

Отчёт по лабораторной работе №2

Управление версиями

Григорьев Давид Андроникович НПМБД-01-21

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	10
4	Контрольные вопросы	11
	Список литературы	15

List of Figures

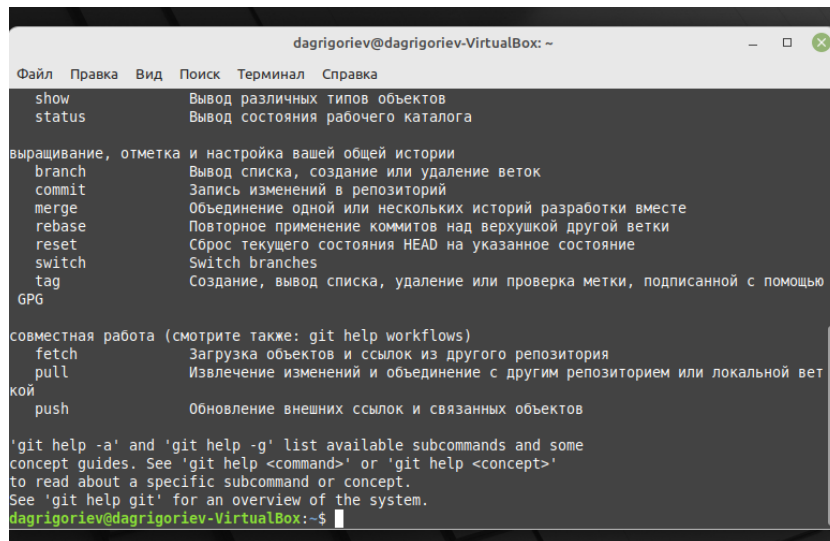
2.1	Загрузка пакетов	5
2.2	Параметры репозитория	5
2.3	rsa-4096	6
2.4	ed25519	6
2.5	GPG ключ	7
2.6	GPG ключ	7
2.7	Параметры репозитория	8
2.8	Связь репозитория с аккаунтом	8
2.9	Загрузка шаблона	9
2.10	Первый коммит	9

1 Цель работы

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

2 Выполнение лабораторной работы

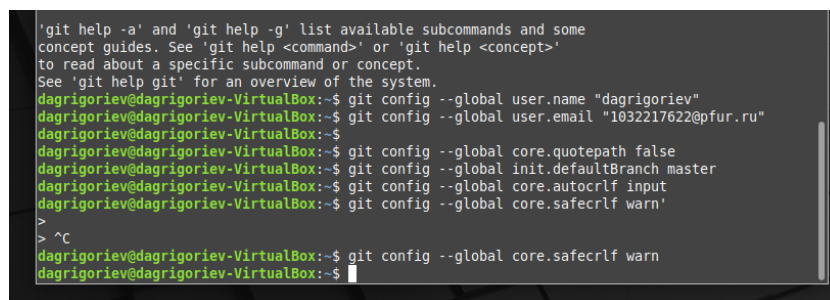
Устанавливаем git, git-flow и gh.



```
dagrigoriev@dagrigoriev-VirtualBox: ~  
Файл Правка Вид Поиск Терминал Справка  
show          Вывод различных типов объектов  
status        Вывод состояния рабочего каталога  
  
выращивание, отметка и настройка вашей общей истории  
branch        Вывод списка, создание или удаление веток  
commit        Запись изменений в репозиторий  
merge         Объединение одной или нескольких историй разработки вместе  
rebase        Повторное применение коммитов над верхушкой другой ветки  
reset         Сброс текущего состояния HEAD на указанное состояние  
switch        Switch branches  
tag           Создание, вывод списка, удаление или проверка метки, подписанной с помощью GPG  
  
совместная работа (смотрите также: git help workflows)  
fetch         Загрузка объектов и ссылок из другого репозитория  
pull          Извлечение изменений и объединение с другим репозиторием или локальной 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.  
dagrigoriev@dagrigoriev-VirtualBox:~$
```

