

# Семинар #0: Основы терминала Linux и основы git

## Задание на работу с терминалом

1. Откройте терминал и узнайте в какой папке вы находитесь. Для этого напечатайте `pwd` и нажмите `Enter`.
2. Перейдите в папку `/home-local/student`. Для этого введите команду:

```
cd /home-local/student
```

3. С помощью команды `pwd` проверьте, что вы действительно находитесь в нужной папке.
4. С помощью команды `ls` просмотрите всё содержимое папки `/home-local/student`. Для этого введите `ls` и нажмите `Enter`.
5. Создайте вашу папку. Используйте команду:

```
mkdir имя_папки
```

За место `имя_папки` подставьте название вашей папки. Желательно, чтобы название содержало только латинские символы или цифры без пробелов.

6. С помощью команды `ls` убедитесь, что ваша папка создалась.
7. Перейдите в вашу созданную папку командой `cd имя_папки`.
8. Перейдите в эту папку с помощью файлового менеджера(проводника) и создайте там файл `hello.c`. Файл обязан оканчиваться на `.c`.
9. С помощью обычного текстового редактора (например, gedit или Sublime Text) напишите в файле `hello.c` текст программы *HelloWorld*:

---

```
#include <stdio.h>
int main()
{
    printf("Hello World\n");
}
```

---

10. В терминале проверьте, что этот файл существует, используя команду `ls`.
11. Скомпилируйте этот файл следующей командой:

```
gcc hello.c
```

После этого в папке создастся новый файл по имени `a.out`.

12. Запустите исполняемый файл `a.out` напечатав полный путь до этого файла:

```
/home/имя_пользователя/имя_папки/a.out
```

13. Точка в имени файлового пути является сокращением для текущей папки. То есть в данном случае точка является сокращением для `/home-local/student/ваша_папка`. Поэтому команду для запуска файла `a.out` можно сократить до:

```
./a.out
```

14. Можно объединить команды компиляции и запуска:

```
gcc hello.c && ./a.out
```

## Задание на работу с git

1. Создайте новый git репозиторий.
2. Создайте два новых файла `cat.txt` и `dog.txt`. Содержимое файлов можете задать сами.
3. Используйте `git status`, чтобы посмотреть изменения.
4. Добавьте эти файлы в индекс.
5. Используйте `git status`, чтобы посмотреть изменения.
6. Добавьте файлы из индекса в локальный репозиторий.
7. Создайте новый файл `mouse.txt` и измените уже существующий файл `cat.txt`.
8. Используйте `git status`, чтобы посмотреть изменения.
9. Добавьте изменения в индекс.
10. Используйте `git status`, чтобы посмотреть изменения.
11. Добавьте изменения из индекса в локальный репозиторий.
12. Удалите файл `dog.txt` и добавьте это изменение сначала в индекс, а потом в локальный репозиторий.
13. Сделайте ещё одно любое изменение и добавьте это изменение сначала в индекс, а потом в локальный репозиторий.
14. Используйте команду `git log` чтобы посмотреть сделанные изменения.
15. Используйте команду `git switch`, чтобы перейти на второй коммит.
16. Используйте команду `git switch`, чтобы вернуться обратно на самый последний коммит.
17. Создайте свой новый пустой репозиторий на gitlab.
18. Свяжите свой локальный репозиторий с удалённым с помощью `git remote add`.
19. Добавьте изменения из локального репозитория в репозиторий на gitlab с помощью `git push` (или `git push -f`).
20. Сделайте ещё одно изменение и добавьте его сначала в индекс, затем локальный репозиторий, а затем в удалённый репозиторий.

## Задание совместную работу с репозиторием

Сделайте задание по адресу `mipt-hsse.gitlab.yandexcloud.net/pro100savant/hellow_word_2025`.