

Отчёт по лабораторной работе №2

Управление версиями

Логинов Андрей НБИбд 01-21

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	11
4	Контрольные вопросы	12
	Список литературы	16

List of Figures

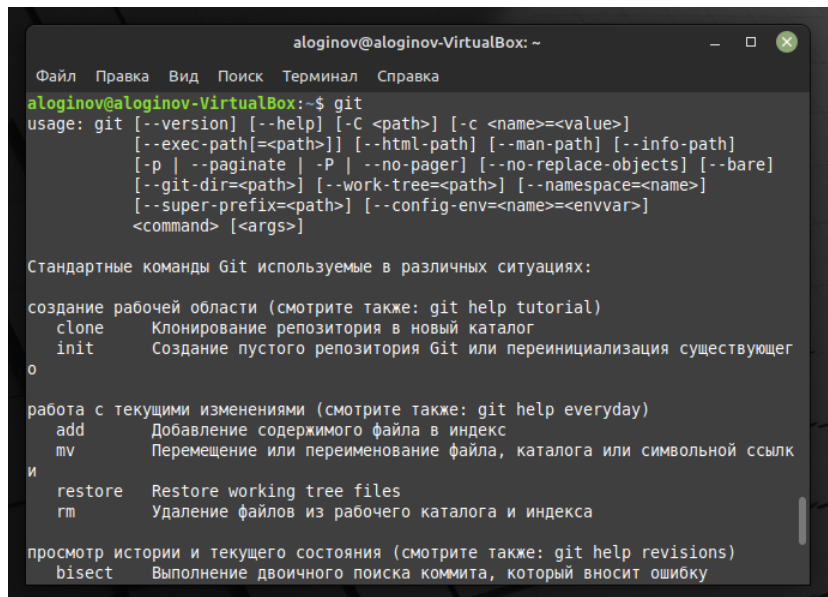
2.1	Загрузка пакетов	5
2.2	Параметры репозитория	6
2.3	rsa-4096	6
2.4	ed25519	7
2.5	GPG ключ	7
2.6	GPG ключ	8
2.7	Параметры репозитория	8
2.8	Связь репозитория с аккаунтом	9
2.9	Загрузка шаблона	9
2.10	Первый коммит	10

1 Цель работы

Целью данной работы является изучение идеологии и применения средств контроля версий и освоение умений работать с git.

2 Выполнение лабораторной работы

Устанавливаем git, git-flow и gh.



```
aloginov@aloginov-VirtualBox: ~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
aloginov@aloginov-VirtualBox:~$ git  
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>]  
  
Стандартные команды Git используемые в различных ситуациях:  
  
создание рабочей области (смотрите также: git help tutorial)  
  clone      Клонирование репозитория в новый каталог  
  init       Создание пустого репозитория Git или переинициализация существующег  
o  
  
работа с текущими изменениями (смотрите также: git help everyday)  
  add        Добавление содержимого файла в индекс  
  mv         Перемещение или переименование файла, каталога или символической ссылк  
и  
  restore    Restore working tree files  
  rm         Удаление файлов из рабочего каталога и индекса  
  
просмотр истории и текущего состояния (смотрите также: git help revisions)  
  bisect     Выполнение двоичного поиска коммита, который вносит ошибку
```

Figure 2.1: Загрузка пакетов

Зададим имя и email владельца репозитория, кодировку и прочие параметры.

```
aloginov@aloginov-VirtualBox: ~
Файл  Правка  Вид  Поиск  Терминал  Справка

совместная работа (смотрите также: git help workflows)
  fetch      Загрузка объектов и ссылок из другого репозитория
  pull       Извлечение изменений и объединение с другим репозиторием или локаль
ной веткой
  push       Обновление внешних ссылок и связанных объектов

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
aloginov@aloginov-VirtualBox:~$
aloginov@aloginov-VirtualBox:~$
aloginov@aloginov-VirtualBox:~$
aloginov@aloginov-VirtualBox:~$ git config --global user.name "andeyloginov"
git config --global user.email "1032210098@pfur.ru"

git config --global core.quotepath false
git config --global init.defaultBranch master
git config --global core.autocrlf input
git config --global core.safecrlf warn
aloginov@aloginov-VirtualBox:~$ ssh-keygen -t rsa -b 4096
```

Figure 2.2: Параметры репозитория

Создаем SSH ключи

```
aloginov@aloginov-VirtualBox: ~
Файл  Правка  Вид  Поиск  Терминал  Справка

git config --global core.safecrlf warn
aloginov@aloginov-VirtualBox:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aloginov/.ssh/id_rsa):
Created directory '/home/aloginov/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aloginov/.ssh/id_rsa
Your public key has been saved in /home/aloginov/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:WGZ9f5CuDwbNArokkWacBruTRq4GB0tB/fFdMg/sYTM aloginov@aloginov-VirtualBox
The key's randomart image is:
+---[RSA 4096]-----+
|.oo      .          |
|..= +   E .   .    |
|..o X o O @ . o    |
|.= * o B + = o .   |
|o B . + S o o o .  |
|. + . o . o . .    |
|..      .   +      |
|.      .   o       |
|..      .          |
+-----[SHA256]-----+
aloginov@aloginov-VirtualBox:~$
```

