Минобрнауки России

ФГБОУ ВО “Удмуртский государственный университет”

Институт математики информационных технологий и физики

Кафедра информатики и вычислительной техники

Отчёт по учебной стационарной практике

Студента группы ОБ - 09.03.01.01 - 31

Вологжанина Егора Алексеевича

Научный руководитель:

к.ф. - м.н., доцент

Тонков Леонид Евгеньевич

# Оглавление

[Оглавление](#_o8pdd9sou0yt) **1**

[Введение](#_xqv916ns84xp) **2**

[Обзор выбранного дистрибутива](#_2vh0x55eilkw) **3**

[Установка](#_lsl1nw3barvw) **4**

[Результат работы основных команд](#_w21waq8gnmu) **6**

[Тесты, пройденные для проверки знаний о Linux](#_n0ums6dcayc5) **19**

[Создание репозитория на GitHub](#_rkjlapbutzkt) **20**

[Заключение](#_8b0dhpz8pena) **23**

[Список литературы](#_oaibgs8m7c2z) **24**

[Приложение](#_b7c21uwybk3k) **25**

# Введение

* Linux:

Linux - семейство Unix-подобных операционных систем на базе ядра Linux, включающих тот или иной набор утилит и программ проекта GNU, и, возможно, другие компоненты. Как и ядро Linux, системы на его основе, как правило, создаются и распространяются в соответствии с моделью разработки свободного и открытого программного обеспечения. Linux-системы распространяются в основном бесплатно в виде различных дистрибутивов - в форме, готовой для установки и удобной для сопровождения и обновлений, также имеющих свой набор системных и прикладных компонентов, как свободных, так и проприетарных (собственнических).

Появившись как решения вокруг созданного в начале 1990-х годов ядра, уже с начала 2000-х годов системы Linux являются основными для суперкомпьютеров и серверов, расширяется применение их для встраиваемых систем и мобильных устройств, некоторое распространение системы получили и для персональных компьютеров.

* GitHub:

GitHub - это крупнейший веб-сервис для хостинга IT - проектов и их совместной разработки.

Веб-сервис основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc (ранее Logical Awesome). Сервис бесплатен для проектов с открытым исходным кодом и (с 2019 года) небольших частных проектов, предоставляя им все возможности (включая SSL), а для крупных корпоративных проектов предлагаются различные платные тарифные планы.

# Обзор выбранного дистрибутива

В качестве дистрибутива для установки был выбран Xubuntu, основанный на проекте Ubuntu. Выбранный мною дистрибутив использует сравнительно лёгкую графическую среду Xfce и приложения на GTK+3. Цель проекта Xubuntu - возможность использования легковесной альтернативы рабочим средам GNOME и KDE. Xubuntu имеет минималистичный интерфейс пользователя, экономно использующий память и другие аппаратные ресурсы компьютера, что позволяет комфортно работать на старых машинах. Но вместе c тем, данный дистрибутив является современной системой, имеющей огромный набор пакетов.

Основное отличие от базового варианта Ubuntu - среда рабочего стола Xfce. Большинство прикладного ПО также взято из этой среды.

Xubuntu - элегантная и простая в использовании операционная система.

# Установка

Xubuntu устанавливается на компьютер, на котором уже была установлена операционная система Windows 10, в качестве второй операционной системы.

1. Загрузка Xubuntu и запись его на носитель:

Загрузить дистрибутив Linux проще простого: необходимо открыть сайт нужного дистрибутива, отыскать раздел загрузок и выберать то, что подходит по разрядности процессору.

Когда дистрибутив в формате ISO скачан, нужно записать его на CD или обычную USB-флешку. Для записи на флешку понадобятся специальные утилиты. Для Windows лучше выбрать Rufus. У этой программы очень простой интерфейс и в ней довольно сложно запутаться.

1. Подготовка раздела диска:

Понадобится открыть “Средство управления дисками” Windows. Выбрать диск или раздел, от которого планируется отрезать место для установки Linux. Большинству дистрибутивов с лихвой хватит 15 ГБ. Щёлкнуть по разделу правой кнопкой мыши и выбрать “Сжать том”. Ввести размер и нажать ОК.

Когда “Средство управления дисками” закончит изменять размеры разделов, на диске появится пустое неразмеченное пространство, отмеченное чёрным цветом. Туда нужно будет установить Linux.

1. Подготовка загрузчика и загрузка Linux с носителя:

На компьютерах под управлением предустановленной Windows 10 используется загрузчик UEFI, который не позволит загрузить какую бы то ни было систему, кроме Windows. Чтобы это исправить, понадобиться зайти в настройки BIOS компьютера и отключите опцию Secure Boot. После чего перезагрузиться.

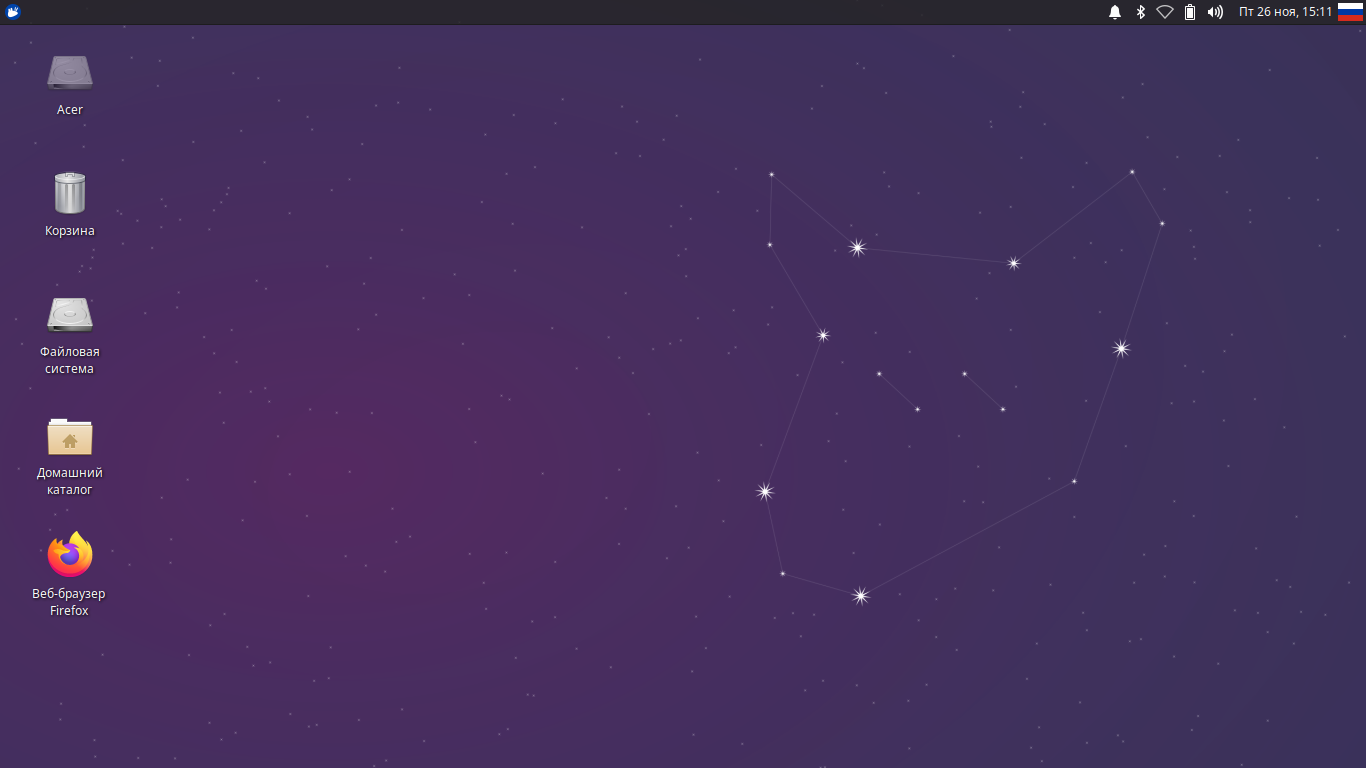
Затем в BIOS необходимо выбрать флешку с Linux в качестве загрузочного носителя.

1. Установка:

Завершающий этап обычная установка, почти ничем не отличающаяся у большинства дистрибутивов, во время которой нужно выполнить ряд действий:

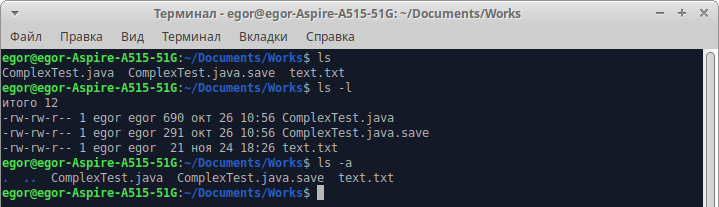
1. Выбрать язык, который будет использоваться в системе;
2. Указать раскладку клавиатуры в дополнение к английской;
3. Отметить пункт “Установить стороннее программное обеспечение” или “Установка мультимедиа кодеков”, чтобы Linux смогла использовать проприетарные драйверы и кодеки;
4. Размещение диска (Автоматически или вручную). Большинство установщиков Linux сразу обнаруживают установленные системы. И даже если пункт ( 2. ) не был выполнен, можно выбрать вариант “Установить рядом с Windows”. Установщик самостоятельно создаст нужные разделы.
5. Завершение установки:

Остаётся лишь указать часовой пояс, выбрать предпочитаемый язык ввода, ввести имя и придумать пароль, после чего начнётся установка.

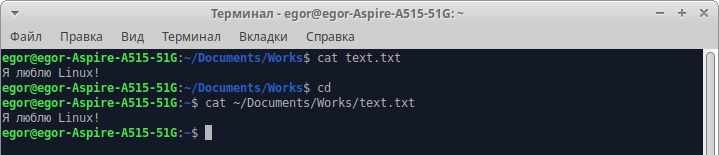


# Результат работы основных команд

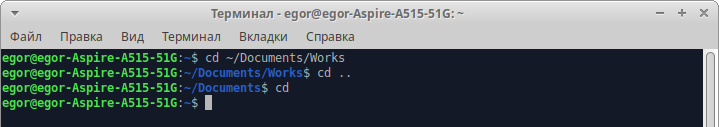
1. Команды для управления файлами:
   1. **ls** - утилита для просмотра содержимого каталогов. По умолчанию показывает текущий каталог. Если в параметрах указать путь, то она перечислит содержимое указанного каталога. Опции “ -l “ (List) и “ -a “ (All): Первая форматирует вывод в виде списка с более подробной информацией, а вторая включает показ скрытых файлов.



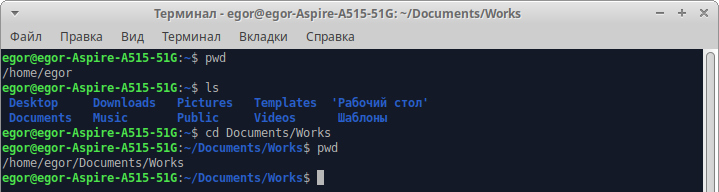
* 1. **cat** - печатает содержимое файла, переданного в параметре, в стандартный вывод. Если передать несколько файлов, команда их склеит. Также можно перенаправить вывод в ещё один файл с помощью символа “ > “.



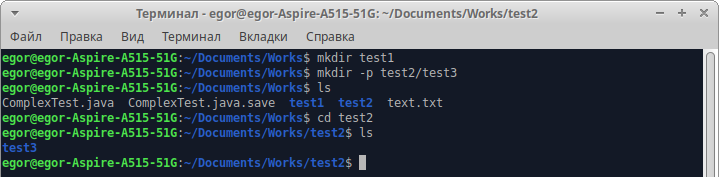
* 1. **cd** - позволяет перейти из текущего каталога в указанный. Если запустить без параметров - возвращает домашний каталог. Вызовы с двумя точками возвращает на уровень вверх, а с тире к предыдущему каталогу.



