

Отчёта по лабораторной работе №2

Операционные системы

Федорова Анжелика Игоревна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	13
6	Ответы на контрольные вопросы	14
7	Список литературы	17

Список иллюстраций

Список таблиц

1 Цель работы

Цель данной лабораторной работы – изучение идеологии и применения средств контроля версий, освоение умения по работе с git.

2 Задание

1. Создать базовую конфигурацию для работы с git
2. Создать ключ SSH
3. Создать ключ GPG
4. Настроить подписи Git
5. Зарегистрироваться на GitHub
6. Создать локальный каталог для выполнения заданий по предмету.

3 Теоретическое введение

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.

Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями. Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

4 Выполнение лабораторной работы

Устанавливаю необходимое программное обеспечение git с помощью команд:
dnf install git (рис. fig:001).

```
aifedorova@aifedorova:~$ sudo -i
[sudo] пароль для aifedorova:
root@aifedorova:~# dnf install git
Последняя проверка окончания срока действия метаданных: 2:22:20 назад, Вс 18 фев
2024 19:51:17.
Пакет git-2.43.0-1.fc39.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
```

Также устанавливаю необходимое программное обеспечение gh с помощью команды dnf install gh (рис. fig:002)

```
root@aifedorova:~# dnf install gh
Последняя проверка окончания срока действия метаданных: 2:23:15 назад, Вс 18 фев
2024 19:51:17.
```

Задаю в качестве имени и email владельца репозитория свои имя, фамилию и электронную почту(рис. fig:003)

```
root@aifedorova:~# git config --global user.name "AngelicaFedorova"
root@aifedorova:~# git config --global user.email "iiiggooo666@gmail.com"
root@aifedorova:~#
```

Начальной ветке задаю имя master2, а также задаю параметры autocrlf и safecrlf для корректного отображения конца строки (рис. fig:004)

```
root@aifedorova:~# git config --global init.defaultBranch master2
root@aifedorova:~# git config --global core.autocrlf input
root@aifedorova:~# git config --global core.safecrlf warn
```

Создаю ключ ssh размером 4096 бит по алгоритму rsa (рис. fig:005)


```

root@aifedorova:~# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): thekey
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in thekey
Your public key has been saved in thekey.pub
The key fingerprint is:
SHA256:UlMya8j6MLz7yaUk6tTK7nfA0x8PgYMjiDP6SY7PctY root@aifedorova
The key's randomart image is:
+---[RSA 4096]-----+
|      o .           |
|      . . =         |
|..    .o.=          |
|=   ..o.o0..        |
|.o o=o..S.          |
|. . .+=..o          |
|=.o+o0..+          |
|o+*oE=+. .         |
| OX..o=             |
+---[SHA256]-----+
root@aifedorova:~#

```

Генерирую ключ GPG, затем выбираю тип ключа RSA and RSA, задаю максимальную длину ключа: 4096, оставляю неограниченный срок действия ключа. (рис. fig:006)

```

Вы выбрали следующий идентификатор пользователя:
"Angelica <iiiiggooo666@gmail.com>"

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: /root/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/root/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/root/.gnupg/openpgp-revocs.d/E4AA62F85B0ADABC
9881190D687242ABC0766045.rev'.
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2024-02-18 [SC]
       E4AA62F85B0ADABC9881190D687242ABC0766045
uid           Angelica <iiiiggooo666@gmail.com>
sub    rsa4096 2024-02-18 [E]

root@aifedorova:~#

```

У меня уже был создан аккаунт на Github, поэтому заново мне регистрироваться не нужно.

Вывожу список созданных ключей в терминал, ищу в результате запроса отпечаток ключа (последовательность байтов для идентификации более длинного, по сравнению с самим отпечатком, ключа), он стоит после знака слеша, копирую его в буфер обмена (рис. fig:007)

```

root@aifedorova:~# 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
[keyboxd]
-----
sec   rsa4096/687242ABC0766045 2024-02-18 [SC]
      E4AA62F85B0ADABC9881190D687242ABC0766045
uid           [ абсолютно ] Angelica <iiiggooo666@gmail.com>
ssb   rsa4096/F67990BC54EDAFc4 2024-02-18 [E]