Figure 2.3: rsa-4096

```
aloginov@aloginov-VirtualBox: ~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
|-----[SHA256]-----+  
aloginov@aloginov-VirtualBox:~$ ssh-keygen -t ed25519  
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/home/aloginov/.ssh/id_ed25519):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/aloginov/.ssh/id_ed25519  
Your public key has been saved in /home/aloginov/.ssh/id_ed25519.pub  
The key fingerprint is:  
SHA256:wmmuT0Sf11HdqCnmAnBpkfh7YnyIVc/7MTEZe51qWcA aloginov@aloginov-VirtualBox  
The key's randomart image is:  
+--[ED25519 256]--+  
| ..+ ..o |  
| o = . .E. o |  
| =.. o .*. . |  
| o+o .B..o |  
| ++*So.o.++ |  
| .+*.+.o o+ |  
| .o+ . .o |  
| o . |  
| ... |  
+-----[SHA256]-----+  
aloginov@aloginov-VirtualBox:~$
```

Figure 2.4: ed25519

Создаем GPG ключ

```
aloginov@aloginov-VirtualBox: ~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
gpg: ключ 37FC9B3DCC2FF40D помечен как абсолютно доверенный  
gpg: создан каталог '/home/aloginov/.gnupg/openpgp-revocs.d'  
gpg: сертификат отзыва записан в '/home/aloginov/.gnupg/openpgp-revocs.d/4FB0C2D  
F5827C09A934BD75137FC9B3DCC2FF40D.rev'.  
открытый и секретный ключи созданы и подписаны.  
  
pub   rsa4096 2022-09-08 [SC]  
      4FB0C2DF5827C09A934BD75137FC9B3DCC2FF40D  
uid           andeyloginov <1032210098@pfur.ru>  
sub   rsa4096 2022-09-08 [E]  
  
aloginov@aloginov-VirtualBox:~$ gpg --list-secret-keys --keyid-format LONG  
gpg: проверка таблицы доверия  
gpg: marginals needed: 3  completes needed: 1  trust model: pgp  
gpg: глубина: 0  достоверных: 1  подписанных: 0  доверие: 0-, 0q, 0n, 0m, 0f  
      , 1u  
/home/aloginov/.gnupg/pubring.kbx  
-----  
sec   rsa4096/37FC9B3DCC2FF40D 2022-09-08 [SC]  
      4FB0C2DF5827C09A934BD75137FC9B3DCC2FF40D  
uid           [ абсолютно ] andeyloginov <1032210098@pfur.ru>  
ssb   rsa4096/BE314568D3C07AC5 2022-09-08 [E]  
  
aloginov@aloginov-VirtualBox:~$ gpg --armor --export 37FC9B3DCC2FF40D
```