Figure 2.1: Загрузка пакетов

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



```
'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.  
dagrigoriev@dagrigoriev-VirtualBox:~$ git config --global user.name "dagrigoriev"  
dagrigoriev@dagrigoriev-VirtualBox:~$ git config --global user.email "1032217622@pfur.ru"  
dagrigoriev@dagrigoriev-VirtualBox:~$  
dagrigoriev@dagrigoriev-VirtualBox:~$ git config --global core.quotepath false  
dagrigoriev@dagrigoriev-VirtualBox:~$ git config --global init.defaultBranch master  
dagrigoriev@dagrigoriev-VirtualBox:~$ git config --global core.autocrlf input  
dagrigoriev@dagrigoriev-VirtualBox:~$ git config --global core.safecrlf warn  
>  
> ^C  
dagrigoriev@dagrigoriev-VirtualBox:~$ git config --global core.safecrlf warn  
dagrigoriev@dagrigoriev-VirtualBox:~$
```

Figure 2.2: Параметры репозитория

Создаем SSH ключи

```

dagrigoriev@dagrigoriev-VirtualBox: ~
Файл  Правка  Вид  Поиск  Терминал  Справка
dagrigoriev@dagrigoriev-VirtualBox:~$ git config --global core.safecrlf warn
dagrigoriev@dagrigoriev-VirtualBox:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/dagrigoriev/.ssh/id_rsa):
Created directory '/home/dagrigoriev/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/dagrigoriev/.ssh/id_rsa
Your public key has been saved in /home/dagrigoriev/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:YHWPLQs2p3hVRxaUfjs+Yblu1Y4QPhB1lPJgg9rpYQ dagrigoriev@dagrigoriev-VirtualBox
The key's randomart image is:
+---[RSA 4096]---+
|          . . 0++O+*          |
|          . . . =B. +         |
|         o  ++B00.           |
|        . . .+. +. .         |
|       .So . + =             |
|      E o . = *o             |
|       . + 000=              |
|        + .=.                |
|         . O..               |
+-----[SHA256]-----+
dagrigoriev@dagrigoriev-VirtualBox:~$

```

Figure 2.3: rsa-4096

```

dagrigoriev@dagrigoriev-VirtualBox: ~
Файл  Правка  Вид  Поиск  Терминал  Справка
|          . . 0..           |
+-----[SHA256]-----+
dagrigoriev@dagrigoriev-VirtualBox:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/dagrigoriev/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/dagrigoriev/.ssh/id_ed25519
Your public key has been saved in /home/dagrigoriev/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:THaA42e80Tfqf4A7FbrsuSB0gl9MHZn03nwTGdL+aU dagrigoriev@dagrigoriev-VirtualBox
The key's randomart image is:
+---[ED25519 256]---+
|          . .              |
|         o .               |
|        . . =0 . .         |
|       . . +.oB. = .        |
|      . +00SX + o o         |
|     . .o.oB.* . E         |
|    o.. = o* .              |
|   ...+.o.o                 |
|  .. o=*o.                  |
+-----[SHA256]-----+
dagrigoriev@dagrigoriev-VirtualBox:~$

```

Figure 2.4: ed25519

Создаем GPG ключ

```
dagrigoriev@dagrigoriev-VirtualBox: ~
Файл Правка Вид Поиск Терминал Справка
"dagrigoriev <1032217622@pfur.ru>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? O
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
gpg: /home/dagrigoriev/.gnupg/trustdb.gpg: создана таблица доверия
gpg: ключ 442027204FA4B0F6 помечен как абсолютно доверенный
gpg: создан каталог '/home/dagrigoriev/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/dagrigoriev/.gnupg/openpgp-revocs.d/4A96E606BFF260642BF
5758C442027204FA4B0F6.rev'.
открытый и секретный ключи созданы и подписаны.

pub  rsa4096 2022-06-12 [SC]
     4A96E606BFF260642BF5758C442027204FA4B0F6
uid                               dagrigoriev <1032217622@pfur.ru>
sub   rsa4096 2022-06-12 [E]

dagrigoriev@dagrigoriev-VirtualBox:~$
```

