

Отчёт по лабораторной работе №2

Управление версиями

Мандрик Анастасия 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.

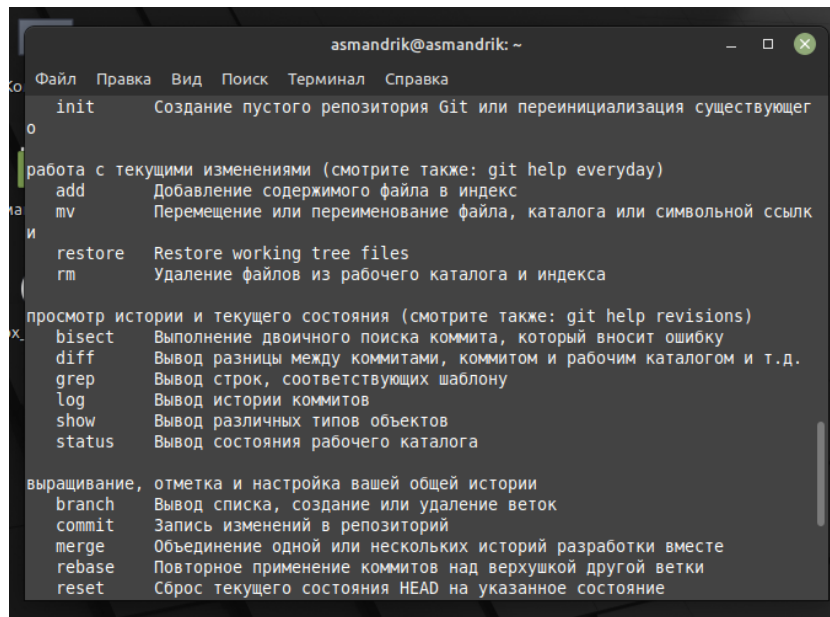


Figure 2.1: Загрузка пакетов

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

```
asmandrik@asmandrik: ~
Файл Правка Вид Поиск Терминал Справка
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.
asmandrik@asmandrik:~$ git config --global user.name "asmandrik"
asmandrik@asmandrik:~$ git config --global user.email "1032210088@pfur.ru"
asmandrik@asmandrik:~$ git config --global core.quotepath false
asmandrik@asmandrik:~$ git config --global init.defaultBranch master
asmandrik@asmandrik:~$ git config --global core.autocrlf input
asmandrik@asmandrik:~$ git config --global core.safecrlf warn
asmandrik@asmandrik:~$
```

Figure 2.2: Параметры репозитория

Создаем SSH ключи

```
asmandrik@asmandrik: ~
Файл Правка Вид Поиск Терминал Справка
asmandrik@asmandrik:~$
asmandrik@asmandrik:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/asmandrik/.ssh/id_rsa):
Created directory '/home/asmandrik/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/asmandrik/.ssh/id_rsa
Your public key has been saved in /home/asmandrik/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:k+6Z9st0V2tWmV4He2qiacNdHBSw33fvEiD8rEcYMM asmandrik@asmandrik
The key's randomart image is:
+----[RSA 4096]-----+
|      .  .  .  .  .  |
|      E    +  +  |
|      . o  ...++ |
|      o + 000* |
|      S + =.=+* |
|      . . + 0+*o |
|      ....0.*. |
|      ..=++.= |
|      .++Bo |
+----[SHA256]-----+
asmandrik@asmandrik:~$
```

Figure 2.3: rsa-4096

```
asmandrik@asmandrik: ~  
Файл Правка Вид Поиск Терминал Справка  
+-----[SHA256]-----+  
asmandrik@asmandrik:~$ ssh-keygen -t ed25519  
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/home/asmandrik/.ssh/id_ed25519):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/asmandrik/.ssh/id_ed25519  
Your public key has been saved in /home/asmandrik/.ssh/id_ed25519.pub  
The key fingerprint is:  
SHA256:D3SLNcighLcIPJD5e1IexXipiHweJ0Hfv0PE07W2MUK asmandrik@asmandrik  
The key's randomart image is:  
+--[ED25519 256]--+  
|+...+o.|  
|=+oo=.+o . E  
|o++B.=. = * o  
|..*o* = = 0  
|o+... S o +  
| . . . 0 .  
| . . .  
| . . .  
| . . .  
| . . .  
+-----[SHA256]-----+  
asmandrik@asmandrik:~$
```

