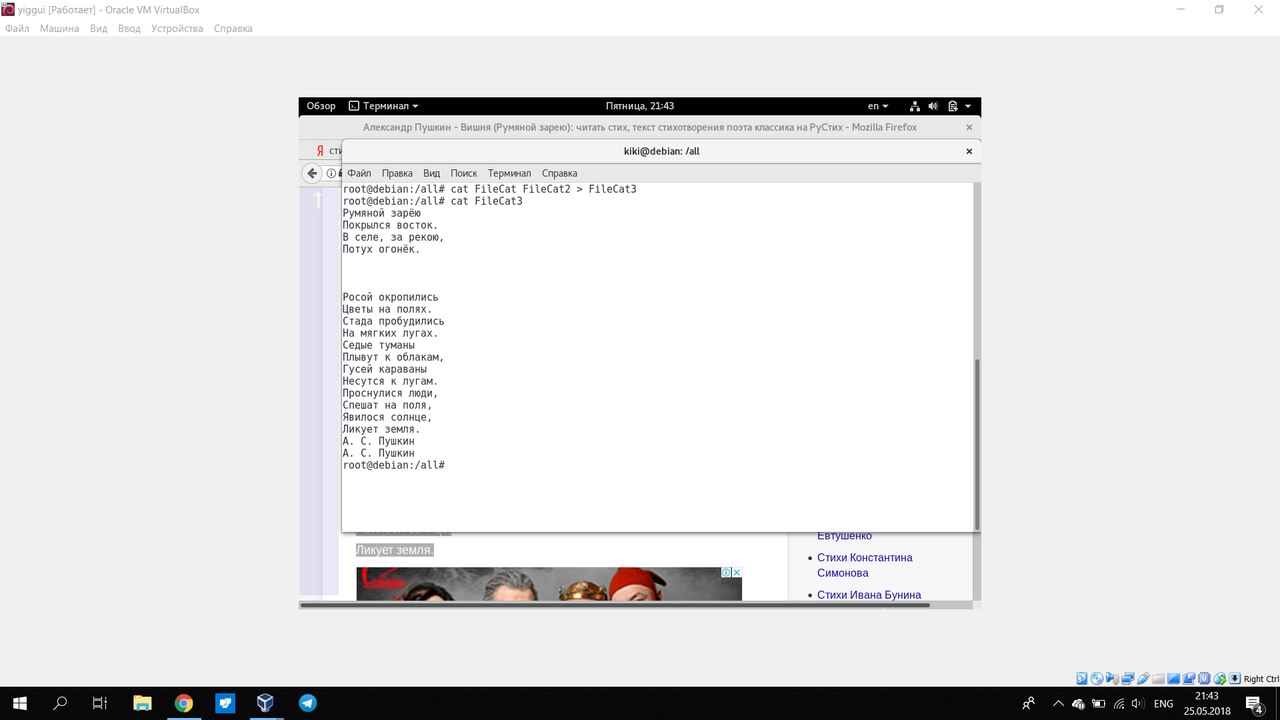


**Задача:** создать файл с записью в него содержимого нескольких других файлов.

**Команда:**

|  |
| --- |
| $ cat FileCat FileCat2 > FileCat3 |