Figure 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

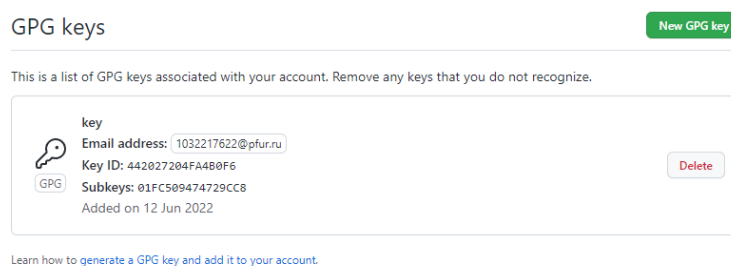


Figure 2.6: GPG ключ

Настройка автоматических подписей коммитов git

```
dagrigoriev@dagrigoriev-VirtualBox: ~
Файл Правка Вид Поиск Терминал Справка
MbuzED2bKX4PhG8Thk9zDN6a1mDNkaIjyEU3uKWvE4vnYiQWHu7r9LrC9cPw03Q+
k2Vw/dHptI/p5T6zUyFz/sVuAlgyslaLq2uXQ6s3gEnL3s6EpcZF2446uqBy/wAR
AQABiQI2BBg8CgAgFIEESpbmBr/yYG0r9XWmRCAnIE+ksPYFAMkLk24CGwwACgkQ
RCAnIE+ksPaXnw//V38jv0G4zWKREm+tvJ3HScMFpi0vRqgEKM6V4cLIEeGAa0yf
L3wvKcrw6fAaA7DSG3VwPLMDfFqb5YkF9ih/ieqo0jbNcadWBL5/N0znlx7NI/W
wRpABz6WgDqD1t/yWvXgnaLTrINTPUIiVHwycx28Xv0v6d4fE0J60pM01TL8rESZ
gHwuo9Z+40Hg3DL2Yopmy4M0HJbJgm/M+mrhY7DlNmAhbx4bgzcv3lFt6gW0Zheq
3ysH99yulvj cAb/ckX4T9U8vMT251AWig7dCKhxAQ1fhzdknzXWhd9rZvpyo7Go3
CZQjmmMq5NxGQUMgWwD2llrKI6yMVoJ+3M12R25FCJGw517LLCgr+aItmRChFewt
96QwpTXAFu3nhs+yAmMV8K62pWp7M1Bmgue4+Eh/uw+W6QJz+vPUFA6f97MNFcke
MF52+Aj8ug9wmSXUSJGvj3yM60LNVN61SuQishfCM9fICcbC4HxQ1FQBHoAPXeQr
HuInjAyJM6C03x54bBuCZKSZb590zHga6ZRLkQW3Fu0D8PLI30LkZaA0jqRZhd2x
3LAAQfsN0P2wW3IIoTnzT4ha6JULKnXkUvStD10QdtLPEsS/15mBzj+xn+rEVn5g
Ii2m1Jq78MK5/tM+fjb0xktCfmPVRCKAj+Oxre0y2WnYTS14JBdRrDPTxs=
=6pYV
-----END PGP PUBLIC KEY BLOCK-----
dagrigoriev@dagrigoriev-VirtualBox:~$
dagrigoriev@dagrigoriev-VirtualBox:~$
dagrigoriev@dagrigoriev-VirtualBox:~$
dagrigoriev@dagrigoriev-VirtualBox:~$ git config --global user.signingkey 442027204FA4B0F6
dagrigoriev@dagrigoriev-VirtualBox:~$ git config --global commit.gpgsign true
dagrigoriev@dagrigoriev-VirtualBox:~$ git config --global gpg.program $(which gpg2)
dagrigoriev@dagrigoriev-VirtualBox:~$
dagrigoriev@dagrigoriev-VirtualBox:~$
```

