

Семинар #0: Основы командной строки и git

Основные команды терминала Linux

pwd	напечатать имя текущей директории
ls	напечатать все файлы и папки текущей директории
ls -l	больше информации
ls -a	показать скрытые файлы
cd имя_папки	перейти в соответствующую папку
	например: cd /home-local/student
cd ..	перейти в папку, содержащую данную
mkdir имя_новой_папки	создать новую папку
touch имя_нового_файла	создать новый пустой файл
cp имя_файла новое_расположение	копировать файл
mv имя_файла новое_расположение	переместить файл
mv имя_файла новое_имя_файла	переименовать файл
rm имя_файла	удалить файл
rm -r имя_папки	удалить папку

Сокращение для имён папок

.	текущая директория
..	родительская директория
~	домашняя директория (на linux это /home/<имя>, на windows C:\Users\<имя>)

Горячие клавиши

Tab	автозаполнение
2 раза Tab	показать возможные варианты
стрелка вверх	перейти к предыдущей команде
Ctrl-C	выход из программы, например той, которая зависла в бесконечном цикле
Ctrl-D	конец ввода
Ctrl-R	поиск по всем предыдущим командам

Компиляция программы на языке C

gcc имя_файла_исходного_кода	скомпилировать программу и создать исполняемый файл a.out
./a.out	запустить файл a.out в текущей директории



Запуск скрипта Python

python имя_скрипта	запустить скрипт
--------------------	------------------

Работа с текстом

nano имя_файла	открыть файл в редакторе nano, если файла не существует, то будет создан новый Ctrl-O - для сохранения изменений Ctrl-X - для выхода
----------------	--

Работа с ssh

ssh-keygen	создаёт публичный и приватный ssh-ключи в папке ~/.ssh во время запуска попросит ввести passphrase, можно ничего не вводить публичный ключ будет иметь расширение .pub, этот ключ можно передавать другим приватный ключ будет без расширения, его нельзя показывать никому после создания, вам нужно скопировать публичный ключ в настройки gitlab или github, тогда вы сможете делать команды git push и git pull
------------	--

Основы git

Создание репозитория

git init	создать новый пустой репозиторий в данной папке
git clone путь_до_репозитория	клонировать репозиторий

Настройки

git config --global user.name "ваше имя"	настройка имени
git config --global user.email "ваш адрес почты"	настройка почты
git config --global core.editor "nano"	установка текстового редактора nano редактором по умолчанию для git

Добавление изменений в локальный репозиторий

git add имена_файлов	добавить файлы в индекс
git add -all	добавить все изменения (новые, изменённые, удалённые файлы) в индекс
git commit	добавить все изменения, добавленные в индекс с момента прошлого коммита, в локальный репозиторий, после этого откроется текстовый редактор куда нужно будет написать сообщение о коммите и закрыть его.
git commit -m "сообщение"	то же, но без открытия редактора

Просмотр информации о добавленных изменениях

git status	показывает произведённые изменения в рабочей папке и в индексе изменения сделанные в папке, но не добавленные в индекс пишутся красным изменения добавленные в индекс, но не в лок. репозиторий пишутся зелёным
git log	показывают информацию о произведённых коммитах
git log --oneline	сокращённая запись
git log --graph --oneline --all	рисует весь граф коммитов

Добавление изменений из локального репозитория в удалённый

git remote -v	просмотр списка удалённых репозиторий
git remote add имя_уд.репозитория адрес	добавить новый удалённый сервер 1) имя удалённого сервера можно задать любым, но обычно используется имя origin. 2) если вы работаете на gitlab или github то адрес можно найти там кнопка Code на этих сайтах, лучше использовать протокол SSH.
git push имя_уд._сервера	отправка изменений из локального репозитория в удалённый (текущая ветка)
git pull имя_уд._сервера	получение изменений из удалённого репозитория (все ветки) + слияние (текущая ветка)