

Лабораторная работа №1.

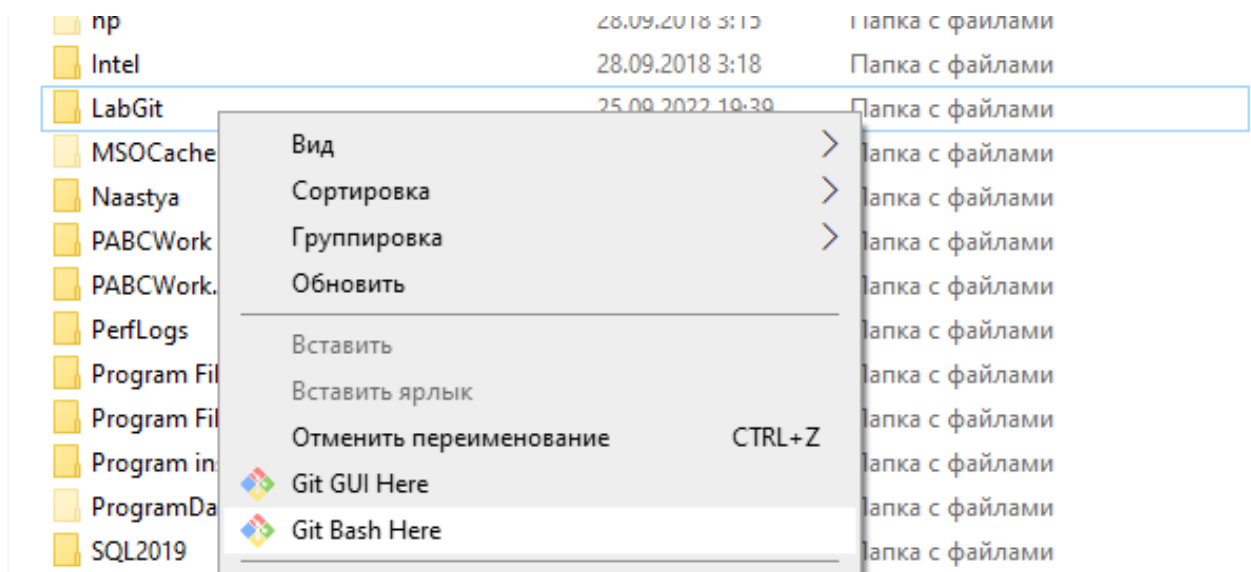
Первоначальная настройка

Цель: установив Git, сделать настройку в эмуляторе командной строки.

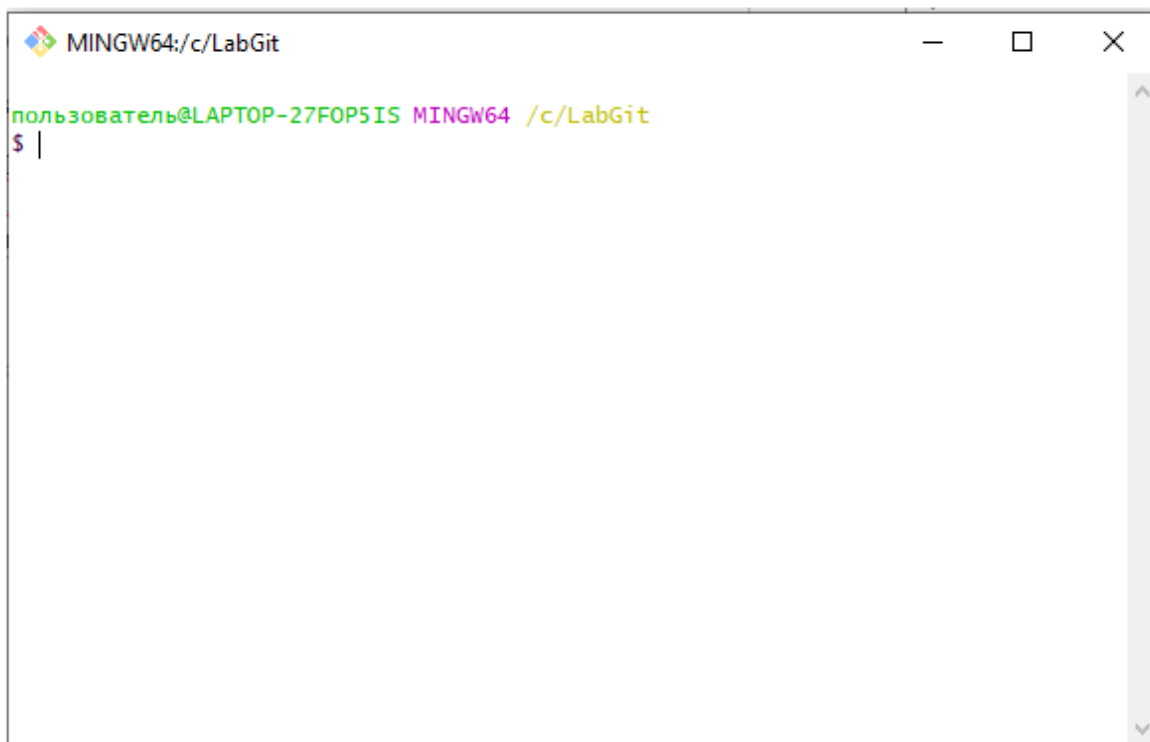
1. На диске C: создадим папку LabGit для хранения проектов.