**Результат:**



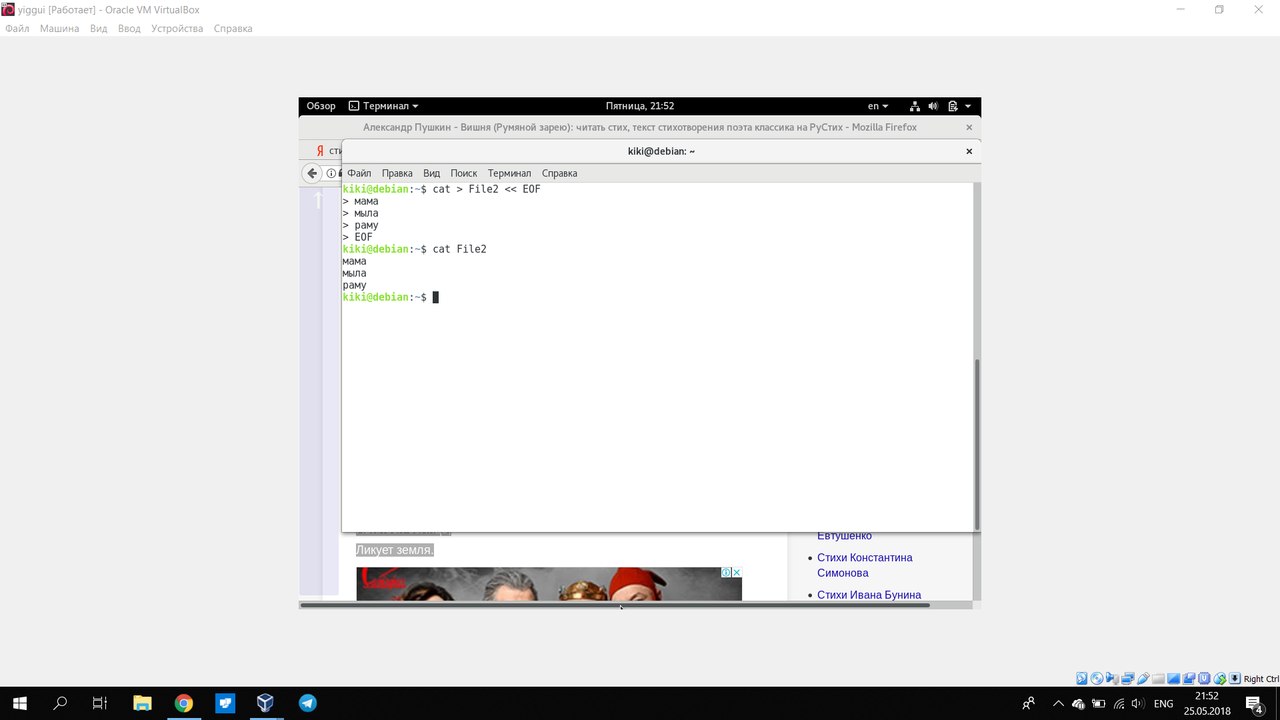
**Задача:** записать многострочный набор данных с отступами и специальными символами.

**Команда:**

|  |
| --- |
| $ cat > File2 << EOF |

EOF – метка конца файла.