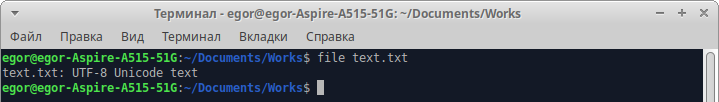
* 1. **pwd** - печатает на экран путь к текущему каталогу.



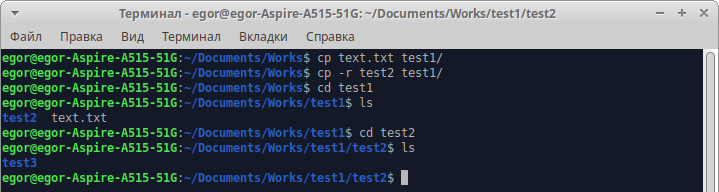
* 1. **mkdir** - создание новых каталогов. Опция -p (Parents), позволяет создать всю структуру подкаталогов.



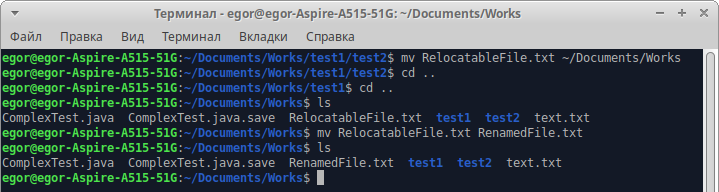
* 1. **file** - показывает тип файла.



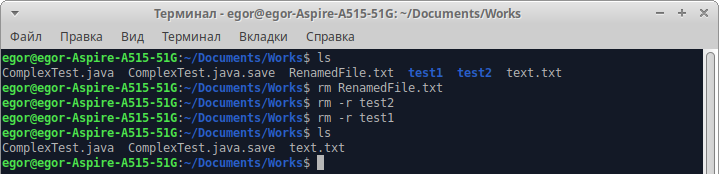
* 1. **cp** - копирование файлов и каталогов. По умолчанию не копирует каталоги рекурсивно, поэтому не стоит забывать об опции “ -r “ (Recursive). Опция “ -a ” включает режим сохранения атрибутов.

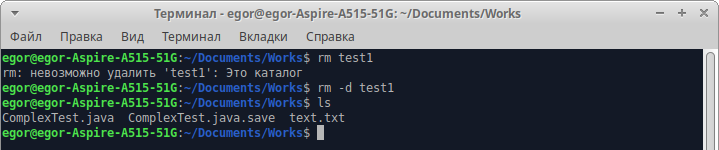
****

* 1. **mv** - перемещение или переименование файлов и каталогов. В Linux переименование - это перемещение файла в ту же папку с другим именем.

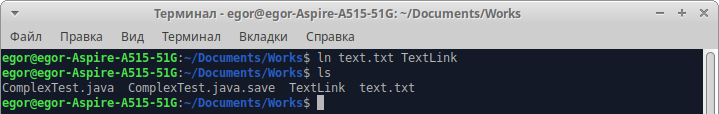


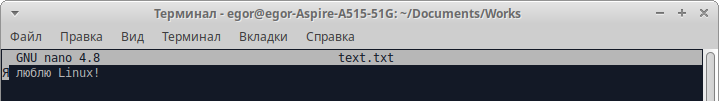
* 1. **rm** - удаляет файлы и папки. Опция “ -r “ позволяет организовать рекурсивное удаление, а опция “ -d ” даёт возможность удалить пустую папку. Действия команды необратимы.

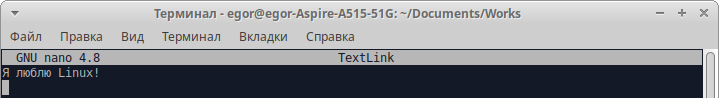




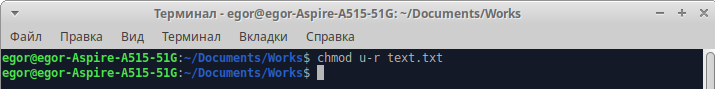
* 1. **ln** - создаёт жёсткие или символические ссылки на файлы. Символические - это что-то похожее на ярлыки в Windows. Жёсткие ссылки указывают на физический адрес области диска, где хранятся данные файлы.

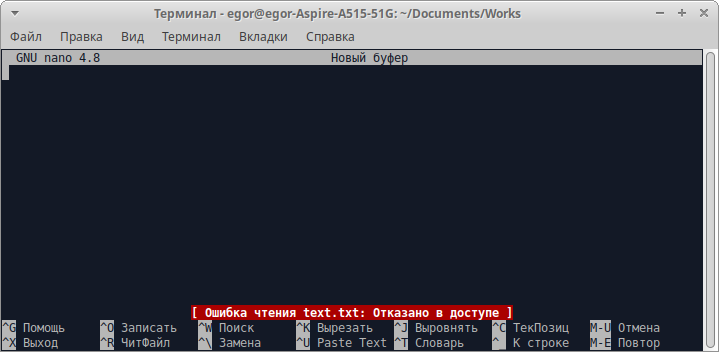




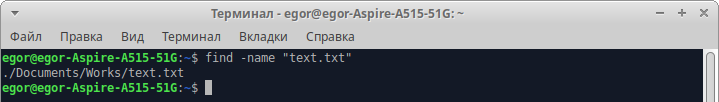


* 1. **chmod** - изменяет права доступа к файлу. Это чтение, запись и выполнение.

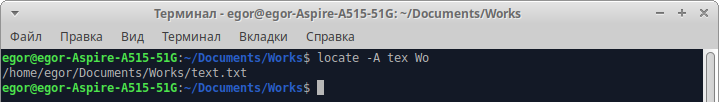




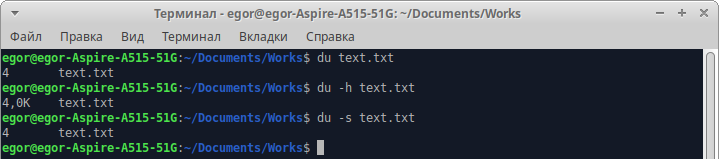
* 1. **find** - поиск в файловой системе, файлах и папках.

