Git- это распределённая система контроля версий.

Clear- очистить терминал

Pwd- показать в какой папке сейчас нахожусь

Ls -la- содержимое папки / ls -ra содержимое других папок в папке

Mkdir namefolder создать папку Что бы создать сразу несколько папок- mkdir dir\_1 dir\_2 dir\_3 Mkdir(make directory) mkdir {00...10} создать 11 папок

Tail -F test.txt содержимое файла в реальном времени

Cd “C:\Users\Tim\Downloads\docx\_info” зайти в папку / cd .. -вернуться обратно на 1 папку

Touch test.txt создать файл или > test.txt создать 1 файл Что бы создать несколько файлов touch test1.txt test2.txt

Vim test.txt открыть файл Что бы его редактировать нужно нажать на клаве insert. Если нужно выйти- жмём Esc, далее : и пишем wq.

Start . открыть дерикторию где нахожусь в проводнике или удержимая ctrl кликнуть по пути в git bash

Mv test.txt mv \*.doc

Mv -r (рекурсия, всё что внутри тоже перенесётся)

Rm -r название папки – удалить папку и всё что в ней

Cp- копирование

Cat 2.txt- показать текстовое содержимое файла

Cat 2.txt 1.txt 3.txt -если нужно сразу несколькЕо посмотреть

Cat file5.txt file6.txt file7.txt > file10.txt объеденить текст в несколько файлах в один файл

Less- открыть файл в bash. Если в нём нужно что то найти, к примеру слово One, то пишем /One. Q- что бы выйти.

Nano 1.txt- более мощный редактор. Ctrl + что бы выполнять команды внутри ctrl

Grep имя файла в котором ищем- регистро зависимо

Grep -I то что ищем в чём ищем- убрать регистро зависимость

Grep -v one 1.txt- все строки в которых нет one

Grep -c one 1.txt или -ci – посчитать количество всех строк в которых содержится наш целевой запрос

Grep -R one folder\_1- посмотреть внутри папки, где в файлах есть one

Find .(точка показывает что мы ищем внутри этой директории -name показывает что ищем в соответсвие этого имени

Find . -name 1.txt

Find . type -d -name folder\_1 искать директорию

Find . -name \*.png- искать все файлы типа .png

Echo Hello- вывести инфу в консоль

Echo Hello > 33.txt -записать текст в файл

Echo Hello >> 33.txt >>- Добавить ещё текст не заменяя всё.

Cat bash.txt | grep -I Test – использовать 2 команды сразу

Ps – показывает процессы в консоле

Ps u процессы заасайненные на 1 юзера

Ps au – все процессы для всех юзеров

Ps aux -вообще все процессы вне терминала

Kill id процесса- закрыть процесс

Ping -c 3 (количество пакетов которое нужно пингануть)

Ping -c 3 -L 4 (-L 4)- каждые 4 секунды будет правлятся пакет

Curl -L google.com

Curl -L google.com –verbose -доп инфа с сервера

Ls –help- подсказка для команды

Git

Git clone link repo- клонировать репозиторий

Git init

Git status- инфа

Git add . или git add 1.txt

Git commit -m “название коммиита”

Git push- закинуть коммит на репозиторий

Git commit -am “название коммита” -добавить изменения и коммит. Только для тех которые уже были закомичены раньше

Ssh-

Git config

Ветки

Git fetch показать если изменения на удалённом репозитории

git pull обновить ветку до текущего состояния(залить на пк то чего нет) Синхронизировать локальный и удалённый репозеторий

git log- информация о всех коммитах которые происходили. Q- закрыть

git log --author “имя автора” посмотреть какие изменения делал автор

git show “id коммита” – информация, которые изменения проводились в коммите

git blame file1.txt- изминения в строках какие проводились

git diff – изменения которые проходят в файле

git merge --abort отменить слияние

git branch first\_branch- создать ветку(всё что было в мейне- копируется)Ветка создаётся на последнем коммите

git checkout firts\_branch – переключится на эту ветку

git push -u origin first\_branch – связать ветки

git checkout - вернуться на предведущую ветку

git merge first\_branch- слияние 2 веток

git branch -m “старое название” “новое название” – переименовать ветку