

Laboratory work

Данная лабораторная работа посвящена изучению систем контроля версий на примере **Git**.

```
$ open https://git-scm.com
```

Tasks

- ☐ 1. Создать публичный репозиторий с названием **lab02** и с лицензией **MIT**
- ☐ 2. Сгенерировать токен для доступа к сервису **GitHub** с правами **repo**
- ☐ 3. Ознакомиться со ссылками учебного материала
- ☐ 4. Выполнить инструкцию учебного материала
- ☐

• Tutorial

Создаем переменные **GITHUB_USERNAME**, **GITHUB_EMAIL** и **GIST_TOKEN**, а так же связываем команду `edit` с вызовом редактора `subl3`

```
$ export GITHUB_USERNAME = OzoNeTT # Присваиваем
переменной GITHUB_USERNAME свое имя на GitHub
$ export GITHUB_EMAIL = gpoff12@mail.ru # Присваиваем
переменной GITHUB_EMAIL свой email
$ export GITHUB_TOKEN = xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx # Присваиваем
переменной GIST_TOKEN свой токен
$ alias edit = subl3 # Биндим
команду edit на subl3
```

Активируем скрипт, лежащий в директории `scripts/`

```
$ cd ${ GITHUB_USERNAME } / workspace # Переходим в директорию workspace
$ source scripts / activate # Активируем скрипт
```

Создаем `.config` и настраиваем его

```
$ mkdir ~/ .config # Создаем.config
$ cat > ~/ .config / hub << EOF # Редактируем его
github.com:
-user : ${ GITHUB_USERNAME }
oauth_token: ${ GITHUB_TOKEN }
protocol: https
EOF # Закрываем редактор
$ git config --global hub.protocol https # Устанавливаем глобальный параметр гита
протокол на https
```

Работа с **git**

```
$ mkdir projects / lab02 && cd projects / lab02
# Создаем директорию lab02 и переходим в нее
```

```

$ git init #
Создание подкаталога с.git
Инициализирован пустой репозиторий Git в / home / ozone / OzoNeTT / workspace /
projects / lab02 / .git /

$ git config --global user.name ${ GITHUB_USERNAME } # Устанавливаем глобальный
параметр гита имя
$ git config --global user.email ${ GITHUB_EMAIL } # Устанавливаем глобальный
параметр гита email
# check your git global settings
$ git config - e --global # Простматриваем глобальные
параметры
[hub]
protocol = https
[user]
name = OzoNeTT
email = gpoff12@mail.ru

$ git remote add origin https ://github.com/${GITHUB_USERNAME}/lab02.git
$ git pull origin master # Вливаем изменения из удаленного
репозитория в локальный
remote : Enumerating objects : 3, done.
remote : Counting objects : 100 % (3 / 3), done.
remote : Compressing objects : 100 % (2 / 2), done.
remote : Total 3 (delta 0), reused 0 (delta 0), pack - reused 0
Распаковка объектов : 100 % (3 / 3), готово.
Из https ://github.com/OzoNeTT/lab02
*branch master->FETCH_HEAD

$ touch README.md
$ git status # Смотрим статус существующих файлов в репозитории
На ветке master
Неотслеживаемые файлы :
(используйте «git add <файл>...», чтобы добавить в то, что будет включено в коммит)

README.md

ничего не добавлено в коммит, но есть неотслеживаемые файлы(используйте «git add»,
чтобы отслеживать их)

$ git add README.md
$ git commit - m"added README.md"
[master 540948c] added README.md
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md

$ git push origin master # Вливаем изменения на
удаленный репозиторий
Username for 'https://github.com': OzoNeTT
Password for 'https://OzoNeTT@github.com' :
Перечисление объектов : 4, готово.
Подсчет объектов : 100 % (4 / 4), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов : 100 % (2 / 2), готово.
Запись объектов : 100 % (3 / 3), 277 bytes | 277.00 KiB / s, готово.
Всего 3 (изменения 0), повторно использовано 0 (изменения 0)
To https ://github.com/OzoNeTT/lab02.git
449c538..540948c master->master

```

Добавить на сервисе **GitHub** в репозитории **lab02** файл **.gitignore** со следующим содержанием :

```
*build* /
*install* /
*.swp
.idea /
```

Скачиваем и разархивируем master branch

```
$ git pull origin master # Пулим основную ветвь с гитхаба
remote : Enumerating objects : 4, done.
remote : Counting objects : 100 % (4 / 4), done.
remote : Compressing objects : 100 % (2 / 2), done.
remote : Total 3 (delta 0), reused 0 (delta 0), pack - reused 0
Распаковка объектов : 100 % (3 / 3), готово.
Из https ://github.com/OzoNeTT/lab02
*branch master->FETCH_HEAD
540948c..c0e3569 master->origin / master
Обновление 540948c..c0e3569
Fast - forward
.gitignore | 4 ++++
1 file changed, 4 insertions(+)
create mode 100644.gitignore
```

```
$ git log # Выводим логи
commit c0e356995e212f751aa78470ebb9668706e61bce(HEAD->master, origin / master)
Author: OzoNeTT <45043082 + OzoNeTT@users.noreply.github.com>
Date : Mon Mar 18 17 : 45 : 15 2019 + 0300
```

Create.gitignore

```
commit 540948ca071dc8dc4544a8ffd57b5a6c20d3c1b6
Author : OzoNeTT <gpoff12@mail.ru>
Date : Mon Mar 18 17 : 43 : 45 2019 + 0300
```

added README.md

```
commit 449c5389e362396c20733593a67ff91736e15f82
Author : OzoNeTT <45043082 + OzoNeTT@users.noreply.github.com>
Date : Mon Mar 18 17 : 34 : 38 2019 + 0300
```

Initial commit

Создание дирректорий **sources**, **include** и **examples** и создание в них файлов

```
$ mkdir sources # Создаем дирректорию source
$ mkdir include # Создаем дирректорию include
$ mkdir examples # Создаем дирректорию examples
$ cat > sources / print.cpp << EOF # Создаем print.cpp в source / и
редактируем его
#include <print.hpp>
```

```
void print(const std::string& text, std::ostream& out)
{
    out << text;
}
```

```
void print(const std::string& text, std::ofstream& out)
{
    out << text;
}
EOF
```

Создание **print.hpp** и его редактирование

```
$ cat > include / print.hpp << EOF
#include <fstream>
#include <iostream>
#include <string>

void print(const std::string& text, std::ofstream& out);
void print(const std::string& text, std::ostream& out = std::cout);
EOF
```

Создание **example1.cpp** и его редактирование

```
$ cat > examples / example1.cpp << EOF
#include <print.hpp>

int main(int argc, char** argv)
{
    print("hello");
}
EOF
```

Создание **example2.cpp** и его редактирование

```
$ cat > examples / example2.cpp << EOF
#include <print.hpp>

#include <fstream>

int main(int argc, char** argv)
{
    std::ofstream file("log.txt");
    print(std::string("hello"), file);
}
EOF
```

Редактирование файла README.md

```
$ edit README.md      # Открываем README.md в subl3
```

Просмотр статуса репозитория и добавление на **GITHUB**

```
$ git status      # Смотрим файлы, которые были изменены
На ветке master
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git checkout -- <file>..." to discard changes in working directory)
```

```
modified : README.md
```

Неотслеживаемые файлы :

(используйте «git add <файл>...», чтобы добавить в то, что будет включено в коммит)

```
examples /
include /
sources /
```

ничего не добавлено в коммит, но есть неотслеживаемые файлы(используйте «git add», чтобы отслеживать их)

```
$ git add .      # Фиксируем изменения
$ git commit - m"added sources"      # Коммитим изменения
[master 540948c] added sources
5 files changed, 324 insertions(+)
```

```

create mode 100644 examples / example1.cpp
create mode 100644 examples / example2.cpp
create mode 100644 include / print.hpp
create mode 100644 sources / print.cpp
$ git push origin master          # Пушим(Загружаем) на ГитХаб
Username for 'https://github.com': OzoNeTT
Password for 'https://OzoNeTT@github.com' :
Enumerating objects : 12, done.
Counting objects : 100 % (12 / 12), done.
Delta compression using up to 4 threads
Compressing objects : 100 % (8 / 8), done.
Writing objects : 100 % (10 / 10), 5.00 KiB | 2.50 MiB / s, done.
Total 10 (delta 0), reused 0 (delta 0)
To https://github.com/OzoNeTT/lab02.git
449f23d..540948c  master->master

```

Report

Создание отчета по лабораторной работе

```

$ cd ~/ workspace / labs /
$ export LAB_NUMBER = 02
$ git clone https://github.com/tp-labs/lab${LAB_NUMBER}.git tasks/lab${LAB_NUMBER}
# Копируем репозиторий
$ mkdir reports / lab${ LAB_NUMBER }
$ cp tasks / lab${ LAB_NUMBER } / README.md reports / lab${ LAB_NUMBER } / REPORT.md
$ cd reports / lab${ LAB_NUMBER }
$ edit REPORT.md
$ gistup - m "lab${LAB_NUMBER}"

```

Homework

Part I

1. Создайте пустой репозиторий на сервисе github.com(или gitlab.com, или bitbucket.com).
2. Выполните инструкцию по созданию первого коммита на странице репозитория, созданного на предыдущем шаге.
3. Создайте файл hello_world.cpp в локальной копии репозитория(который должен был появиться на шаге 2).Реализуйте программу **Hello world** на языке C++ используя плохой стиль кода.Например, после заголовочных файлов вставьте строку `using namespace std;` .
4. Добавьте этот файл в локальную копию репозитория.
5. Закоммитьте изменения с *осмысленным* сообщением.
6. Измените исходный код так, чтобы программа через стандартный поток ввода запрашивалось имя пользователя.А в стандартный поток вывода печаталось сообщение `hello world from @name`, где @name имя пользователя.
7. Закоммитьте новую версию программы.Почему не надо добавлять файл повторно `git add`?
8. Запуште изменения в удалённый репозиторий.

9. Проверьте, что история коммитов доступна в удалённый репозитории.

Part II

Note: *Работать продолжайте с теми же репозиториями, что и в первой части задания.*

1. В локальной копии репозитория создайте локальную ветку `patch1`.
2. Внесите изменения в ветке `patch1` по исправлению кода и избавления от `using namespace std;`.
3. **commit, push** локальную ветку в удалённый репозиторий.
4. Проверьте, что ветка `patch1` доступна в удалённый репозитории.
5. Создайте pull - request `patch1->master`.
6. В локальной копии в ветке `patch1` добавьте в исходный код комментарии.
7. **commit, push**.
8. Проверьте, что новые изменения есть в созданном на `**stage 5 * *` pull - request
9. В удалённый репозитории выполните слияние PR `patch1->master` и удалите ветку `patch1` в удалённом репозитории.
10. Локально выполните **pull**.
11. С помощью команды **git log** просмотрите историю в локальной версии ветки `master`.
12. Удалите локальную ветку `patch1`.

Part III

Note: *Работать продолжайте с теми же репозиториями, что и в первой части задания.*

1. Создайте новую локальную ветку `patch2`.
2. Измените *code style* с помощью утилиты `**clang - format * *`. Например, используя опцию `-style=Mozilla`.
3. **commit, push**, создайте pull - request `patch2->master`.
4. В ветке **master** в удалённом репозитории измените комментарии, например, расставьте знаки препинания, переведите комментарии на другой язык.
5. Убедитесь, что в pull - request появились *конфликтны*.
6. Для этого локально выполните **pull +rebase** (точную последовательность команд, следует узнать самостоятельно). **Исправьте конфликты**.
7. Сделайте *force push* в ветку `patch2`
8. Убедитесь, что в pull - request пропали конфликты.
9. Вмержите pull - request `patch2->master`.

Links

- [hub](#)
- [GitHub](#)
- [Bitbucket](#)
- [Gitlab](#)
- [LearnGitBranching](#)