****

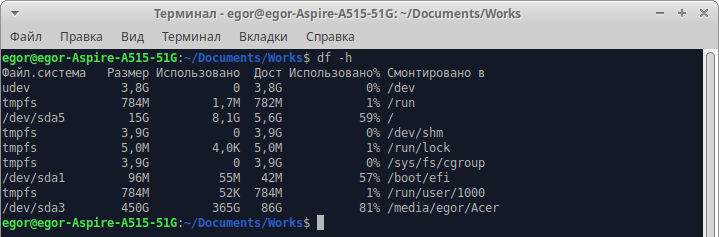
* 1. **locate** - ведёт поиск в базе данных updatedb для шаблонов имён файлов. Эта база данных содержит снимок файловой системы. Опция -A (--all) - выводит только файлы, соответствующие всем заданным шаблонам.



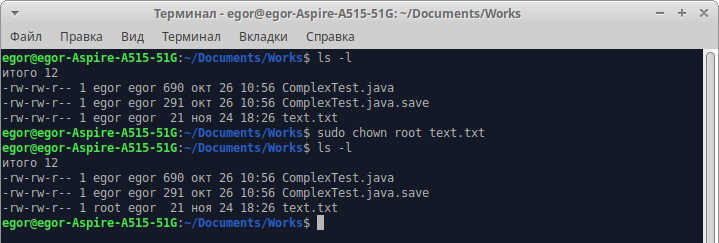
* 1. **du** - показывает размер файла или каталога. Опции “ -h “ (Human) - преобразует размеры в легко читаемый формат, “ -s “ (Summarize) - выводит минимум данных, и “ -d “ (Depth) - устанавливает глубину рекурсии по каталогам.



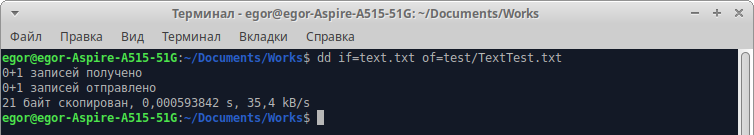
* 1. **df** - анализатор дискового пространства. По умолчанию вывод: перечислены все файловые системы, их размер, количество использованного и свободного пространства. Опция “ -h “ делает размеры легко читаемыми.



* 1. **chown** - изменяет владельца файла. Может делать только суперпользователь. Опция “ -r “ для рекурсивного изменения. В текущем примере пришлось использовать команду sudo о которой будет сказано далее.

****

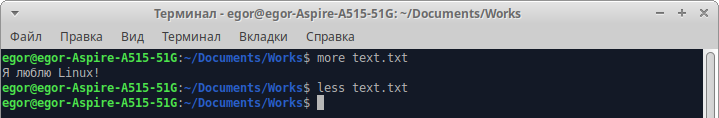
* 1. **dd** - команда для копирования и преобразования файлов. Так же при желании вы можете задать точный размер данных, которые нужно записать или скопировать.

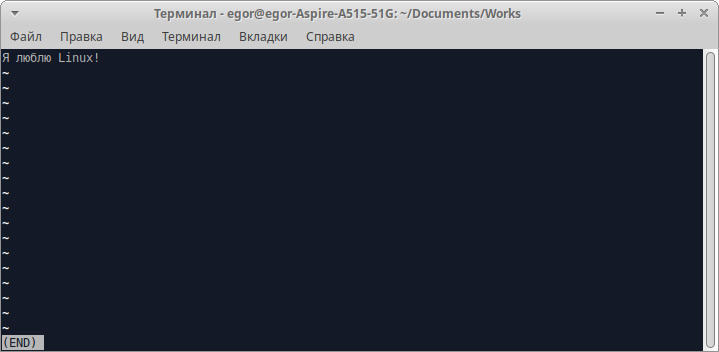


* 1. **mount / umount** - команда для подключения или отключения файловой системы Linux. Можно подключать всё: от USB накопителей до ISO образов. Только у суперпользователя есть права на это.

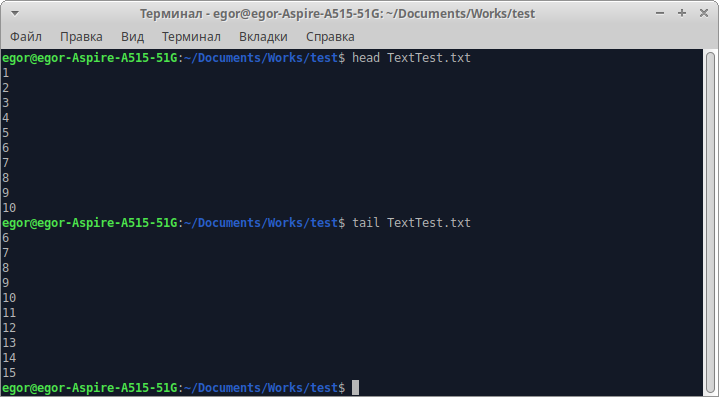
**[ Скриншот ]**

1. Linux команды для работы с текстом:
   1. **more / less** - это команды для просмотра длинных текстов, не вмещающихся на одном экране. Less новее чем more.

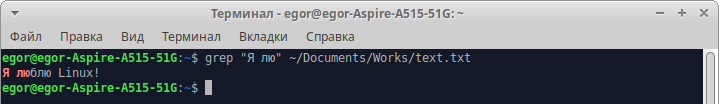




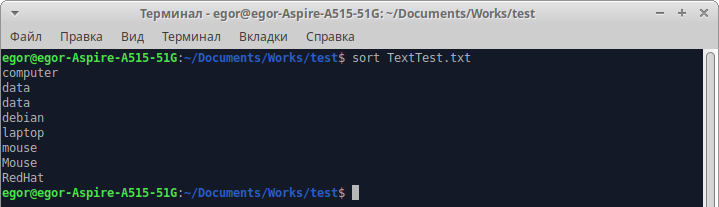
* 1. **head / tail** - команда head выводит несколько первых строк из файла (голова), а tail выдаёт несколько последних строк (хвост). По умолчанию каждая утилита выводит по 10 строк, но опция “ -n “ решает эту проблему. Опция “ -f ” (follow - следовать) - постоянно выводит изменения в файле на экран.