```

Ввожу в терминале команду, с помощью которой копирую сам ключ GPG в буфер обмена (рис. fig:008)

```

root@aifedorova:~# gpg --armor --export 687242ABC0766045
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGXSZDKBEADTF7BlFbAj4Zgcjrp0wN8sHwGy+ICjhj02Qn/3IA29Fkxlg3VT
ZsZPcKUMiwSs589rTZaccs99IyecRwxZ+AbxygipjiidWhXXL+5AUjXx8Q2qiJM
AHxap47IkiJN/zLX7selvMZBTmas0Kn1qYqR49if4HRwPVD002QafYXvdfPBLc0U
Has8yXdw7jDB/iLGVZHTaRCWYmMsDfA7JmOrSjFQshU/xk3q6i/em+15374tMZZf
/C+ZF1khSEkWQnJY8fS05nKGx8zdiYu2CFwkm2X8pofIGedpCvBpyNVwRyZIUdUa
Ms0kof9JL1Lkm+zdB5LekTqf/aQlQtITd9aMoeGZZBIp/RXWWYB/iPZlBo01iuK
lo5wPjrEwnaHc83aYQ/sK/WnsbYyzYCzm7JMi59GpzFaXw8CrxrENhovTolXbHnq
jJbMHZ6p62J25QJpT0pMf+0SON0koC23+76qukY3o7ElTyAmYgBU1HhzersS5ST5
gtdZ0Y6tCLwzMRc2lrUvKw90BU/o2/0yIzVvl3faa5boFH0pkLKzchqmtVDrbHs/
lpm2YI5H8frXzRS7DBonhnAjLVBucm9CARwwRRYEh6sirX2NxBIq81cuqNv2fcsC
D5UCpsgEzkX0eJhVYPwv/i8uKkrWt8L+emYpF5W6D5WBK9iFuFN0Bxk7QARAQAB
tCFBbmdlbGlljYSA8awlpZ2dnb29vNjY2QGdtYWlsLmNvbT6JA1EEewEIADsWIQTk
qmL4WwrvavJiBGQ1ockKrwHZgRQUCzdJk0QIbAwULCQgHAgIiAgYVCgkICwIEFgID
AQIeBwIXgAAKCRBockKrwHZgRcR1EADM70V3A/SHuoIT+WGzXSdtgDj6zPAW0iIr


```

Я добавила ключ GPG на GitHub (рис. fig:009)

GPG keys

New GPG key

This is a list of GPG keys associated with your account. Remove any keys that you do not recognize.



Newkey

Email address: `iiiggooo666@gmail.com`

Key ID: `687242ABC0766045`

Subkeys: `F67990BC54EDAFc4`

Added on Feb 19, 2024

Delete

Настраиваю автоматические подписи коммитов git: используя введенный ранее email, указываю git использовать его при создании подписей коммитов (рис. fig:010)

```

root@aifedorova:~# git config --global user.signingkey 687242ABC0766045
root@aifedorova:~# git config --global commit.gpgsign true
root@aifedorova:~# git config --global gpg.program $(which gpg2)
root@aifedorova:~#

```

Начинаю авторизацию в gh, отвечаю на наводящие вопросы от утилиты, в конце выбираю авторизоваться через браузер. После завершения получаю сообщение о завершении авторизации под своим именем. (рис. fig:011)

```
root@aifedorova:~  
libEGL warning: egl: failed to create dri2 screen  
(epiphany:7350): Gtk-CRITICAL **: 00:15:22.269: Unable to register the application: GDBus.Error:org.freedesktop.DBus.Error.NameHasNoOwner: Could not activate remote peer: unit failed.  
** (process:2): WARNING **: 00:15:22.589: Failed to get atspi registered event listeners: GDBus.Error:org.freedesktop.DBus.Error.NameHasNoOwner: Could not activate remote peer: unit failed.  
(epiphany:7350): Gtk-CRITICAL **: 00:15:35.767: Unable to register the application: GDBus.Error:org.freedesktop.DBus.Error.NameHasNoOwner: Could not activate remote peer: unit failed.  
(epiphany:7350): Gtk-CRITICAL **: 00:15:35.865: Unable to register the application: GDBus.Error:org.freedesktop.DBus.Error.NameHasNoOwner: Could not activate remote peer: unit failed.  
(WebKitWebProcess:2): GStreamer-WARNING **: 00:26:13.663: External plugin loader failed. This most likely means that the plugin loader helper binary was not found or could not be run. You might need to set the GST_PLUGIN_SCANNER environment variable if your setup is unusual. This should normally not be required though.  
(epiphany:7350): Gtk-CRITICAL **: 00:33:24.956: Unable to register the application: GDBus.Error:org.freedesktop.DBus.Error.NameHasNoOwner: Could not activate remote peer: unit failed.  
(epiphany:7350): Gtk-CRITICAL **: 00:33:29.953: Unable to register the application: GDBus.Error:org.freedesktop.DBus.Error.NameHasNoOwner: Could not activate remote peer: unit failed.  
(epiphany:7350): epiphany-WARNING **: 00:33:36.978: Failed to store password record for (https://github.com, https://github.com, iiiiiggoo066@gmail.com, login, password): Объект по пути «/org/freedesktop/secrets/collection/login» не существует  
failed to authenticate via web browser: This 'device_code' has expired. (expired_token)
```

Сначала создаю директорию с помощью утилиты `mkdir` и флага `-p`, который позволяет установить каталоги на всем указанном пути. После этого с помощью утилиты `cd` перехожу в только что созданную директорию “Операционные системы”.(рис. fig:012)

```
root@aifedorova:~# mkdir -p ~/work/study/2022-2023/"Операционные системы"  
root@aifedorova:~# cd ~/work/study/2022-2023/"Операционные системы"
```

Теперь я снова захожу под своим именем через утилиту `gh` (рис. fig:013)

```
root@aifedorova:~/work/study/2022-2023/Операционные системы# gh auth login  
? What account do you want to log into? GitHub.com  
? What is your preferred protocol for Git operations on this host? SSH  
? Generate a new SSH key to add to your GitHub account? No  
? How would you like to authenticate GitHub CLI? Paste an authentication token  
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens  
The minimum required scopes are 'repo', 'read:org'.  
? Paste your authentication token: *****  
- gh config set -h github.com git_protocol ssh  
✓ Configured git protocol  
✓ Authentication credentials saved in plain text  
✓ Logged in as Angelica23i4i4
```

Далее в терминале ввожу команду `gh repo create study_2023-2024_os-intro --template yamadharm/course-directory-student-trmplate --public`, чтобы создать репозиторий на основе шаблона репозитория. (рис. fig:014)

```
root@aifedorova:~/work/study/2022-2023/Операционные системы# gh repo create study_2023-2024_os-intro --template=yamadharm/course-directory-student-template --public  
✓ Created repository Angelica23i4i4/study_2023-2024_os-intro on GitHub  
https://github.com/Angelica23i4i4/study_2023-2024_os-intro
```

После этого клонирую репозиторий к себе в директорию, я указываю ссылку с протоколом `https`, а не `ssh`, потому что при авторизации в `gh` выбрала протокол `https` (рис. fig:015)

```

root@aifedorova: /work/study/2022-2023/Операционные системы# git clone --recursive https://github.com/Angelica231414/study_2023-2024_os-intro.git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 32 (delta 1), reused 18 (delta 0), pack-reused 0
Получение объектов: 100% (32/32), 18.60 КиБ | 334.00 КиБ/с, готово.
Определение изменений: 100% (1/1), 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 «/root/work/study/2022-2023/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 95, done.
remote: Counting objects: 100% (95/95), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 95 (delta 34), reused 87 (delta 26), pack-reused 0
Получение объектов: 100% (95/95), 96.99 КиБ | 834.00 КиБ/с, готово.
Определение изменений: 100% (34/34), готово.
Клонирование в «/root/work/study/2022-2023/Операционные системы/os-intro/template/report»...
remote: Enumerating objects: 126, done.
remote: Counting objects: 100% (126/126), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 126 (delta 52), reused 108 (delta 34), pack-reused 0
Получение объектов: 100% (126/126), 335.80 КиБ | 1.23 МБ/с, готово.
Определение изменений: 100% (52/52), готово.
Submodule path 'template/presentation': checked out '40a1761813e197d00e8443ff1ca72c60a304f24c'
Submodule path 'template/report': checked out '7c31ab8e5dfa8cdb2d67caeb8a19ef8028ced88e'

```

Перехожу в каталог курса с помощью утилиты `cd` и Удаляю лишние файлы с помощью утилиты `rm` (рис. fig:016)

```

root@aifedorova: /work/study/2022-2023/Операционные системы# cd -/work/study/2022-2023/"Операционные системы"/os-intro
root@aifedorova: /work/study/2022-2023/Операционные системы/os-intro# rm package.json
rm: удалить обычный файл 'package.json'? y
root@aifedorova: /work/study/2022-2023/Операционные системы/os-intro#

```

Далее создаю необходимые каталоги используя `makefile` (рис. fig:017)

```

root@aifedorova: /work/study/2022-2023/Операционные системы/os-intro# echo os-intro > COURSE
root@aifedorova: /work/study/2022-2023/Операционные системы/os-intro# make

```

Добавляю все новые файлы для отправки на сервер (сохраняю добавленные изменения) с помощью команды `git add` и комментирую их с помощью `git commit` (рис. fig:018)

```

aifedorova@aifedorova: /work/study/2022-2023/Операционные системы/os-intro$ git commit -am 'feat(main): cleanups'
[master cd75412] feat(main): cleanups
361 files changed, 98412 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/_init_.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 labs/lab01/report/report.md
create mode 100644 labs/lab02/presentation/Makefile
create mode 100644 labs/lab02/presentation/image/kulyabov.jpg
create mode 100644 labs/lab02/presentation/presentation.md
create mode 100644 labs/lab02/report/Makefile
create mode 100644 labs/lab02/report/bib/cite.bib
create mode 100644 labs/lab02/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab02/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_tablenos.py

```

Отправляю файлы на сервер с помощью `git push` (рис. fig:019)

```

aifedorova@aifedorova: /work/study/2022-2023/Операционные системы/os-intro$ git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 341.45 КиБ | 4.32 МБ/с, готово.
Всего 38 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/Angelica231414/study_2023-2024_os-intro.git
   b53c7ce..cd75412  master -> master
aifedorova@aifedorova: /work/study/2022-2023/Операционные системы/os-intro$

```

5 Выводы

Я изучила идеологию и применение средств контроля версий, освоение умения по работе с git.

6 Ответы на контрольные вопросы

1. Системы контроля версий (VCS) - программное обеспечение для облегчения работы с изменяющейся информацией. Они позволяют хранить несколько версий изменяющейся информации, одного и того же документа, может предоставить доступ к более ранним версиям документа. Используется для работы нескольких человек над проектом, позволяет посмотреть, кто и когда внес какое-либо изменение и т. д. VCS применяются для: Хранения полной истории изменений, сохранения причин всех изменений, поиска причин изменений и совершивших изменение, совместной работы над проектами.
2. Хранилище – репозиторий, хранилище версий, в нем хранятся все документы, включая историю их изменения и прочей служебной информацией. commit – отслеживание изменений, сохраняет разницу в изменениях. История – хранит все изменения в проекте и позволяет при необходимости вернуться/обратиться к нужным данным. Рабочая копия – копия проекта, основанная на версии из хранилища, чаще всего последней версии.
3. Централизованные VCS (например: CVS, TFS, AccuRev) – одно основное хранилище всего проекта. Каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет, затем добавляет изменения обратно в хранилище. Децентрализованные VCS (например: Git, Bazaar) – у каждого пользователя свой вариант репозитория (возможно несколько вариантов), есть возможность добавлять и забирать

изменения из любого репозитория. В отличие от классических, в распределенных (децентрализованных) системах контроля версий центральный репозиторий не является обязательным.

4. Сначала создается и подключается удаленный репозиторий, затем по мере изменения проекта эти изменения отправляются на сервер.
5. Участник проекта перед началом работы получает нужную ему версию проекта в хранилище, с помощью определенных команд, после внесения изменений пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются. К ним можно вернуться в любой момент.
6. Хранение информации о всех изменениях в вашем коде, обеспечение удобства командной работы над кодом.
7. Создание основного дерева репозитория: `git init`

Получение обновлений (изменений) текущего дерева из центрального репозитория: `git pull`

Отправка всех произведённых изменений локального дерева в центральный репозиторий: `git push`

Просмотр списка изменённых файлов в текущей директории: `git status`

Просмотр текущих изменений: `git diff`

Сохранение текущих изменений: добавить все изменённые и/или созданные файлы и/или каталоги: `git add .`

добавить конкретные изменённые и/или созданные файлы и/или каталоги: `git add имена_файлов`

удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории): `git rm имена_файлов`

Сохранение добавленных изменений:

сохранить все добавленные изменения и все изменённые файлы: `git commit -am 'Описание коммита'`

сохранить добавленные изменения с внесением комментария через встроенный редактор: `git commit`

создание новой ветки, базирующейся на текущей: `git checkout -b имя_ветки`

переключение на некоторую ветку: `git checkout имя_ветки` (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)

отправка изменений конкретной ветки в центральный репозиторий: `git push origin имя_ветки`

слияние ветки с текущим деревом: `git merge --no-ff имя_ветки`

Удаление ветки:

удаление локальной уже слитой с основным деревом ветки: `git branch -d имя_ветки`

принудительное удаление локальной ветки: `git branch -D имя_ветки`

удаление ветки с центрального репозитория: `git push origin :имя_ветки`

8. `git push -all` отправляем из локального репозитория все сохраненные изменения в центральный репозиторий, предварительно создав локальный репозиторий и сделав предварительную конфигурацию.
9. Ветвление - один из параллельных участков в одном хранилище, исходящих из одной версии, обычно есть главная ветка. Между ветками, т. е. их концами возможно их слияние. Используются для разработки новых функций.
10. Во время работы над проектом могут создаваться файлы, которые не следуют добавлять в репозиторий. Например, временные файлы. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл `.gitignore` с помощью сервисов.

7 Список литературы

Github tutorial