

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Кафедра Компьютерных Систем и Программных Технологий

ОТЧЕТ

по лабораторной работе №2

Тема: «Система контроля версий Git»

Дисциплина: «Методы и средства защиты информации»

Выполнил: студент гр. 53501/2
Пономарев М.А

Преподаватель
Антонов А.П

Санкт-Петербург
2015

Содержание

1	Задание	2
2	Выполнение	3
2.1	Создаём временный репозиторий	3
2.2	Получить содержимое репозитория	3
2.3	Добавить новую папку и первый файл под контроль версий	4
2.4	Зафиксировать изменения в локальном репозитории	4
2.5	Внести изменения в файл и просмотреть различия	4
2.6	Зафиксировать изменения в локальном репозитории	4
2.7	Отменить локальные изменения	5
2.8	Внести изменения в файл и просмотреть различия	5
2.9	Зафиксировать изменения в локальном репозитории	5
2.10	Зафиксировать изменения в центральном репозитории	6
2.11	Получить изменения из центрального репозитория	6

1 Задание

- а) Изучить справку основных команд
- б) Получить содержимое репозитория
- в) Добавить новую папку и первый файл под контроль версий
- г) Зафиксировать изменения в локальном репозитории
- д) Внести изменения в файл и просмотреть различия
- е) Отменить локальные изменения
- ж) Внести изменения в файл и просмотреть различия
- з) Зафиксировать изменения в локальном репозитории
- и) Зафиксировать изменения в центральном репозитории
- к) Получить изменения из центрального репозитория
- л) *Описать свои впечатления от работы*

2 Выполнение

После изучения основных возможностей Git'а перейдём к выполнению задания:

2.1 Создаём временный репозиторий

Для изучения возможностей системы контроля версий создадим временный репозиторий на github под названием «tempRepo». Далее, инициализируем пустой git репозиторий локально, добавляем файл README, для дальнейшей работы с репозиторием и создаём новую (главную) ветку разработки, фиксируем изменения в центральном репозитории.

```
Misha$ (temp) echo "# tempRepo" >> README.md
Misha$ (temp) git init
Initialized empty Git repository in /Users/Misha/Programming/Projects/temp/.git/
Misha$ (temp) git add README.md
Misha$ (temp) git commit -m "first commit"
[master (root-commit) 52a8251] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
Misha$ (temp) git remote add origin https://github.com/Allycen/tempRepo.git
Misha$ (temp) git push -u origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 221 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Allycen/tempRepo.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
Misha$ (temp)
```

2.2 Получить содержимое репозитория

Для получения содержимого центрального репозитория используется команда «git pull»

```
Misha$ (temp) git pull
Already up-to-date.
```

Система говорит нам, что локальный репозиторий не отличается от центрального.

2.3 Добавить новую папку и первый файл под контроль версий

```
Misha$: (temp) mkdir first_folder
Misha$: (temp) touch first_file
Misha$: (temp) git st
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        first_file
nothing added to commit but untracked files present (use "git add" to track)
Misha$: (temp) git add .
```

2.4 Зафиксировать изменения в локальном репозитории

```
Misha$: (temp) git commit -m "add first files"
[master 016761a] add first files
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 first_file
```

2.5 Внести изменения в файл и просмотреть различия

```
Misha$: (temp) git st
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
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:   first_file
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .gitignore
no changes added to commit (use "git add" and/or "git commit -a")
```

2.6 Зафиксировать изменения в локальном репозитории

```
Misha$: (temp) git add .
Misha$: (temp) git commit -m "change first_file"
[master 3c8c25d] change first_file
2 files changed, 2 insertions(+)
create mode 100644 .gitignore
```

2.7 Отменить локальные изменения

```
Misha$: (temp) git hist
* 3c8c25d 2015-06-01 | change first_file (HEAD, master) [Allycen]
* 016761a 2015-06-01 | add first files [Allycen]
* 52a8251 2015-06-01 | first commit (origin/master) [Allycen]
Misha$: (temp) git reset --soft HEAD~1
Misha$: (temp) git hist
* 016761a 2015-06-01 | add first files (HEAD, master) [Allycen]
* 52a8251 2015-06-01 | first commit (origin/master) [Allycen]
```

2.8 Внести изменения в файл и просмотреть различия

```
Misha$: (temp) git st
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   .gitignore
    modified:   first_file

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:   first_file
```

Здесь очень важно, что когда мы применили команду «reset –soft», мы всего лишь откатали последний коммит, но не обновление файла в системе контроля версий!

2.9 Зафиксировать изменения в локальном репозитории

```
Misha$: (temp) git add .
Misha$: (temp) git commit -m "one more commit"
[master 864487a] one more commit
2 files changed, 2 insertions(+)
create mode 100644 .gitignore
```

2.10 Зафиксировать изменения в центральном репозитории

```
Misha$: (temp) git push
warning: push.default is unset; its implicit value has changed in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the traditional behavior, use:

    git config --global push.default matching

To squelch this message and adopt the new behavior now, use:
    git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (7/7), 585 bytes | 0 bytes/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To https://github.com/Allycen/tempRepo.git
  52a8251..864487a master -> master
Misha$: (temp) git st
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
```

2.11 Получить изменения из центрального репозитория

```
Misha$: (temp) git pull
Already up-to-date.
```