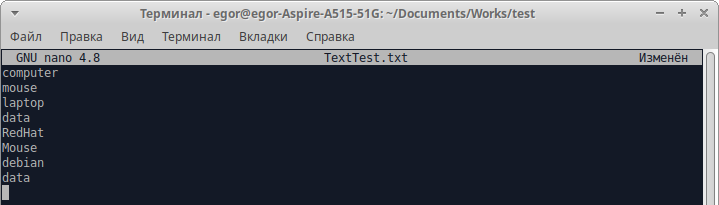


* 1. **grep** - ищет текст по шаблону. По умолчанию принимает стандартный ввод, но можно искать и в файлах. Шаблон может быть строкой или регулярным выражением.

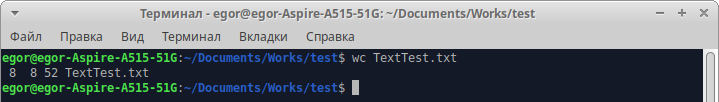


* 1. **sort** - сортировка строк текста по различным критериям. Опции “-n“ (Numeric) по числовому значению, “ -r “ (Reverse) переворачивает вывод.

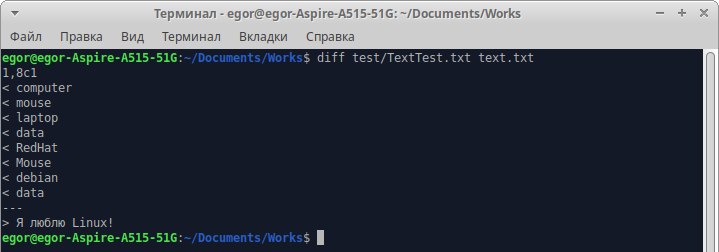




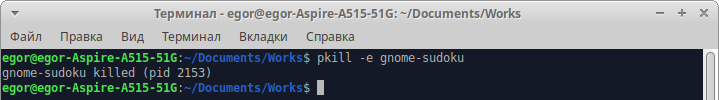
* 1. **wc** - команда для подсчёта количества слов, строк, байт и символов.



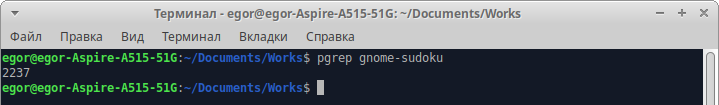
* 1. **diff** - показывает различия между двумя файлами в построчном сравнении. Выводятся лишь строки с отличиями. Изменённые строки помечаются символами “ c “, удалённые - “ d “, а новые “ a “.

****

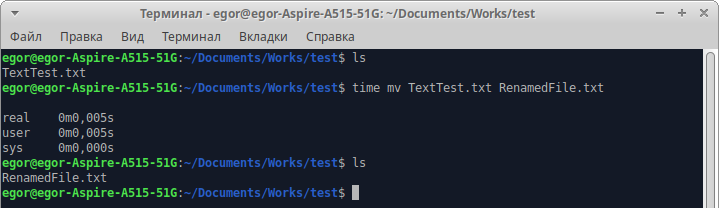
1. Команды для управления процессами:
   1. **kill / xkill / pkill / killall** - служат для завершения процессов. Но они принимают различные параметры для идентификации процессов. Kill нужен PID процесса, xkill - достаточно кликнуть по окну, чтобы закрыть его, killall и pkill принимают имя процесса.



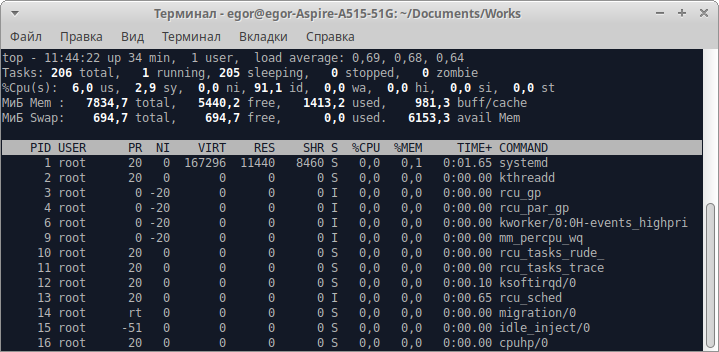
* 1. **ps / pgrep** - команда необходима для получения идентификатора, использующегося в уничтожении или закрытии процесса, она печатает информацию о запущенных процессах. Опция “ -e “ нужна чтобы увидеть информацию об определённом процессе.



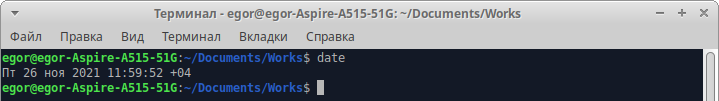
* 1. **time** - время выполнения процесса. Это секундомер для выполнения программы.



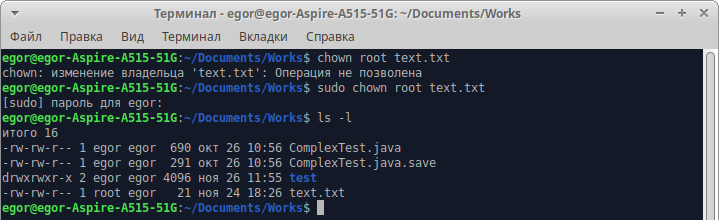
* 1. **top / htop** - обе команды отображают процессы. Htop лучше и новее.



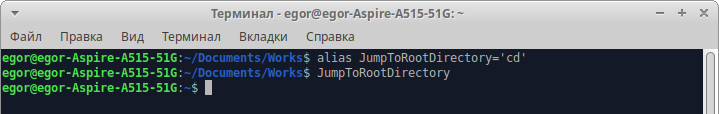
1. Команды окружения пользователя:
   1. **date** - выводит дату и время в стандартный вывод. Имеет гибкую настройку вывода.



