

Отчёт по лабораторной работе №2

Управление версиями

Юсуф Абдулаев

Содержание

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

Список иллюстраций

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.

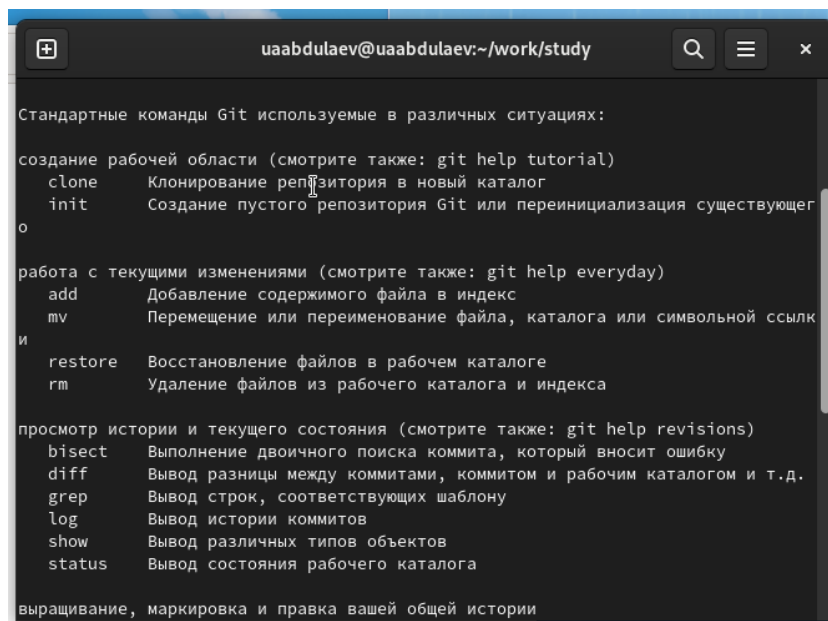


Рис. 2.1: Загрузка пакетов

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

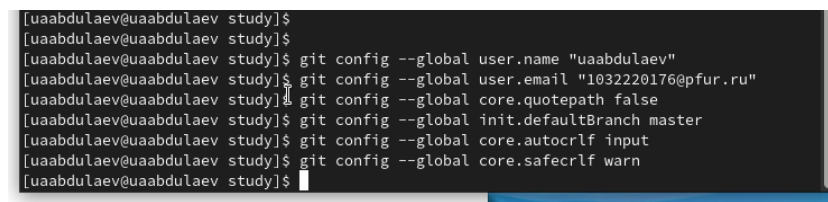
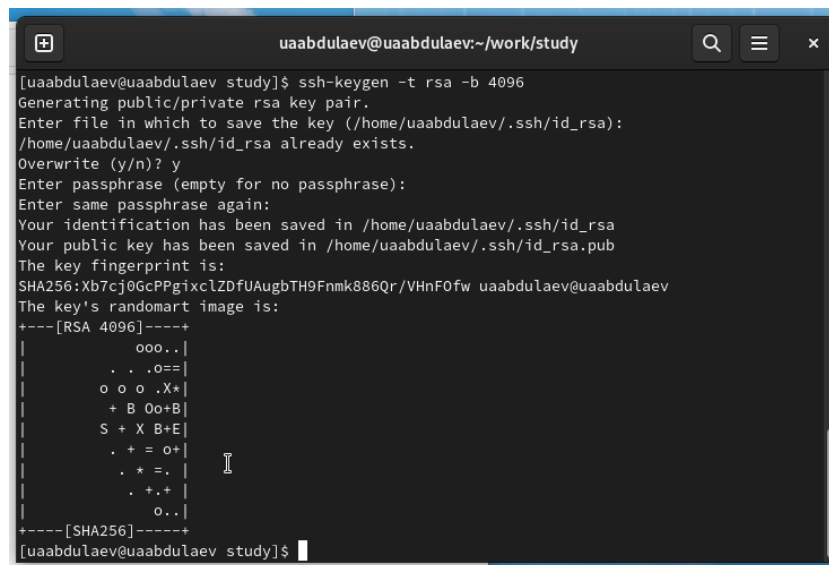