!Не рекомендуется создавать папки с пробелами или кириллическими символами в названии.

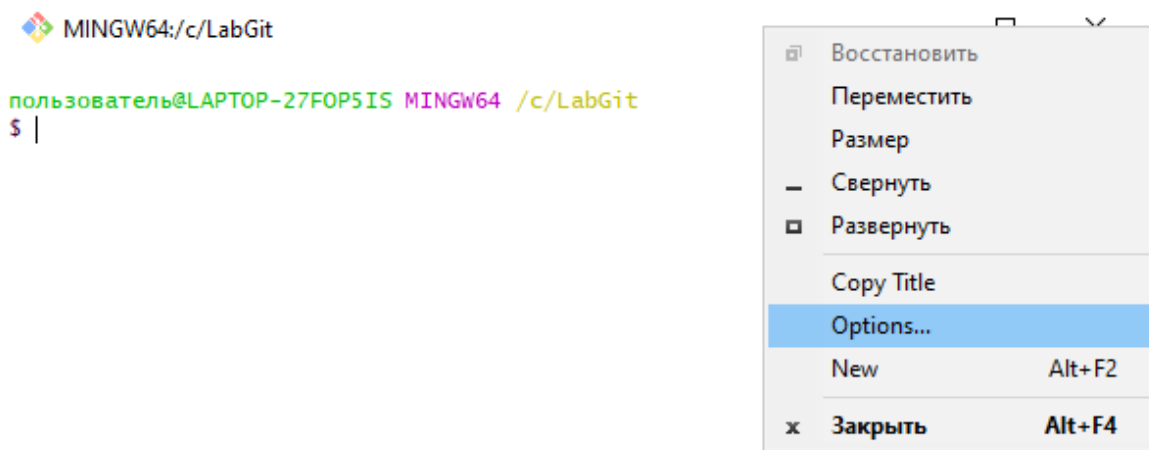
Запустите приложение Git Bash (клик правой кнопкой мыши на созданный каталог).



В появившемся окне, как в командной строке, мы и будем работать:



2. Запустите Git Bash. Нажатием правой кнопкой мыши можно открыть вкладку Options (Опции) и изменить внешний вид окна (цвет фона и текста) для дальнейшего удобства работы.



3. Добавим наше имя пользователя и адрес электронной почты. Эти данные нужны, чтобы отслеживать какие изменения кем были сделаны. Для этого введем команды (скопировать и с помощью клавиш Shift+Ins вставить):

```
git config --global user.name "My Name"
```

```
git config --global user.email myEmail@example.com
```

После написания команды для ее выполнения нажимаем Enter.

```
MINGW64:/c/LabGit
```

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit  
$ git config --global user.name "for-you-nastya"
```

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit  
$ git config --global user.email "nastenka.kravchenko.01@mail.ru"
```

Данные будут сохранены для всех ваших действий в Git. Чтобы сделать эти настройки глобальными, то есть применимыми ко всем проектам, необходимо добавить флаг `--global`. Если вы этого не сделаете, они будут распространяться только на текущий репозиторий.

4. Напишем команду:

```
git config --list
```

Заметим, что name и email такие, как мы ввели выше. Данная команда позволяет просмотреть все настройки системы.

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit  
$ git config --list  
diff.astextplain.textconv=astextplain  
filter.lfs.clean=git-lfs clean -- %f  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
filter.lfs.required=true  
http.sslbackend=openssl  
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt  
core.autocrlf=true  
core.fscache=true  
core.symlinks=false  
pull.rebase=false  
credential.helper=manager-core  
credential.https://dev.azure.com.usehttppath=true  
init.defaultbranch=master  
filter.lfs.required=true  
filter.lfs.clean=git-lfs clean -- %f  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
user.name=for-you-nastya  
user.email=nastenka.kravchenko.01@mail.ru
```

5. При возникновении трудностей можно открыть список команд и параметрами для них. Для этого вводим команду:

```
git --help
```

```
пользователь@LAPTOP-27F0P5IS MINGW64 /c/LabGit
```

```
$ git --help
```

```
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      [--super-prefix=<path>] [--config-env=<name>=<envvar>]
      <command> [<args>]
```

```
These are common Git commands used in various situations:
```

```
start a working area (see also: git help tutorial)
```

```
  clone          Clone a repository into a new directory
  init           Create an empty Git repository or reinitialize an existing
one
```

```
work on the current change (see also: git help everyday)
```

```
  add           Add file contents to the index
  mv            Move or rename a file, a directory, or a symlink
  restore       Restore working tree files
  rm            Remove files from the working tree and from the index
```

Изучите команды, в последующем мы ознакомимся с ними более подробно.