**Результат:**

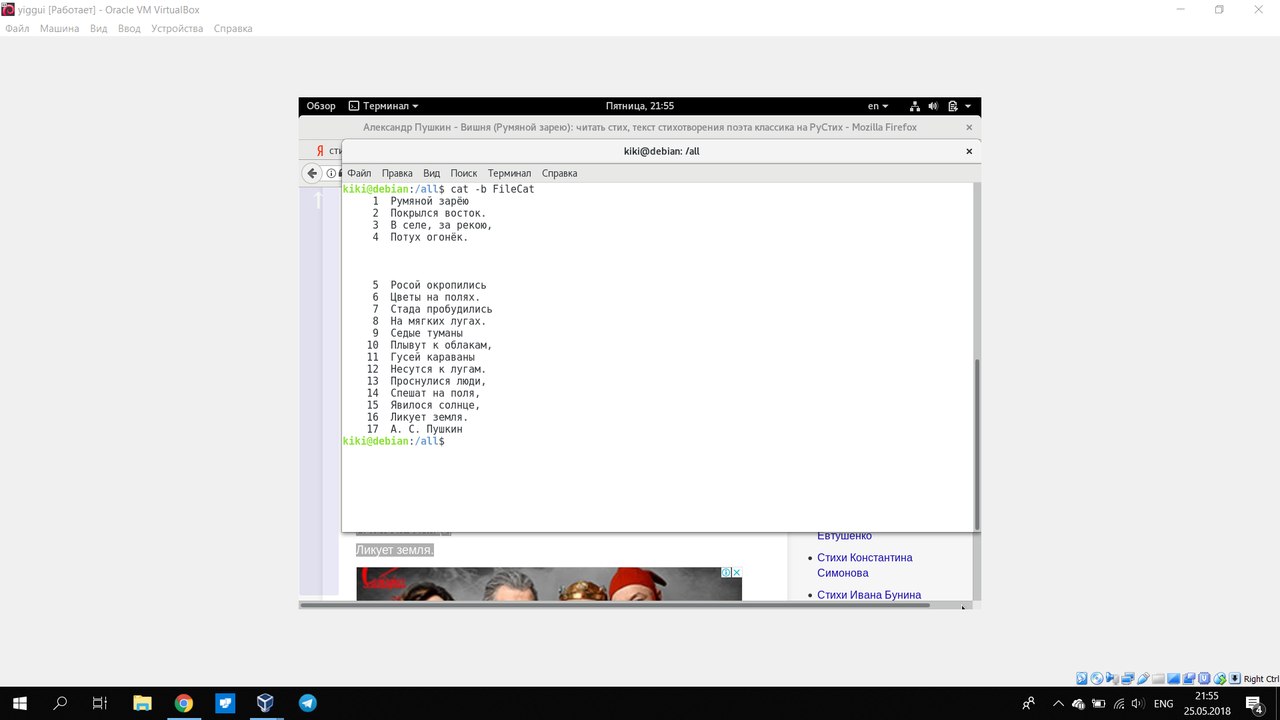


**Задание:** пронумеровать только непустые строки в файле.

**Команда:**

|  |
| --- |
| $ cat –b FileCat |

**Результат:**

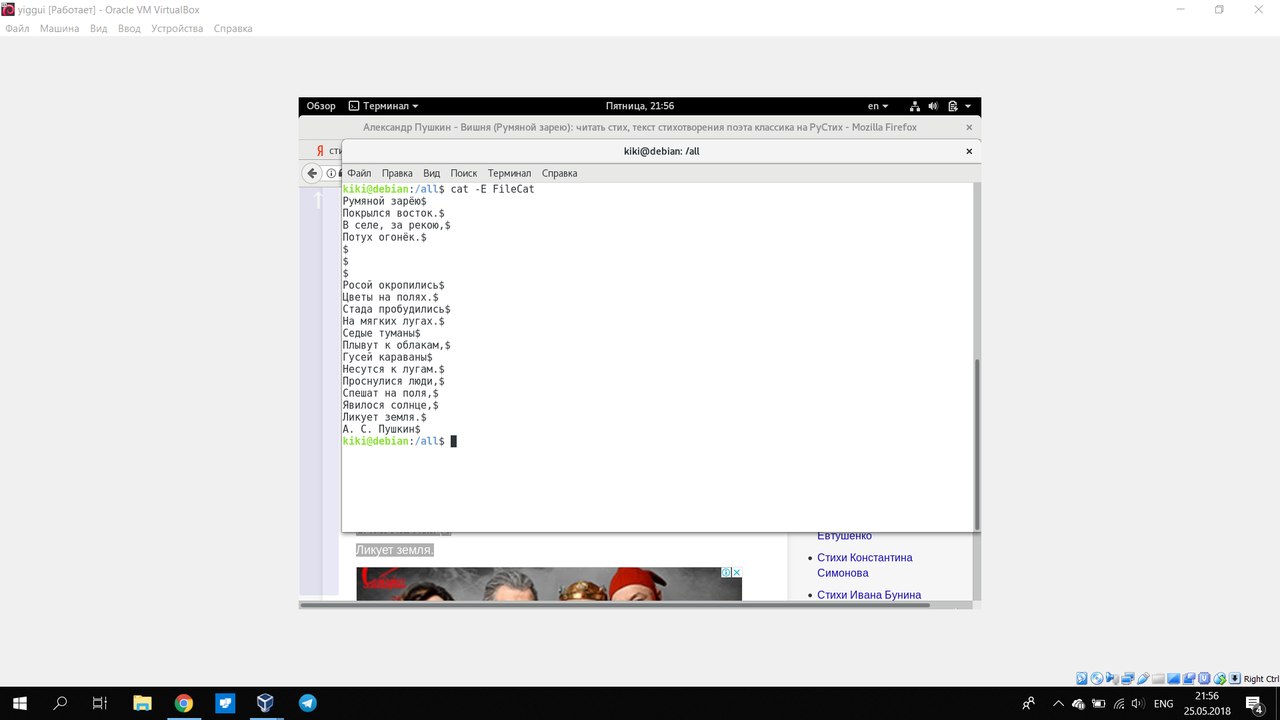


**Задача:** вывести файл со знаком $ в конце каждой строки.

**Команда:**

|  |
| --- |
| $ cat –E FileCat |

**Результат:**

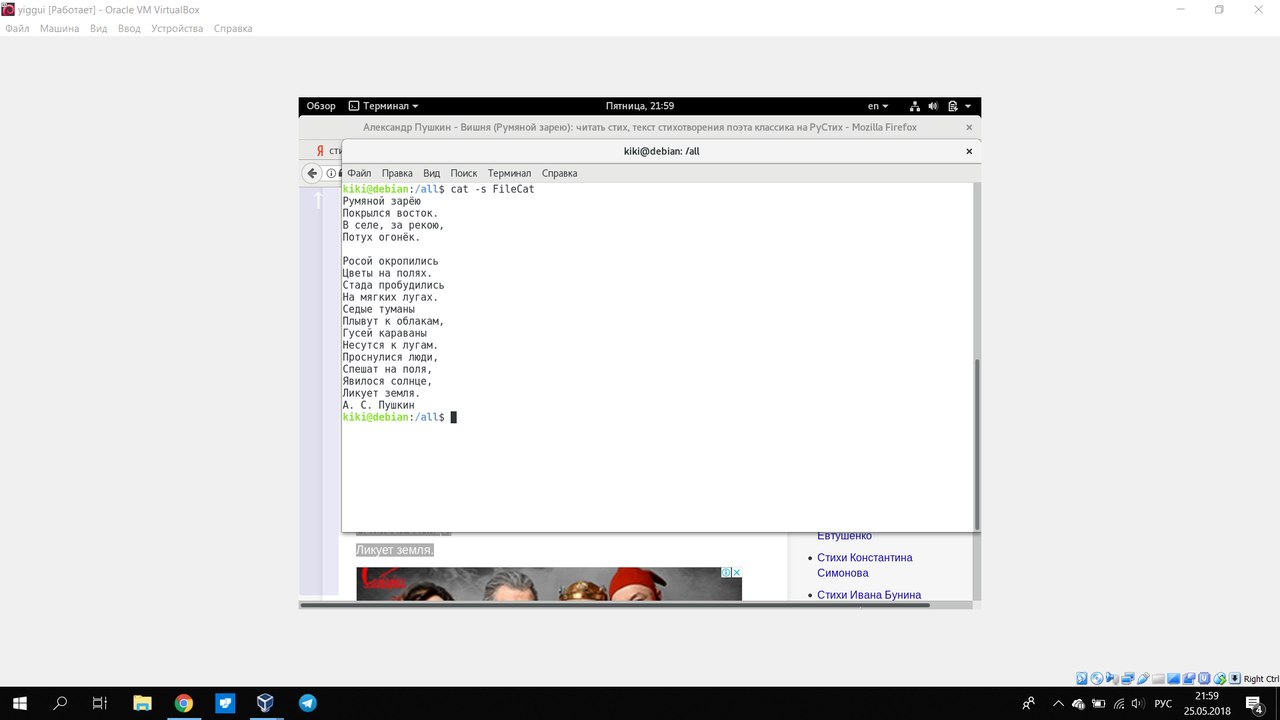


**Задача:** удалить пустые повторяющиеся строки.

**Команда:**

|  |
| --- |
| $ cat –s FileCat |

**Результат:**



**2.6 Команда tee**

Команда tee перенаправляет входной поток в файл и в стандартный вывод, т.е. то, что мы введем запишется в файл, который мы укажем, и выведется на экран.

Ключи:

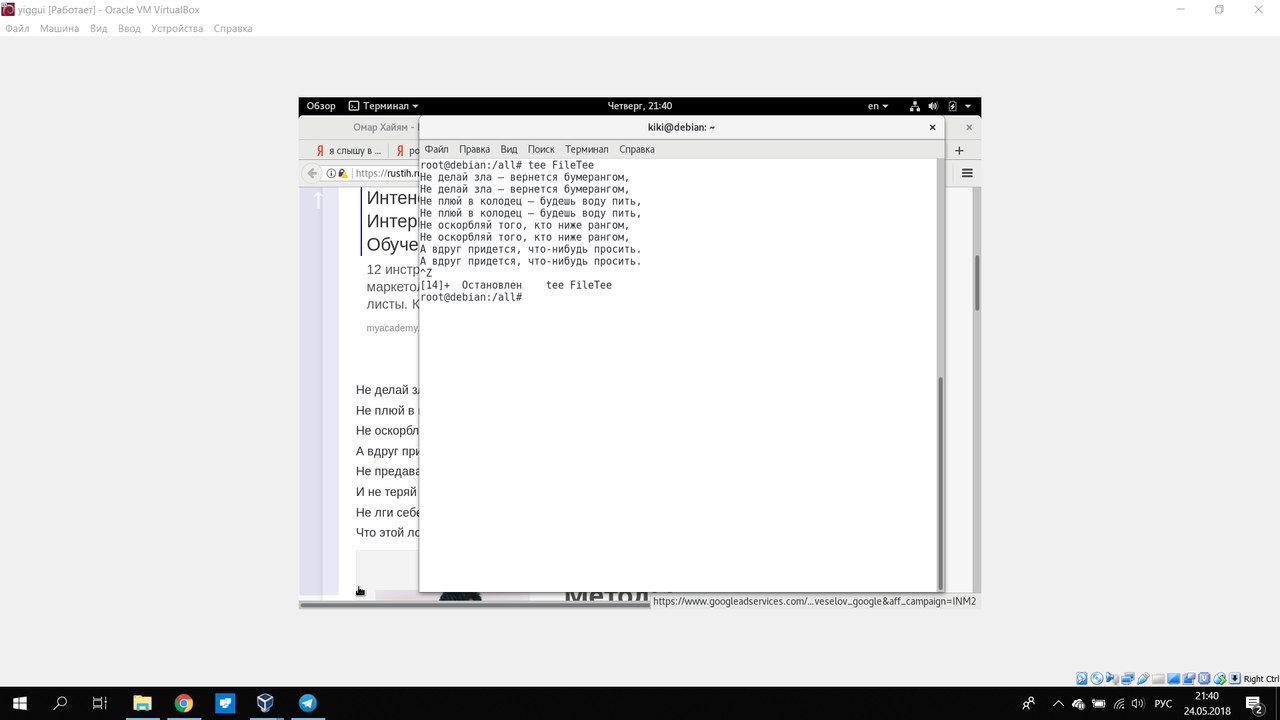
* -а - дописать выходящие данные в конец файла, не переписывая его содержимого.

**Задача:** создать файл.

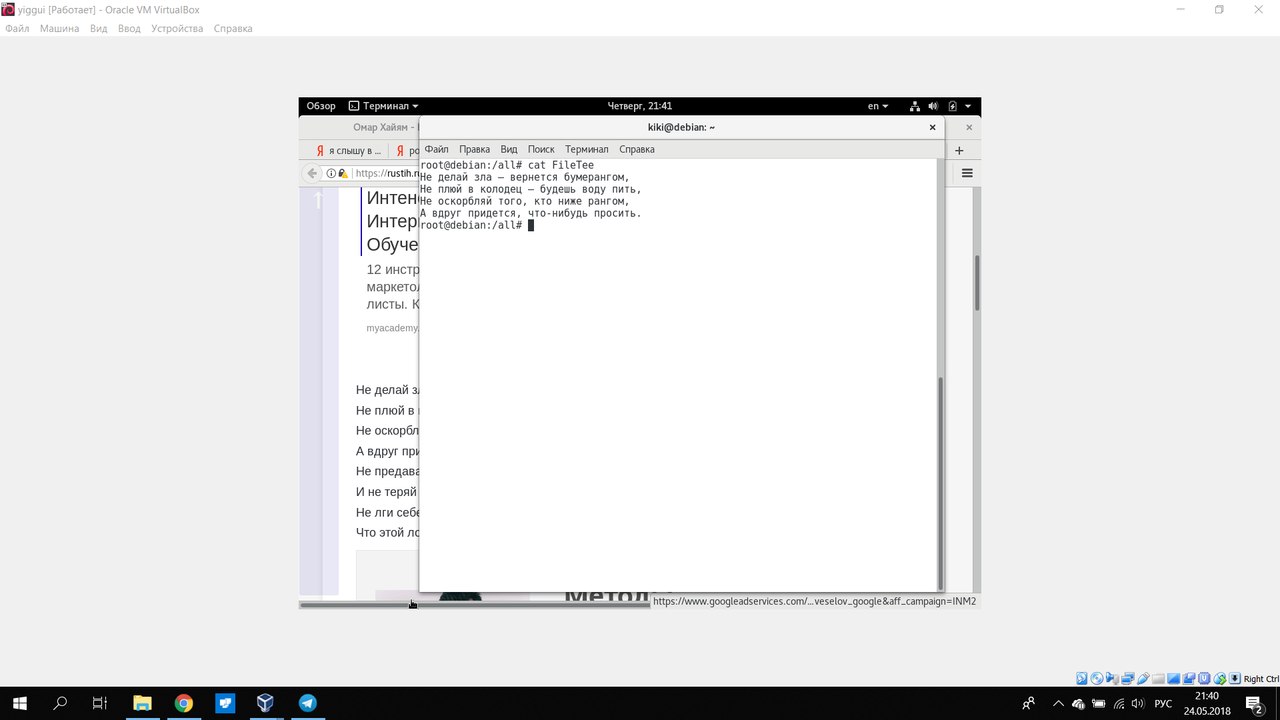
**Команда:**

|  |
| --- |
| $ tee FileTee |

**Результат:**



Посмотрим получившийся файл командой cat:

