

Отчёт по лабораторной работе №2

Управление версиями

Чермашенцев Павел Андреевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Вывод	17
4	Контрольные вопросы	18

Список иллюстраций

2.1	Загрузка пакетов	7
2.2	Параметры репозитория	8
2.3	rsa-4096	9
2.4	ed25519	10
2.5	GPG ключ	11
2.6	GPG ключ	12
2.7	Параметры репозитория	13
2.8	Связь репозитория с аккаунтом	14
2.9	Загрузка шаблона	15
2.10	Первый коммит	16

Список таблиц

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.

```

pachermashentsev@pachermashentsev:~$ git
использование: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
                [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
                [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy-fetch]
                [--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]
                [--work-tree=<path>] [--namespace=<name>] [--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 существующего

работа с текущими изменениями (смотрите также: git help everyday)
add        Добавление содержимого файла в индекс
mv         Перемещение или переименование файла, каталога или символической ссылки
restore    Восстановление файлов в рабочем каталоге
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
diff       Вывод разницы между коммитами, коммитом и рабочим каталогом и т.д.
grep       Вывод строк, соответствующих шаблону
log        Вывод истории коммитов
show       Вывод различных типов объектов
status     Вывод состояния рабочего каталога

```

Рис. 2.1: Загрузка пакетов

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

```
pachermashentsev@pachermashentsev:~$  
pachermashentsev@pachermashentsev:~$ git config --global user.name "palcher1212"  
pachermashentsev@pachermashentsev:~$ git config --global user.email "1132246450@rudn.university"  
pachermashentsev@pachermashentsev:~$ git config --global core.quotepath false  
pachermashentsev@pachermashentsev:~$ git config --global init.defaultBranch master  
pachermashentsev@pachermashentsev:~$ git config --global core.autocrlf input  
pachermashentsev@pachermashentsev:~$ git config --global core.safecrlf warn  
pachermashentsev@pachermashentsev:~$  
pachermashentsev@pachermashentsev:~$
```

Рис. 2.2: Параметры репозитория

Создаем SSH ключи


```

pachermashentsev@pachermashentsev:~$
pachermashentsev@pachermashentsev:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pachermashentsev/.ssh/id_rsa):
Created directory '/home/pachermashentsev/.ssh'.
Enter passphrase for "/home/pachermashentsev/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/pachermashentsev/.ssh/id_rsa
Your public key has been saved in /home/pachermashentsev/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:jHQhBfhjzP7dflj2Jx2rt3iZn2w0hTRTRID0cy3eQ8 pachermashentsev@pachermashentsev
The key's randomart image is:
+---[RSA 4096]-----+
|  .000 ....++ |
|  . . . . 0 |
|  +. .   B + |
|  .*+   . 0 o|
|  o..S   .E+.|
|  .      +00|
|  . . . = 0*|
|  . . 0 *B=|
|  . .+**=|
+----[SHA256]-----+
pachermashentsev@pachermashentsev:~$

```

Рис. 2.3: rsa-4096

```

pachermashentsev@pachermashentsev:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/pachermashentsev/.ssh/id_ed25519):
Enter passphrase for "/home/pachermashentsev/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/pachermashentsev/.ssh/id_ed25519
Your public key has been saved in /home/pachermashentsev/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:GM/nFQUBdUvLBbDdwGZHLGnUmwCMmxuT5ECepTXCbcY pachermashentsev@pachermashentsev
The key's randomart image is:
+--[ED25519 256]--+
|      ooo+***=B*O|
|    .o+E.. B+O+|
|    .+* + oo=. =|
|      = B  . o |
|    . S = . |
|      + . |
|      . |
|      | |
+----[SHA256]-----+
pachermashentsev@pachermashentsev:~$

```

Рис. 2.4: ed25519

Создаем GPG ключ

```
pachermashentsev@pachermashentsev:~$  
pachermashentsev@pachermashentsev:~$ gpg --full-generate-key  
gpg (GnuPG) 2.4.5; Copyright (C) 2024 g10 Code GmbH  
This is free software: you are free to change and redistribute it.  
There is NO WARRANTY, to the extent permitted by law.  
  
gpg: создан каталог '/home/pachermashentsev/.gnupg'  
Выберите тип ключа:  
  (1) RSA and RSA  
  (2) DSA and Elgamal  
  (3) DSA (sign only)  
  (4) RSA (sign only)  
  (9) ECC (sign and encrypt) *default*  
 (10) ECC (только для подписи)  
 (14) Existing key from card  
Ваш выбор? 1  
длина ключей RSA может быть от 1024 до 4096.  
Какой размер ключа Вам необходим? (3072) 4096  
Запрошенный размер ключа - 4096 бит  
Выберите срок действия ключа.  
  0 = не ограничен  
  <n> = срок действия ключа - n дней  
  <n>w = срок действия ключа - n недель  
  <n>m = срок действия ключа - n месяцев  
  <n>y = срок действия ключа - n лет  
Срок действия ключа? (0) 0  
Срок действия ключа не ограничен  
Все верно? (y/N) y  
  
GnuPG должен составить идентификатор пользователя для идентификации ключа.  
  
Ваше полное имя: palcher1212  
Адрес электронной почты: 1132246450@rudn.university  
Примечание:  
Вы выбрали следующий идентификатор пользователя:  
  "palcher1212 <1132246450@rudn.university>"  
  
Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? O  
Необходимо получить много случайных чисел. Желательно, чтобы Вы
```

Рис. 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

```

pachermashentsev@pachermashentsev:~$ 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
[keyboard]
-----
sec   rsa4096/76C04DF9E69BCBFD 2025-02-24 [SC]
      0F4443D443FCD1192988060E76C04DF9E69BCBFD
uid    [ абсолютно ] palcher1212 <1132246450@rudn.university>
ssb    rsa4096/D0F5116A61BF1A48 2025-02-24 [E]

pachermashentsev@pachermashentsev:~$ 
pachermashentsev@pachermashentsev:~$ 
pachermashentsev@pachermashentsev:~$ gpg --armor --export 76C04DF9E69BCBFD
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGe8IdABEAC4WHPbKYDy+m/xus9tn40YeJ3RyiG8A54FUNu2I/kxH70DZ09C
eDphn7rbPUAinSZSt+LpANxRYtc4CJJp0gyIK0hAF0Lxppi57vma3En2dIU4WZMK
tux0RFzXIdCEeFr4Bu1MTK0aNeLv7R+6FhNqYoCAsuH879S01M9QDrKyCK1b0J
iukNjXyLZ8iNysgSe6Qy0H+js0p1YGPFeWkt+TeB7XPrIj1fjpRBP6AY8bLgVQqE
+Xodv4k7wkmBg2KxR1lEym5yco6txvDu1906VvyepB9Kka65glrbwJkHvi+B3Qz9
ED0v7DAwwIB9yRFHSMk6ar0FUBn9D+0aszhd8JwFxVQqm0lRniKWY41/0dn3mKo
Nrd1SdWGUZol/ucNcRU6v/XxoeW00S/zRhp/BnIuCU9qDmWxjrjf8VWy/Ty0iZKi
S/I5SjujT207UtUixs7EsR9g8RlNe8nSFq/AbfIm01D7IBEYcEYnZRQuUoS+hlhg
GSD+ZkqVX3KzphCRFdnkz4WwMzP0d1Z+WjHimgGKoXg4Qnrhik2gGI5ax4+KcLjT
+BtLYC3luFE3gGycH2gI31jzQwpARYDNhvnYZ1wwqk1ogDSe0yxwBw67Iz5pWgNj
a47L4/jZvp/r39RrS+30sRVG/13aT/1/fTAxmAAjwwZaJo9lMHhCGc2htwARAQAB
tChwYVwxjaGVyMTIxMiA8MTEzMjI0NjQ1MEBydWRuLnVuaXZlcuNpdHk+iQJRBMBM
CAA7FiEED0RD1EP80RkpiAYOdsBN+eaby/0FAme8IdACGwMFCwkIBwICIgIGFQoJ
CAAsCBByCAwEChgcCF4AACgkQdsBN+eaby/26jxAArmeVGcMBQyeHfdm9JyC0lEa+
p7Vk+KFyUwcsaEUUVfVHxgRcuH3trQ9oe3BA5nPIh5RHG+apHBtxMrV/tWr+dx/LJ
ZrtrRVfwqipHK+ajQaCRopOd2Q/gMj9loy00jcvb5wBbjlHjrpmxIy3juwPNFg81P
lKR9RbSS5ihZmzE1X3nBp6dgXyZ89oIynLF76UdtX+PsfeKQ7K8Ay+RakhyKv/
zPKt+MT0z0zS80lE] 3i4ncF8E+8ikYhik3BYvVTouHnDMhtdhXgCCTHavoE7ace0

```

Рис. 2.6: GPG ключ

Настройка автоматических подписей коммитов git

```
5mC4iVDvcljRYFWnNk9ca0CzD7iyxcR8lw6fDa74r90X4l4qA0qx08zp7e8ft0w  
x2hCKUisgeFad4xbgNQY8TVdf15306EKajnlLqhkuXNkeY6jIPps0nvtKLf+FA  
VVKWzSx2tRzEqScvRrHrwx26QBH89T37+A7UerSZgCM+kuNFmb2rm52RSZn5mN  
H0s9vgToX1kYzrDdc7aB9pylSBne4K2dbd627TxXlJN0rUwNUU9sMpeQERSZxwIi  
SSFo1S0Jsz0rjdU4ibJJIoKMvQj9P9S1FAovJU8wT+gxKZjQxII3A0DjUlhzt3I+  
B6QzXU1EcswkHJFw1zpJX6X9VkvzY0LA/q/4Z8SFrdXgrtL4yLpekTN4u9TyE  
93o66nyXHQ==
```

```
=xyxX
```

```
-----END PGP PUBLIC KEY BLOCK-----
```

```
pachermashentsev@pachermashentsev:~$
```

```
pachermashentsev@pachermashentsev:~$
```

```
pachermashentsev@pachermashentsev:~$ git config --global user.signingkey 76C040F9E69BC8FD
```

```
pachermashentsev@pachermashentsev:~$ git config --global commit.gpgsign true
```

```
pachermashentsev@pachermashentsev:~$ git config --global gpg.program $(which gpg2)
```

```
pachermashentsev@pachermashentsev:~$
```

Рис. 2.7: Параметры репозитория

Настройка gh

```
pachermashentsev@pachermashentsev:~$ gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? SSH
? Upload your SSH public key to your GitHub account? /home/pachermashentsev/.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: 8BE4-6BED
Press Enter to open https://github.com/login/device 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/pachermashentsev/.ssh/id_rsa.pub
✓ Logged in as palcher1212
pachermashentsev@pachermashentsev:~$
```

Рис. 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация

```
pachermashentsev@pachermashentsev:~$  
pachermashentsev@pachermashentsev:~$ mkdir -p ~/work/study/2024-2025/"Операционные системы"  
pachermashentsev@pachermashentsev:~$ cd ~/work/study/2024-2025/"Операционные системы"  
pachermashentsev@pachermashentsev:~/work/study/2024-2025/Операционные системы$ gh repo create os-intro --template=yamadharma/course-di  
rectory-student-template --public  
✓ Created repository palcher1212/os-intro on GitHub  
https://github.com/palcher1212/os-intro  
pachermashentsev@pachermashentsev:~/work/study/2024-2025/Операционные системы$ git clone --recursive git@github.com:palcher1212/os-int  
ro.git os-intro  
Клонирование в «os-intro»...  
The authenticity of host 'github.com (140.82.121.3)' can't be established.  
ED25519 key fingerprint is SHA256:+DiY3wvV6Tu3JhbpZisF/zLDA0zPMSvHdKx4UvCOqU.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Рис. 2.9: Загрузка шаблона

Подготовка репозитория и коммит изменений

```

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 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/pandocattributes.py
create mode 100644 project-personal/stage6/report/report.md
pachermashentsev@pachermashentsev:~/work/study/2024-2025/Операционные системы/os-intro$ git push
Перечисление объектов: 38, готово.
Подсчет объектов: 100% (38/38), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (38/38), готово.
Запись объектов: 100% (37/37), 342.27 КиБ | 2.28 МБ/с, готово.
Total 37 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:palcher1212/os-intro.git
   0f256fb..b8c1f52  master -> master
pachermashentsev@pachermashentsev:~/work/study/2024-2025/Операционные системы/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: