|  |  |  |
| --- | --- | --- |
| Название команды | Значение команды | |
| pwd | Показывает в какой директории я сейчас нахожусь. | |
| ls -la | Показывает содержимое папки. | |
| mkdir название папки | Создать папку | |
| cd ..  cd ssh/ | Сменить папку | |
| touch (название и расширение файла) | Создать файл | |
| mv название файла а затем куда хоти его переместить полностью | Переместить или переименовать | |
| vim (Insert (I) начать печатать, закончить нажми ESC)  :wq сохранить и выйти | Вызвать текстовый редактор | |
| find . -name "DM.txt" | Найти файл | |
| Начало работы с ГитХабом | | |
| git config --global user.name "My Name"  git config --global user.email myEmail@example.com | 1 | |
| git clone https://github.com/tutorialzine/awesome-project.git | 2 | |
|  | | |
| cp имя файла / полный адрес куда копируем | | копировать файл по месту назначения |
| tail  **-c** - выводить указанное количество байт с конца файла;  **-f** - обновлять информацию по мере появления новых строк в файле;  **-n** - выводить указанное количество строк из конца файла;  **--pid** - используется с опцией -f, позволяет завершить работу утилиты, когда завершится указанный процесс;  **-q** - не выводить имена файлов;  **--retry** - повторять попытки открыть файл, если он недоступен;  **-v** - выводить подробную информацию | | вывести заданное количество строк c конца |
| head  остальные команды the same as under! | | вывести заданное количество строк с начала |
| less имя файла  **-a**, **--search-skip-screen** — не осуществлять поиск в тексте, который в данный момент отображен на экране;  **-bn**, **--buffers=n** — задать размер буфера памяти;  **-c**, **--clear-screen** — листать текст, полностью стирая содержимое экрана (построчная прокрутка работать не будет);  **-Dxcolor**, **--color=xcolor** — задать цвет отображаемого текста;  **-E**, **--QUIT-AT-EOF** — выйти, когда утилита достигнет конца файла;  **-e**, **--quit-at-eof** — выйти, когда утилита второй раз достигнет конца файла;  **-F**, **--quit-if-one-screen** — выйти, если содержимое файла помещается на одном экране;  **-f**, **--force** — открыть специальный файл;  **-hn**, **--max-back-scroll=n** — задать максимальное количество строк для прокрутки назад;  **-yn**, **--max-forw-scroll=n** — задать максимальное количество строк для прокрутки вперёд;  **-i**, **--ignore-case** — игнорировать регистр;  **-I**, **--IGNORE-CASE** — игнорировать регистр, даже если паттерн для поиска содержит заглавные буквы;  **-jn**, **--jump-target=n** — указать, в какой строке должна быть выведена искомая информация;  **-J, --status-column** — пометить строки, соответствующие результатам поиска;  **-n**, **--line-numbers** — не выводить номера строк;  -**N**, **--LINE-NUMBERS** — вывести номера строк;  **-s**, **--squeeze-blank-lines** — заменить множество идущих подряд пустых строк одной пустой строкой;  **-w**, **--hilite-unread** — выделить первую строку нового фрагмента текста. | | просмотреть содержимое файла (длинного) |
| date | | вывести дату и время |
| ssh-keygen -t rsa -C "Maksimkoshko" | | создать ключи шифрования ssh |
| git config --global user.name "Maksim Koshko"  git config --global user.email [Metalgerray@gmail.com](mailto:Metalgerray@gmail.com)  git config --globa user.email | |  |
| rm удалить файл | |  |
| git add . или git add file name | | добавить в репозиторий гита на сайте |
| git status | | проверить статус файлов в папке |
| git add \* | | добавить файл в репозиторий, все файлы |
| git commit -m "add the first file" | | предыдущая версия файла |
| pwd;ls –la etc. | | скрипт на выполнение нескольких действий по порядку |
| ./ название скрипта | | выполнить скрипт |
| #!/bin/bash | | пишем в виме для того чтобы исполнить скрипт на языке баша |
| git branch | | проверить в какой ветке мы находимся. Должна быть звёздочка! |
| git checkout –b first\_branch | | создать ветку (b-branch) |
| git commit –m “название файла” | | дать название изменениям. |
| git add . ; git add ‘название файла’ | | добавить изменеия |
| git branch | | посмотреть наличие веток |
| git checkout master | | перейти в ветку в мастер ветку |
| git commit –a –m “” | | добавить изменение в файле чтобы это увидел гитхаб (часть с –а) (a –add) (m- message название комита) |
| git push –u origin first\_branch | | пуш бранчи на удаленку если ее там ранее не было (u- upstream загрузить в сервер гита) |
| git log | | вывести список коммитов |
| Git checkout в ветку в которую хочешь смержить  Git merge название ветки которую мержишь в мейн ветку | |  |
| git push | | запушить обычную мейн ветку а не новую то есть запушить уже существующую |
| git pull | | синхронизировать ветку между локальным и удаленным репозиторием |
| git gui | | запустить гит гуи |
| git gui& | | запустить гит гуи совместно с гит башем |
| gitk& | | запустить просмотр истории |