

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	подробная информация
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
cp -r имя_папки новое_расположение	копировать папку и всё её содержимое
mv имя новое_расположение	переместить файл или папку
mv имя новое_имя	переименовать файл или папку
rm имя_файла	удалить файл
rm -r имя_папки	удалить папку и всё её содержимое
cat имя_файла	просмотреть содержимое текстового файла

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-L	очистить экран
Ctrl-C	завершить текущую программу
Ctrl-D	конец ввода
Ctrl-Z	"свернуть" текущую программу, fg чтобы развернуть jobs чтобы посмотреть все "свёрнутые" программы
Ctrl-R	поиск по всем предыдущим командам
Ctrl-A	переместиться в начало строки ввода
Ctrl-E	переместиться в конец строки ввода
Alt-←	переместиться на одно слово влево
Alt-→	переместиться на одно слово вправо
Ctrl-W	удалить одно слово слева от курсора
Alt-Backspace	удалить одно слово или до любого спец символа слева от курсора
Alt-D	удалить одно слово или до любого спец символа справа от курсора
Ctrl-U	удалить с начала до текущего положения

Справка

man ls	системная справка о командах, например о команде ls
curl cheat.sh/ls	примеры использование команды, например команды ls

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	
Ctrl-K - удалить одну строку	

Компиляция программы на языке С

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 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

<code>git status</code>	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
<code>git log</code> <code>git log -oneline</code> <code>git log -graph -oneline -all</code>	показывают информацию о произведённых коммитах сокращённая запись рисует весь график коммитов

Добавление изменений из локального репозитория в удалённый

<code>git remote -v</code> <code>git remote add</code> имя_уд.репозитория адрес	просмотр списка удалённых репозиториев 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.
<code>git push имя_уд._сервера</code> <code>git pull имя_уд._сервера</code>	отправка изменений из локального репозитория в удалённый (текущая ветка) получение изменений из удалённого репозитория (все ветки) + слияние (текущая ветка)