

# Лабораторная работа № 5

## Управление версиями с помощью Git

Для выполнения этого задания рекомендуется использовать ОС GNU/Linux. В случае использования ОС семейства Windows, необходимо установить эмулятор среды UNIX — CyGNUs (скачать дистрибутив и узнать подробности об установке можно по адресу <http://git-scm.com/download> или <https://msysgit.github.io/>).

### Задание 1. Работа с локальным репозиторием

1. Установите на свой компьютер Git.
2. Каждый коммит в Git связан с определённой персоной, выполнившей то или иное действие. Поэтому, первым делом введите свои данные:

```
git config --global user.name "Имя"  
git config --global user.email "адрес почты"
```

3. Проверьте конфигурацию Git, выполнив команду

```
git config --list
```

4. Что бы настроить Git на использование HTTP-прокси надо создать файл настроек в своей домашней директории. Для этого надо выполнить команду добавления данных доступа к прокси-серверу по следующему шаблону:

```
git config --global http.proxy http://proxyuser:proxypass@proxyaddress:proxyport
```

Например, в компьютерных классах Университета можно ввести команду:

```
git config --global http.proxy http://fio_user:kurs2016@proxy.asu.ru:3168
```

Проверьте содержимое файла.

5. Создайте на своём компьютере директорию `PM-projectName`, здесь вместо `projectName` напишите имя вашего командного проекта. Скопируйте туда все свои файлы, созданные в рамках этой дисциплины (можно использовать поддиректории).

Все дальнейшие команды выполняются в директории `PM-projectName`.

6. Инициализируйте репозиторий:

```
git init
```

Эта ветвь в Git, по умолчанию, называется `master`.

7. Добавьте все файлы из директории `PM-projectName` в индекс Git:

```
git add .
```

В дальнейшем, если надо обновить уже существующие файлы, используйте команду (см. рис. 1)

```
git add -u
```

В именах файлов можно использовать шаблоны.

8. Добавьте все проиндексированные файлы в репозиторий:

```
git commit -m "Новые файлы."
```

Эта ветвь в Git, по умолчанию, называется `master`.

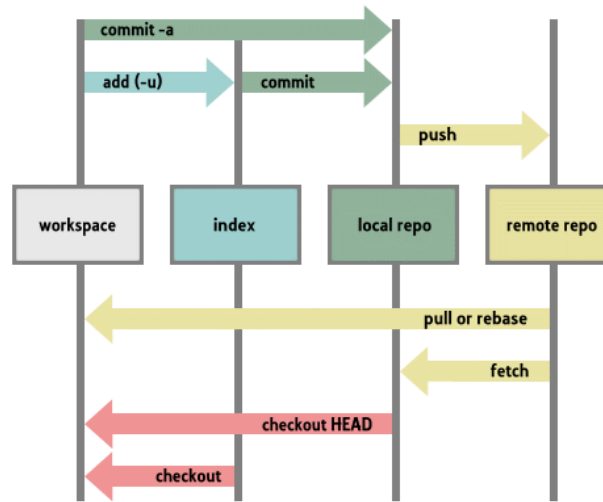


Рис. 1. Структура Git и процессы управления репозиторием

## Задание 2. Работа с удалённым репозиторием

1. Зарегистрируйтесь на [GitHub](https://github.com).
2. Создайте репозиторий PM-projectName. Обязательно инициализируйте репозиторий файлом README.
3. Для синхронизации локального и удалённого репозитория выполните команду:

```
git pull https://github.com/ВашеИмя/ВашРепозиторий.git
```

4. Скопируйте локальный репозиторий в удалённый:

```
git push https://github.com/ВашеИмя/ВашРепозиторий.git
```

5. Скопируйте ветку <https://github.com/zhurenkov/PM> в свой репозиторий, используя инструмент Fork (API GitHub). Для этого перейдите в нужную директорию и нажмите кнопку [Fork](#).
6. Для копирования своего репозитория из GitHub на другой компьютер, выполните пункты 1–6 из первого задания и задайте команду

```
git clone https://github.com/ВашеИмя/ВашРепозиторий.git
```

В отчёт (на Moodle) напишите адрес своего удалённого репозитория на GitHub.