Figure 2.7: Параметры репозитория

Настройка gh

```
dagrigoriev@dagrigoriev-VirtualBox: ~
Файл Правка Вид Поиск Терминал Справка
dagrigoriev@dagrigoriev-VirtualBox:~$
dagrigoriev@dagrigoriev-VirtualBox:~$ git config --global user.signingkey 442027204FA4B0F6
dagrigoriev@dagrigoriev-VirtualBox:~$ git config --global commit.gpgsign true
dagrigoriev@dagrigoriev-VirtualBox:~$ git config --global gpg.program $(which gpg2)
dagrigoriev@dagrigoriev-VirtualBox:~$
dagrigoriev@dagrigoriev-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/dagrigoriev/.ssh/id\_rsa.pub
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 9F3B-49B7
Press Enter to open github.com in your browser...
- Authentication complete.
- gh config set -h github.com git_protocol ssh
- Configured git protocol
- Uploaded the SSH key to your GitHub account: /home/dagrigoriev/.ssh/id\_rsa.pub
- Logged in as dagrigoriev

A new release of gh is available: 2.9.0 → v2.12.1
https://github.com/cli/cli/releases/tag/v2.12.1
dagrigoriev@dagrigoriev-VirtualBox:~$
```

Figure 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация


```
dagrigoriev@dagrigoriev-VirtualBox: ~/work/study/2021-2022/Операционные системы
Файл Правка Вид Поиск Терминал Справка
remote: Total 20 (delta 2), reused 15 (delta 2), pack-reused 0
Получение объектов: 100% (20/20), 12.49 КиБ | 4.16 МБ/с, готово.
Определение изменений: 100% (2/2), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/dagrigoriev/work/study/2021-2022/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 42, done.
remote: Counting objects: 100% (42/42), done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 42 (delta 9), reused 40 (delta 7), pack-reused 0
Клонирование в «/home/dagrigoriev/work/study/2021-2022/Операционные системы/os-intro/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
Подмодуль по пути «template/presentation»: забрано состояние «3eae7b7586f8a9aded2b506cd1018e625b228b93a»
Подмодуль по пути «template/report»: забрано состояние «df7b2ef80f8def3b9a496f8695277469a1a7842a»
dagrigoriev@dagrigoriev-VirtualBox:~/work/study/2021-2022/Операционные системы$
```

Figure 2.9: Загрузка шаблона

Подготовка репозитория и коммит изменений

```
dagrigoriev@dagrigoriev-VirtualBox: ~/work/study/2021-2022/Операционные системы/os-intro
Файл Правка Вид Поиск Терминал Справка
create mode 100644 labs/lab{01..15}/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab{01..15}/report/report.md
delete mode 100644 package.json
create mode 100644 project-personal/stage{1..6}/presentation/Makefile
create mode 100644 project-personal/stage{1..6}/presentation/presentation.md
create mode 100644 project-personal/stage{1..6}/report/Makefile
create mode 100644 project-personal/stage{1..6}/report/bib/cite.bib
create mode 100644 project-personal/stage{1..6}/report/image/placeholder.jpg
create mode 100644 project-personal/stage{1..6}/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 project-personal/stage{1..6}/report/report.md
create mode 100644 structure
dagrigoriev@dagrigoriev-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ git push
Перечисление объектов: 20, готово.
Подсчет объектов: 100% (20/20), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (14/14), готово.
Запись объектов: 100% (19/19), 266.47 КиБ | 3.06 МБ/с, готово.
Всего 19 (изменения 2), повторно использовано 0 (изменения 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:dagrigoriev/study_2021-2022_os-intro.git
  362a66a..4979f7b master -> master
dagrigoriev@dagrigoriev-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