Figure 2.4: ed25519

Создаем GPG ключ

```
asmandrik@asmandrik: ~  
Файл Правка Вид Поиск Терминал Справка  
"asmandrik <1032210088@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/asmandrik/.gnupg/trustdb.gpg: создана таблица доверия  
gpg: ключ 76BCAE5CB05C041D помечен как абсолютно доверенный  
gpg: создан каталог '/home/asmandrik/.gnupg/openpgp-revocs.d'  
gpg: сертификат отзыва записан в '/home/asmandrik/.gnupg/openpgp-revocs.d/AB046F  
1F15DFA0CBE8CDFE6776BCAE5CB05C041D.rev'.  
открытый и секретный ключи созданы и подписаны.  
  
pub   rsa4096 2022-08-23 [SC]  
      AB046F1F15DFA0CBE8CDFE6776BCAE5CB05C041D  
uid    asmandrik <1032210088@pfur.ru>  
sub    rsa4096 2022-08-23 [E]  
asmandrik@asmandrik:~$
```

Figure 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

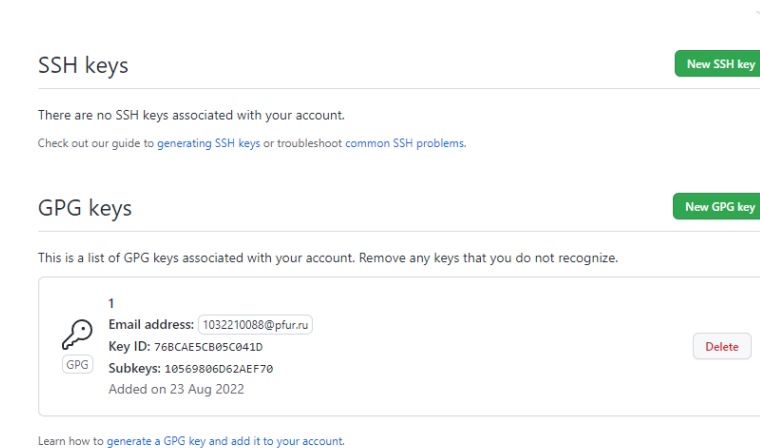


Figure 2.6: GPG ключ

Настройка автоматических подписей коммитов git

```
asmandrik@asmandrik: ~  
Файл Правка Вид Поиск Терминал Справка  
RjdyX3oD9vCo7Is+IksB5QE5WNxmHLRh640oGtghl08vCdcCX/bebdyHui5YJvMY  
fTKQqMxXaFChnLXlrRfS+LfcVwE/uGNpcRjILL6PbePvjPbw1F7Dygh5j7T+D9E  
dcIuSNXNTnzSUzbytv9tKhdRD9h/86hiUJl2D//tUfF6Iu+NsdC3RcnLBxNiFMvi  
BeNkQ4ZKFEIfvNi6NqSI2rum0z23xwA13iz3Uk3ared7biXVarZ/Ul0+sZ8AEQEA  
AYkCNQYQAQoAIBYhBKsEbx8V36DL6M3+Z3a8rlywXAQd8QJjBLk5AhsMAAoJEHa8  
rlywXAQdDwQALfL8B+zLdaiJpERv4gwIaIt0IqWguCTgfMBLTLAaJ0T0eGsRP  
+nIkGoRrTDkcTGQhUVFEQvPPdp0XzqyZEVvykXNwJg3f4cF7eHSfJZotpDM9MT0+  
/CwAlq9nHnadYba206IRDLSXeA5qAispf2SxbzbzoL00Hbh0Z00n/8+snpc7b0aF  
ubAYvn8fd6kscsyI+P1UyMiQ7330Vmb1Latngf4/MsvLCxJhZPbLTdDrDArrvIkp  
sf09Bj2zS0ph8z0fGRIKM1QvYJ3mtf+Zi4F3L8AhPiIHn82Nf8A7ij2zo88hJ8uu  
mLjtwyA5LsYn4YHfsc8wa/6C8C3090/u/gLDG3JdI88+QoUYFeynkYi/2Sv7dDrP  
Bvy1Vv9HeJXr7BziZ4G8hCR9+/F0L0F06Yx0E3024tlf76eQy3m0v8L0kMKiLM6C  
2qUDzYnMhEiXtTjDeBnBmjjj0m3pajyPIC/u/LptCncfMCchLksyyKQ00CtLLAEb  
8hgdUNUNInx+funWYRjy/Ucg/jPoINqFqPEjcbnfHFuJF//QwuL+J0RxfnugcqP  
zg8u6GAt0/00VS2GoI9sIuYETxT0y0+m1FXkwWT7PnHxsg48n1LZA4jLXup2caBg  
jmmYNAbsXuDRqZrKkirPuWxkvoGstjUc+UIbEr/DZSKf+CwzxCduCY  
=PTYh  
-----END PGP PUBLIC KEY BLOCK-----  
asmandrik@asmandrik:~$  
asmandrik@asmandrik:~$  
asmandrik@asmandrik:~$ git config --global user.signingkey 76BCAE5CB05C041D  
asmandrik@asmandrik:~$ git config --global commit.gpgsign true  
asmandrik@asmandrik:~$ git config --global gpg.program $(which gpg2)  
asmandrik@asmandrik:~$
```