Figure 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

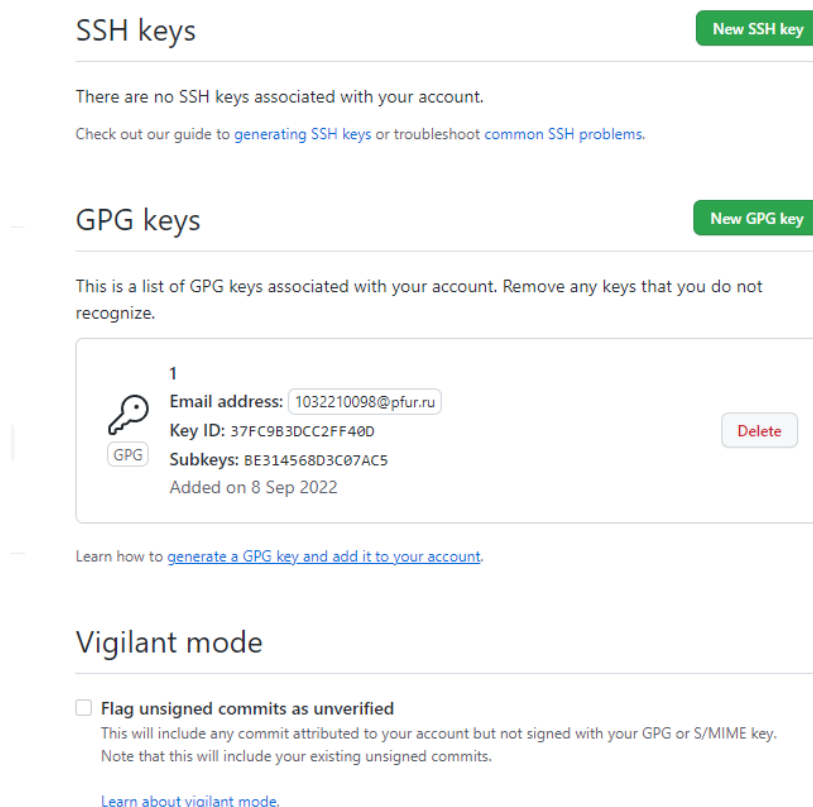


Figure 2.6: GPG ключ

Настройка автоматических подписей коммитов git

```
-----END PGP PUBLIC KEY BLOCK-----
aloginov@aloginov-VirtualBox:~$ 
aloginov@aloginov-VirtualBox:~$ 
aloginov@aloginov-VirtualBox:~$ 
aloginov@aloginov-VirtualBox:~$ 
aloginov@aloginov-VirtualBox:~$ git config --global user.signingkey 37FC9B3DCC2F
F40D
git config --global commit.gpgsign true
git config --global gpg.program $(which gpg2)
aloginov@aloginov-VirtualBox:~$
```

Figure 2.7: Параметры репозитория

Настройка gh


```
aloginov@aloginov-VirtualBox: ~  
Файл Правка Вид Поиск Терминал Справка  
aloginov@aloginov-VirtualBox:~$ git config --global user.signingkey 37FC9B3DCC2F  
F40D  
git config --global commit.gpgsign true  
git config --global gpg.program $(which gpg2)  
aloginov@aloginov-VirtualBox:~$  
aloginov@aloginov-VirtualBox:~$  
aloginov@aloginov-VirtualBox:~$ gh auth login  
? What account do you want to log into? GitHub.com  
? What is your preferred protocol for Git operations? SSH  
? Upload your SSH public key to your GitHub account? /home/aloginov/.ssh/id_rsa.  
pub  
? How would you like to authenticate GitHub CLI? Login with a web browser  
  
! First copy your one-time code: CFF4-251A  
- Press Enter to open github.com in your browser...  
✓ Authentication complete. Press Enter to continue...  
2022-09-08T16:26:22Z ERROR viaduct::backend::ffi Missing HTTP status  
2022-09-08T16:26:22Z ERROR viaduct::backend::ffi Missing HTTP status  
  
- gh config set -h github.com git_protocol ssh  
✓ Configured git protocol  
✓ Uploaded the SSH key to your GitHub account: /home/aloginov/.ssh/id_rsa.pub  
✓ Logged in as andeyloginov  
aloginov@aloginov-VirtualBox:~$
```

Figure 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация

```
aloginov@aloginov-VirtualBox: ~/work/study/2021-2022/Операционные системы  
Файл Правка Вид Поиск Терминал Справка  
tation-markdown-template.git) зарегистрирован по пути «template/presentation»  
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-r  
eport-template.git) зарегистрирован по пути «template/report»  
Клонирование в «/home/aloginov/work/study/2021-2022/Операционные системы/os-intr  
o/template/presentation»...  
remote: Enumerating objects: 71, done.  
remote: Counting objects: 100% (71/71), done.  
remote: Compressing objects: 100% (49/49), done.  
remote: Total 71 (delta 23), reused 68 (delta 20), pack-reused 0  
Получение объектов: 100% (71/71), 88.89 КиБ | 489.00 КиБ/с, готово.  
Определение изменений: 100% (23/23), готово.  
Клонирование в «/home/aloginov/work/study/2021-2022/Операционные системы/os-intr  
o/template/report»...  
remote: Enumerating objects: 78, done.  
remote: Counting objects: 100% (78/78), done.  
remote: Compressing objects: 100% (52/52), done.  
remote: Total 78 (delta 31), reused 69 (delta 22), pack-reused 0  
Получение объектов: 100% (78/78), 292.27 КиБ | 3.28 МиБ/с, готово.  
Определение изменений: 100% (31/31), готово.  
Submodule path 'template/presentation': checked out '2703b47423792d472694aaf7555  
a5626dce51a25'  
Submodule path 'template/report': checked out 'df7b2ef80f8def3b9a496f8695277469a  
1a7842a'  
aloginov@aloginov-VirtualBox:~/work/study/2021-2022/Операционные системы$
```