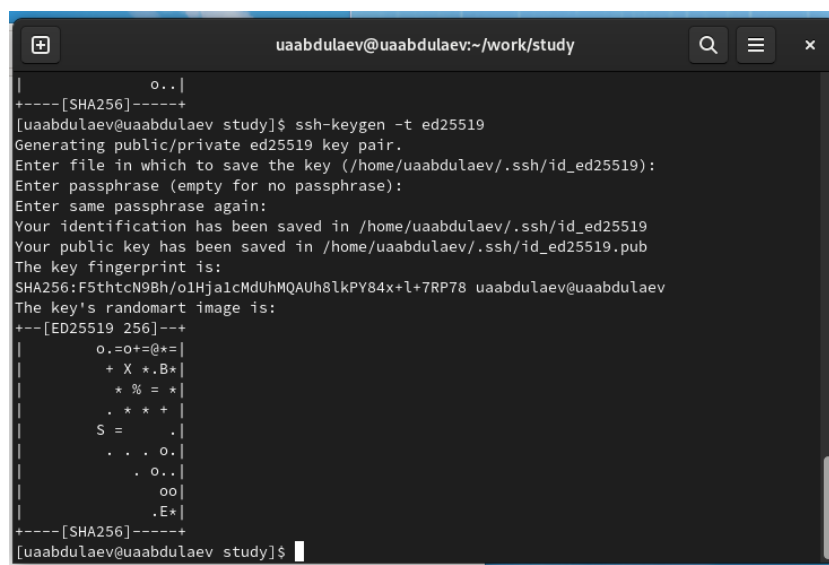
Рис. 2.2: Параметры репозитория

Создаем SSH ключи



```
uaabdulaev@uaabdulaev:~/work/study
[uaabdulaev@uaabdulaev study]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/uaabdulaev/.ssh/id_rsa):
/home/uaabdulaev/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/uaabdulaev/.ssh/id_rsa
Your public key has been saved in /home/uaabdulaev/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Xb7cj0GcPPgiXclZDfUAugbTH9Fnmk886Qr/VHnF0fw uaabdulaev@uaabdulaev
The key's randomart image is:
+----[RSA 4096]-----+
|
| ooo..|
| . . .O=|
| o o o .X*|
| + B Oo+B|
| S + X B+E|
| . + = O+|
| . * =. |
| . +. |
| o..|
+-----[SHA256]-----+
[uaabdulaev@uaabdulaev study]$
```

Рис. 2.3: rsa-4096



```
uaabdulaev@uaabdulaev:~/work/study
[uaabdulaev@uaabdulaev study]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/uaabdulaev/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/uaabdulaev/.ssh/id_ed25519
Your public key has been saved in /home/uaabdulaev/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:F5thtcN9Bh/o1Hja1cMdUhmQAUh8lkPY84x+l+7RP78 uaabdulaev@uaabdulaev
The key's randomart image is:
+--[ED25519 256]--+
|
| o..o+=@*+=|
| + X *.B*|
| * % = *|
| . * * + |
| S = .|
| . . .O.|
| . o..|
| oo|
| .E*|
+-----[SHA256]-----+
[uaabdulaev@uaabdulaev study]$
```

Рис. 2.4: ed25519

Создаем GPG ключ

```
uaabdulaev@uaabdulaev:~/work/study
Вы выбрали следующий идентификатор пользователя:
"uaabdulaev <1032220176@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/uaabdulaev/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/home/uaabdulaev/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/uaabdulaev/.gnupg/openpgp-revocs.d/F4F312CBC0F967
2F971152E55449C2F84591768C.rev'.
открытый и секретный ключи созданы и подписаны.

pub  rsa4096 2023-06-09 [SC]
     F4F312CBC0F9672F971152E55449C2F84591768C
uid          uaabdulaev <1032220176@pfur.ru>
sub  rsa4096 2023-06-09 [E]

[uaabdulaev@uaabdulaev study]$
```