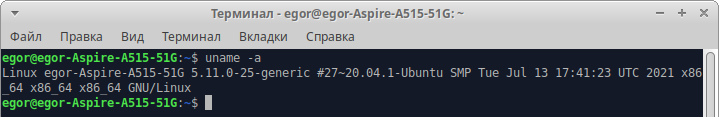
* 1. **su / sudo** - команды для выполнения одной и той же задачи: запустить программу от имени другого пользователя. Разница лишь в том, что su переключает на другого пользователя, а sudo только выполняет команду от его имени.



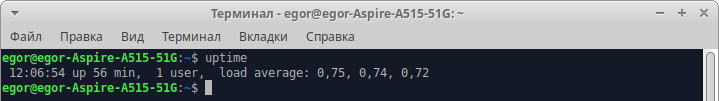
* 1. **alias** - команда создаёт синонимы для других команд Linux. То есть можно делать новые команды или группы команд, а также переименовывать существующие.



* 1. **uname** - выводит некую основную информацию о системе. Без параметров она не покажет ничего полезного, кроме строчки Linux, но, если задать параметр -a (All), можно получить информацию о ядре, имени хоста и узнать архитектуру процессора.

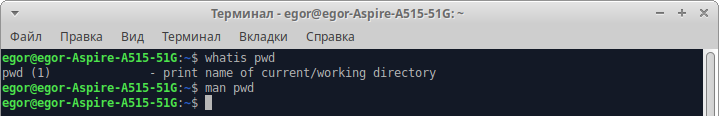


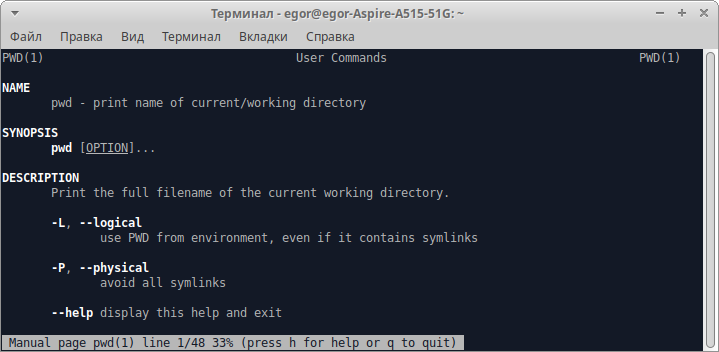
* 1. **uptime** - сообщает время работы системы.



* 1. **sleep** - команда используемая , если вы хотите выключить компьютер через определенный промежуток времени или использовать в качестве импровизированной тревоги.

1. Команды для управления пользователями:
   1. **useradd / userdel / usermod** - позволяют добавлять, удалять и изменять учётные записи пользователей.
   2. **passwd** - позволяет изменить пароль от учётной записи пользователя.
2. Команды для просмотра документации:
   1. **man / whatis** - команда man открывает руководство по определённой команде. Whatis показывает какие разделы руководств есть для данной команды.



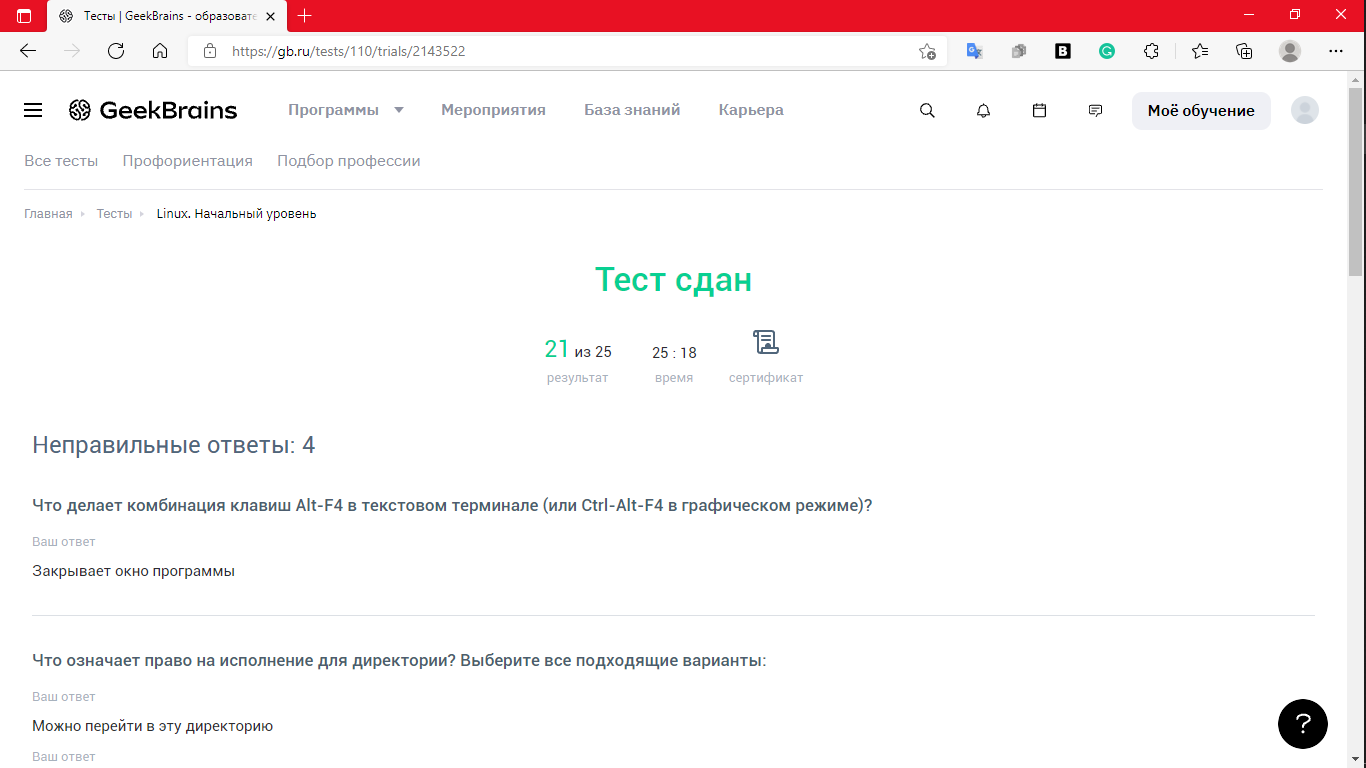


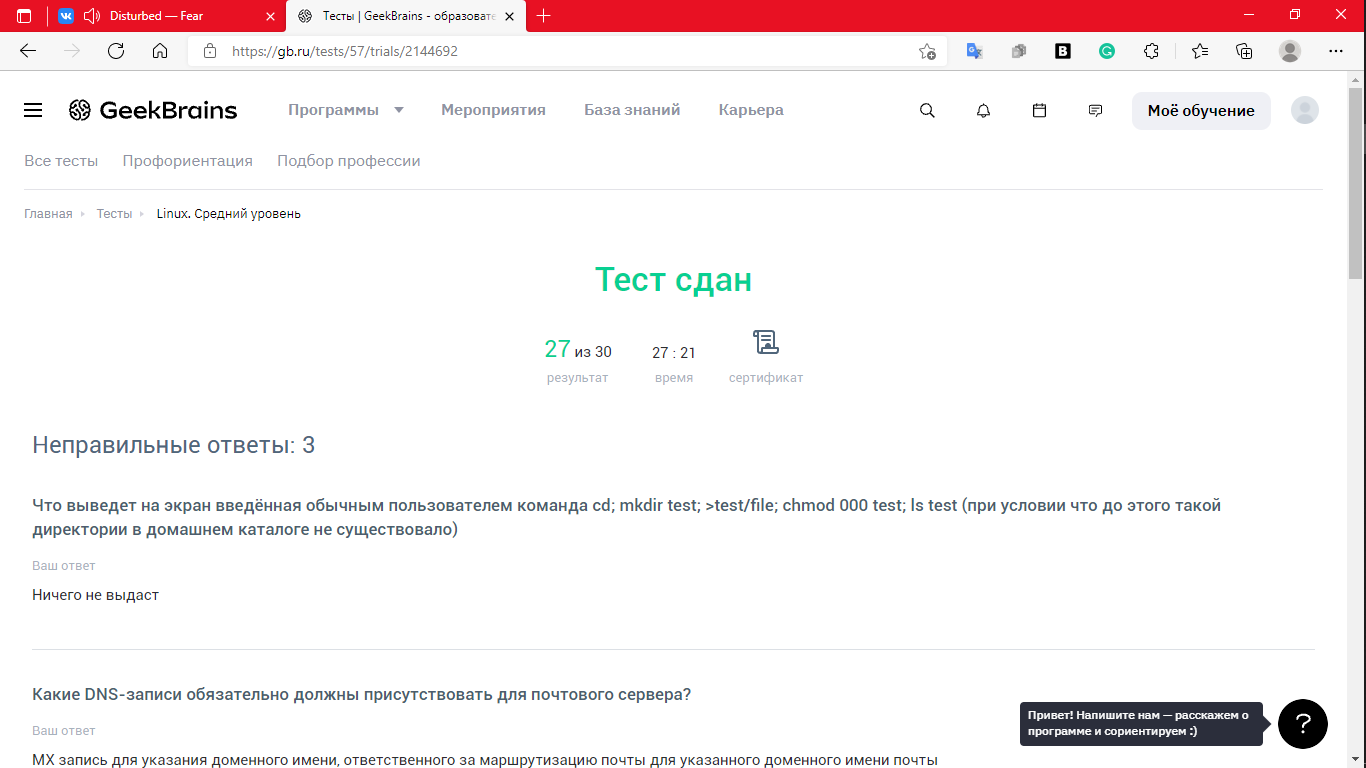
* 1. **whereis** - показывает полный путь к исполняемому файлу программы. Также может показать путь к исходникам, если они есть в системе.

1. Команды для управления сетью:
   1. **ip** - позволяет посмотреть сетевые интерфейсы и IP адреса им присвоенные, посмотреть и настроить таблицу маршрутизации, включать или отключать сетевые интерфейсы, управлять маршрутизацией и ARP, а также многое другое.
   2. **ping** - может быть очень полезным диагностическим инструментом. Она поможет быстро проверить, присутствует ли подключение к маршрутизатору или к интернету, а также дает кое-какое представление о качестве этой связи.
   3. **nethogs** - помогает определить сколько трафика использует какая-либо программа в Linux или какая программа потребляет всю скорость. Для того чтобы задать сетевой интерфейс используется опция “ -i “.
   4. **traceroute** - это усовершенствованная версия команды ping. При исполнении команды можно увидеть не только полный маршрут сетевых пакетов, но и доступность узла, а также время доставки этих пакетов на каждый из узлов.

# Тесты, пройденные для проверки знаний о Linux

Для проверки полученных знаний были пройдены два теста по OC Linux на портале GeekBrains.

Результаты прохождения первого теста:

Результаты прохождения второго теста:

# Создание репозитория на GitHub

* Обзор Git:

Git - это распределённая [система управления версиями](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F%D0%BC%D0%B8), которая помогает отслеживать историю изменений в файлах. Git используют программисты для совместной работы над проектами.

В самом простом виде контроль версий - это сохранение на компьютере серии измененных файлов, например с разными датами в названии, или режим отслеживания исправлений в текстовых документах.

Разработчикам часто бывает нужно вернуться к предыдущей версии кода:

1. если оказывается, что решаемая задача больше не актуальна;
2. если требуется внести исправления в более раннюю версию программы;
3. если ошибка нашлась во время работы над новой задачей.

Если над проектом работает много людей, нужно, чтобы они могли вносить изменения в одни и те же файлы без конфликтов и потерь кода. Все эти задачи удобно решаются с помощью Git.