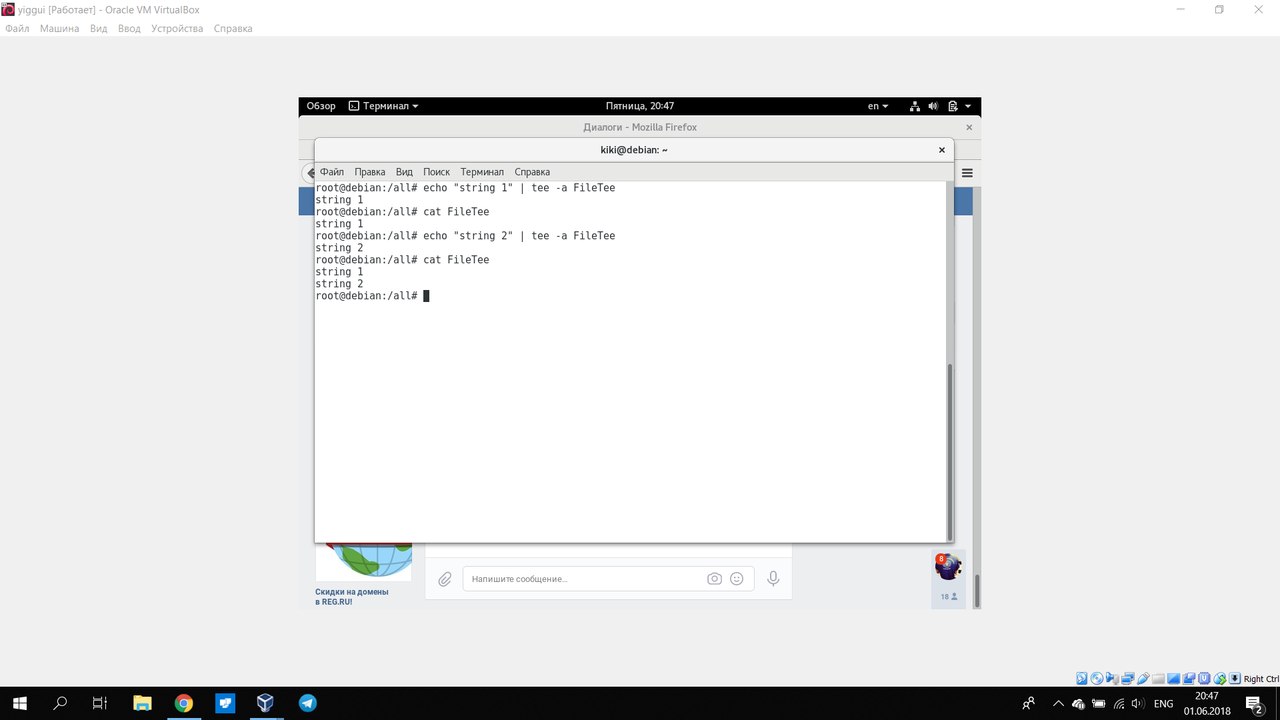


**Задача:** дописать данные в конец файла, не переписывая его.

**Команда:**

|  |
| --- |
| $ tee –a FileTee |

**Результат:**



**Заключение**

В данной курсовой работе мы познакомились с оболочкой bash и консольными командами категории «Print File Contents», научились пользоваться ими на практике.

**Список литературы**

1. [https](https://itshaman.ru/articles/7/komandy-linux" \l "file)://itshaman.ru/articles/7/komandy-linux#file
2. <https://www.opennet.ru/man.shtml?topic=bash&category=1>
3. <https://habr.com/post/47163/>
4. <https://geektimes.com/post/95646/>
5. <https://www.opennet.ru/docs/RUS/linux_base/node159.html>
6. <http://www.linux-ink.ru/static/SL.5.1_Docs/Russification/Docs/sbs-sl-ru/s1-navigating-files-reading.html>
7. http://www.rootfront.com/article/6152481/2013-03-07/13-osnovnyh-primerov-komandy-sat-v-linux