Figure 2.7: Параметры репозитория

Настройка gh


```
asmandrik@asmandrik: ~  
Файл Правка Вид Поиск Терминал Справка  
Выбор ранее не выбранного пакета gh.  
(Чтение базы данных ... на данный момент установлен 284941 файл и каталог.)  
Подготовка к распаковке .../archives/gh_2.14.4_amd64.deb ...  
Распаковывается gh (2.14.4) ...  
Настраивается пакет gh (2.14.4) ...  
Обрабатываются триггеры для man-db (2.10.2-1) ...  
asmandrik@asmandrik:~$  
asmandrik@asmandrik:~$ 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/asmandrik/.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: 2BDA-F42F  
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/asmandrik/.ssh/id_rsa.pub  
✓ Logged in as asmandrik  
asmandrik@asmandrik:~$ 2022-08-23T11:30:23Z ERROR viaduct::backend::ffi Missin  
g HTTP status
```

Figure 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация

```
asmandrik@asmandrik: ~/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  
report-template.git) зарегистрирован по пути «template/report»  
Клонирование в «/home/asmandrik/work/study/2021-2022/Операционные системы/os-int  
ro/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  
Получение объектов: 100% (42/42), 31.19 КиБ | 679.00 КиБ/с, готово.  
Определение изменений: 100% (9/9), готово.  
Клонирование в «/home/asmandrik/work/study/2021-2022/Операционные системы/os-int  
ro/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 | 3.21 МБ/с, готово.  
Определение изменений: 100% (31/31), готово.  
Submodule path 'template/presentation': checked out '3eae6b7586f8a9aded2b506cd10  
18e625b228b93'  
Submodule path 'template/report': checked out 'df7b2ef80f8def3b9a496f8695277469a  
1a7842a'  
asmandrik@asmandrik:~/work/study/2021-2022/Операционные системы$
```

Figure 2.9: Загрузка шаблона

Подготовка репозитория и коммит изменений

```
asmandrik@asmandrik: ~/work/study/2021-2022/Операционные системы/os-intro
Файл Правка Вид Поиск Терминал Справка
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/placeimg_800_600_t
ech.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
asmandrik@asmandrik:~/work/study/2021-2022/Операционные системы/os-intro$ git pu
sh
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), 266.46 КиБ | 2.26 МиБ/с, готово.
Всего 19 (изменений 2), повторно использовано 0 (изменений 0), повторно использо
вано пакетов 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:asmandrik/study_2021-2022_os-intro.git
41c24f8..0daf133 master -> master
asmandrik@asmandrik:~/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