Figure 2.9: Загрузка шаблона

Подготовка репозитория и коммит изменений

```
aloginov@aloginov-VirtualBox: ~/work/study/2021-2022/Операционные системы/os...  
Файл Правка Вид Поиск Терминал Справка  
numeric.csl  
create mode 100644 project-personal/stage5/report/report.md  
create mode 100644 project-personal/stage6/presentation/Makefile  
create mode 100644 project-personal/stage6/presentation/image/kulyabov.jpg  
create mode 100644 project-personal/stage6/presentation/presentation.md  
create mode 100644 project-personal/stage6/report/Makefile  
create mode 100644 project-personal/stage6/report/bib/cite.bib  
create mode 100644 project-personal/stage6/report/image/placeimg_800_600_tech.jpg  
create mode 100644 project-personal/stage6/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl  
create mode 100644 project-personal/stage6/report/report.md  
Перечисление объектов: 21, готово.  
Подсчет объектов: 100% (21/21), готово.  
При сжатии изменений используется до 4 потоков  
Сжатие объектов: 100% (17/17), готово.  
Запись объектов: 100% (20/20), 311.68 КиБ | 2.66 МБ/с, готово.  
Всего 20 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0  
remote: Resolving deltas: 100% (1/1), completed with 1 local object.  
To github.com:andeyloginov/study_2021-2022_os-intro.git  
c4e532e..4b2480f master -> master  
aloginov@aloginov-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$
```

Figure 2.10: Первый коммит

3 Вывод

Мы приобрели практические навыки работы с сервисом github.

4 Контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

- хранилище - пространство на накопителе где расположен репозиторий
- commit - сохранение состояния хранилища
- история - список изменений хранилища (коммитов)
- рабочая копия - локальная копия сетевого репозитория, в которой работает программист. Текущее состояние файлов проекта, основанное на версии, загруженной из хранилища (обычно на последней)

3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере. Доступ к нему осуществлялся через специальное клиентское приложение. В качестве примеров таких программных продуктов можно привести CVS, Subversion.

Распределенные системы контроля версий (Distributed Version Control System, DVCS) позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой. При этом можно выделить центральный репозиторий (условно), в который будут отправляться изменения из локальных и, с ним же эти локальные репозитории будут синхронизироваться. При работе с такой системой, пользователи периодически синхронизируют свои локальные репозитории с центральным и работают непосредственно со своей локальной копией. После внесения достаточного количества изменений в локальную копию они (изменения) отправляются на сервер. При этом сервер, чаще всего, выбирается условно, т.к. в большинстве DVCS нет такого понятия как “выделенный сервер с центральным репозиторием”.

4. Опишите действия с VCS при единоличной работе с хранилищем.

Один пользователь работает над проектом и по мере необходимости делает коммиты, сохраняя определенные этапы.

5. Опишите порядок работы с общим хранилищем VCS.

Несколько пользователей работают каждый над своей частью проекта. При этом каждый должен работать в своей ветки. При завершении работы ветка пользователя сливается с основной веткой проекта.

6. Каковы основные задачи, решаемые инструментальным средством git?

- Ведение истории версий проекта: журнал (log), метки (tags), ветвления (branches).

- Работа с изменениями: выявление (diff), слияние (patch, merge).
- Обеспечение совместной работы: получение версии с сервера, загрузка обновлений на сервер.

7. Назовите и дайте краткую характеристику командам git.

- git config - установка параметров
- git status - полный список изменений файлов, ожидающих коммита
- git add . - сделать все измененные файлы готовыми для коммита.
- git commit -m "[descriptive message]" - записать изменения с заданным сообщением.
- git branch - список всех локальных веток в текущей директории.
- git checkout [branch-name] - переключиться на указанную ветку и обновить рабочую директорию.
- git merge [branch] — соединить изменения в текущей ветке с изменениями из заданной.
- git push - запустить текущую ветку в удаленную ветку.
- git pull - загрузить историю и изменения удаленной ветки и произвести слияние с текущей веткой.

8. Приведите примеры использования при работе с локальным и удалённым репозиториями.

- git remote add [имя] [url] — добавляет удалённый репозиторий с заданным именем;
- git remote remove [имя] — удаляет удалённый репозиторий с заданным именем;
- git remote rename [старое имя] [новое имя] — переименовывает удалённый репозиторий;
- git remote set-url [имя] [url] — присваивает репозиторию с именем новый адрес;

- `git remote show [имя]` — показывает информацию о репозитории.

9. Что такое и зачем могут быть нужны ветви (branches)?

Ветвление — это возможность работать над разными версиями проекта: вместо одного списка с упорядоченными коммитами история будет расходиться в определённых точках. Каждая ветвь содержит легковесный указатель HEAD на последний коммит, что позволяет без лишних затрат создать много веток. Ветка по умолчанию называется `master`, но лучше назвать её в соответствии с разрабатываемой в ней функциональностью.

10. Как и зачем можно игнорировать некоторые файлы при `commit`?

Зачастую нам не нужно, чтобы Git отслеживал все файлы в репозитории, потому что в их число могут входить:

Список литературы

1. Лекция Системы контроля версий
2. GitHub для начинающих