К базовым возможностям Git относятся:

1. возврат к любой предыдущей версии кода;
2. просмотр истории изменений;
3. параллельная работа над проектом;
4. backup кода.

* Создание репозитория:

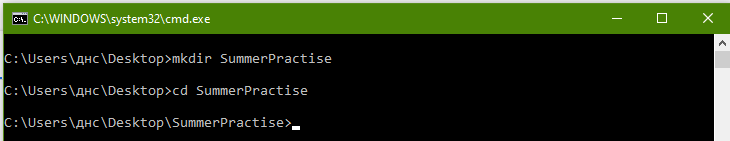
Для создания репозитория используется git Bash, являющееся приложением для сред Microsoft Windows, эмулирующее работу командной строки Git. И хотя на данный момент помимо git Bash существует множество приложений для работы с git, являющееся интуитивно понятными. В данной работе рассматриваться создание репозитория с использованием команд git Bash.

Порядок команд для создания локального репозитория:

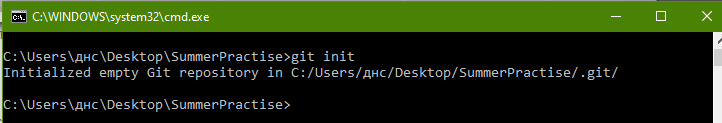
1. Перейти в необходимую папку, в которой будет размещён репозиторий (в моём случае C:\Users\днс\Desktop);



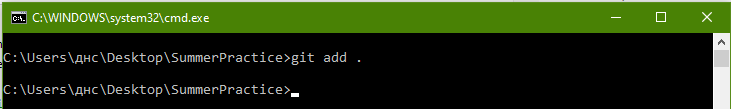
1. Создать папку, которая станет локальным репозиторием и зайти в неё;

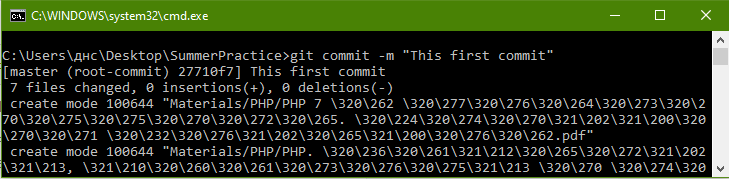


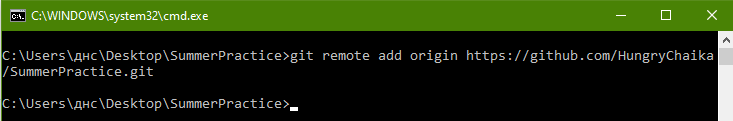
1. Инициализировать репозиторий командой git init, после чего в инициализированной папке появится скрытая папка .git , в которой хранится вся git информация (к примеру commits и конфигурации);



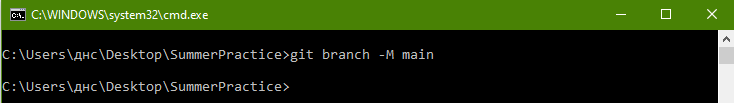
1. Перед тем как создать commit необходимо сделать индексацию файлов с помощью команды git add { путь к файлу или папке }. Символ “ . “ , вместо параметров пути добавляет все существующие папки;



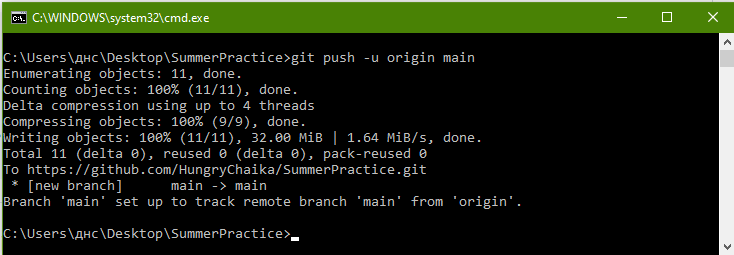
1. Commit создаётся командой commit -m “Название”, где -m это добавление сообщения к commit;
2. Добавление адреса удалённого репозитория в конфигурацию локального командой git remote add origin “Адрес url или ssh”. Для этого необходимо иметь созданный, удалённый репозиторий на GitHub.



1. Переименование главной ветки локального репозитория командой git branch -M main. Было имя master, а стало имя main. Это было необходимо потому, что у удалённого репозитория основная ветка носит имя main;



1. Отправка коммита командой git push -u origin main. Первый коммит нужно отправить с флагом -u, это необходимо для связывания веток локального и удалённо репозиториев;



# Заключение

В результате прохождения учебной практики были приобретены навыки установки и владения операционной системой системы Linux. Были пройдены тесты для проверки полученных знаний об OC Linux. Также был изучен веб-сервис GitHub и система git, посредством создания и выкладывания репозитория.

# Список литературы

1. Скот Чакон, Бен Штрауб. Pro Git - профессиональный контроль версий. 2019 - 608 с.
2. Сайт «Losst»/42 команды Linux которые вы должны знать/

<https://losst.ru/42-komandy-linux-kotorye-vy-dolzhny-znat>

1. Сайт “Лайфхакер”/ Как установить Linux: подробнейшая инструкция/

<https://lifehacker.ru/kak-ustanovit-linux/>

1. Шоттс У. - Командная строка Linux. Полное руководство. - СПб.: Питер, 2017. - 480 с.: ил. - (Серия “Для профессионалов”).
2. Колисничеко Д. Н. Linux. От новичка к профессионалу - 6-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2018 - 672 с.: ил. - (В подлиннике)

# Приложение

1. Ссылка на сертификат о прохождении первого теста:

[Сертификат Linux. Начальный уровень. | GeekBrains (gb.ru)](https://gb.ru/certificates/1413956)

1. Ссылка на сертификат о прохождении второго теста:

[Сертификат Linux. Средний уровень. | GeekBrains (gb.ru)](https://gb.ru/certificates/1416478)

1. Ссылка на репозиторий GitHub:

<https://github.com/HungryChaika/SummerPractice>