Рис. 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

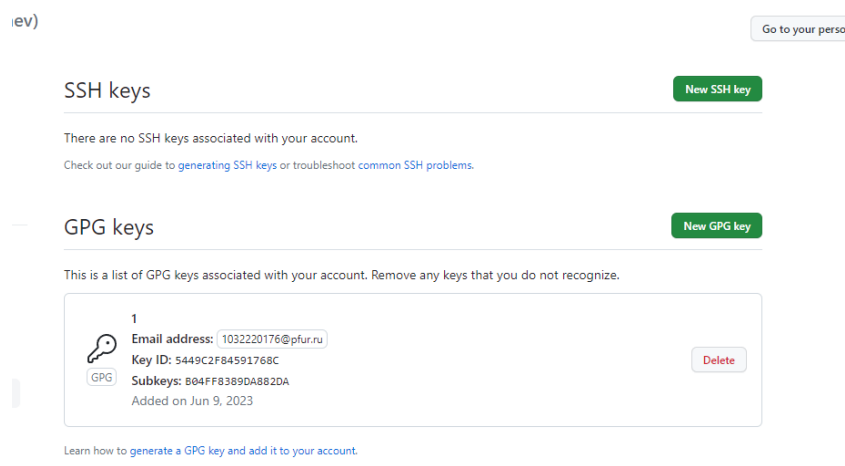


Рис. 2.6: GPG ключ

Настройка автоматических подписей коммитов git

```
uaabdulaev@uaabdulaev:~/work/study
70UJeJ2k43HbLcHx78C0EKA+hPybZbDF1ubfekpMCK11vu+nm1tpmMRXBQe6cZB0
AE2ZN2WnBF3mXBE6jBxJbXfNdqs1znSkQyvumBY+sWwFRf+RJUQg7GnVYojImDG
H0yKypXIIHtZ0csNBruThVsBNTPrpg4JSkq5oFSDiaWaC/YD47YfsxzyJRW57nW
bRAU3Gjie5Akj19s8oG53oVMCFVmgEK0mz64Qg656e8Bj7vUXjo8KuMpluVkk2ah
ABEBAAGJAjYEGAEIACAWIQT08xLLwPlnL5cRUuVUScL4RZF2jAUCZIL7EQIbDAAK
CRBUScL4RZF2jCM5D/41x0sLhPGHoSRWzQUr87SVBWARFAZ414Eg6gehPg1hdlu
KSwoDf21mHgI+I+ipmxukJfXhtpUfjBnXSDWZaglhg0o3HZ94h+IKUPvy950g/By
BSRqU3E8xpTB0/OUuCIVRU+UUR6iZxW6MsUZ0UcIHh3fdJrtJb8wftXwd8qaZqFL
UoD28Nz9/8UBxgFPiH3G0CKRyeAbjmhXSSa8GuqbcMQk0NuSySs4FMG3N0i/sRXm
ugf/VqFMu25zu8cKBTJQnxYb9MA6o085H+gkWONKhQ+EBBZML9qiJQ3dC+QmhdCY
QFWYvg1fnBx3hzBQrIr9ZAQWd+Wav+6u0FrOqEaAEZYGM/AOpB92q9poVM2PrJeb
yZ3HwdIn1HuQY1h3xoGw3jyEpXu6bIRZv4r5Ko6otxNHkbrsySiGH0Fc+R3Hk1D
+EMTXPyuz9s0FfKH0b6FaGzVu9DFP0xrw0LkTyPAD1W1Dg103TN2nsjstzw4LeQ
Ig2Tib5QyViHAVNFdfj/BKgv5hArg7x0dvLhqq8Hk6XpCk3XvDcX4PhgTd7eVmBp
8+YUsEr9l8qT+CnAfUZsfzfi43cy0P9IkQ72mjQe7sDEUY4ujKbC4ZnSczmRfa+2r
mfQUCjggkR/hfLBFSWmX+ae8heK3phCf/JwwYD4GRFsamcKvKH6NhwH8GXKA==
=b2Lt
-----END PGP PUBLIC KEY BLOCK-----
[uaabdulaev@uaabdulaev study]$
[uaabdulaev@uaabdulaev study]$
[uaabdulaev@uaabdulaev study]$ git config --global user.signingkey 5449C2F84591768C
[uaabdulaev@uaabdulaev study]$ git config --global commit.gpgsign true
[uaabdulaev@uaabdulaev study]$ git config --global gpg.program $(which gpg2)
[uaabdulaev@uaabdulaev study]$
```

Рис. 2.7: Параметры репозитория

Настройка gh

```
[uaabdulaev@uaabdulaev study]$ git config --global gpg.program $(which gpg2)
[uaabdulaev@uaabdulaev study]$ 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/uaabdulaev/.ssh/id_rsa.pub
? Title for your SSH key: GitHub CLI
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 0C43-A609
Press Enter to open github.com in your browser...
[GFx1-]: glxtest: VA-API test failed: failed to initialise VAAPI connection.
[GFx1-]: Unrecognized feature ACCELERATED_CANVAS2D
[2023-06-09T10:14:35Z ERROR glean_core::metrics::ping] Invalid reason code startup for pi
ng background-update
✓ Authentication complete.
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Uploaded the SSH key to your GitHub account: /home/uaabdulaev/.ssh/id_rsa.pub
✓ Logged in as uaabdulaev
[uaabdulaev@uaabdulaev study]$
```

Рис. 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация


```
uaabdulaev@uaabdulaev:~/work/study/2022-2023/Операционные системы
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-ma
rkdwn-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-tem
plate.git) зарегистрирован по пути «template/report»
Клонирование в «/home/uaabdulaev/work/study/2022-2023/Операционные системы/os-intro/templ
ate/presentation»...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
Получение объектов: 100% (82/82), 92.90 КиБ | 2.21 МБ/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/home/uaabdulaev/work/study/2022-2023/Операционные системы/os-intro/templ
ate/report»...
remote: Enumerating objects: 101, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 101 (delta 40), reused 88 (delta 27), pack-reused 0
Получение объектов: 100% (101/101), 327.25 КиБ | 3.41 МБ/с, готово.
Определение изменений: 100% (40/40), готово.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5809264cb755d316174540b
753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3aef11a33b1e3b2'
[uaabdulaev@uaabdulaev Операционные системы]$
```

Рис. 2.9: Загрузка шаблона

Подготовка репозитория и коммит изменений

```
uaabdulaev@uaabdulaev:~/work/study/2022-2023/Операционные систе...
create mode 100644 project-personal/stage6/report/pandoc/csl/gost-r-7-0-5-2008-numeric.c
sl
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_fignos.py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_secnos.py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/core.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/main.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/pandocattrib
utes.py
create mode 100644 project-personal/stage6/report/report.md
[uaabdulaev@uaabdulaev os-intro]$ git push
Перечисление объектов: 38, готово.
Подсчет объектов: 100% (38/38), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (37/37), 343.00 КиБ | 3.30 МБ/с, готово.
Всего 37 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано паке
тов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:uaabdulaev/os-intro.git
7dfd129..b308da1 master -> master
[uaabdulaev@uaabdulaev os-intro]$
```

Рис. 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: