

Отчёта по лабораторной работе № 2

Дисциплина: Операционные системы

Неустроева Ирина Николаевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	15
6	Ответы на контрольные вопросы	16

Список иллюстраций

4.1	Установка гита	8
4.2	Установка gh	8
4.3	Базовая настройка git	8
4.4	Создайте ключи ssh	9
4.5	Создайте ключи ssh	9
4.6	Создайте ключа pgr	10
4.7	Добавление PGP ключ в GitHub	10
4.8	Добавление PGP ключ в GitHub	11
4.9	Настройка автоматических подписей коммитов	11
4.10	Авторизование	11
4.11	Авторизование	12
4.12	Создание рабочего пространства	12

Список таблиц

1 Цель работы

Изучить идеологию и применение средств контроля версий и освоить умения по работе с git.

2 Задание

1. Установка программного обеспечения.
2. Сделать базовую настройку git.
3. Создать ключи ssh и pgr.
4. Авторизоваться в GitHub.
5. Настроить каталог курса.
6. Удалить лишние файлы и отправить файлы на сервер.

3 Теоретическое введение

Системы контроля версий (Version Control System, VCS) применяются при работе несколь

В классических системах контроля версий используется централизованная модель, предпо
компрессию – сохранять только изменения между последовательными версиями, что позвол

Системы контроля версий поддерживают возможность отслеживания и разрешения конфликто

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функци

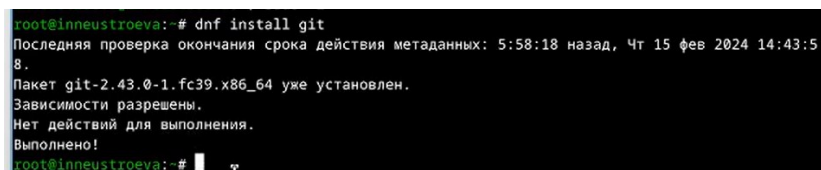
В отличие от классических, в распределённых системах контроля версий цен-
тральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди распре-
делённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они
в основном синтаксисом используемых в работе команд.

4 Выполнение лабораторной работы

- Установка программного обеспечения

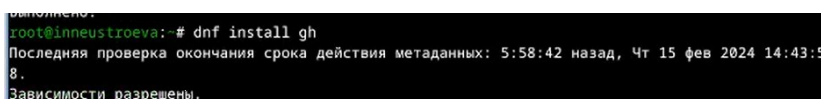
При помощи команды: `dnf install git`, устанавливаем git (рис. 4.1).



```
root@inneustroeva:~# dnf install git
Последняя проверка окончания срока действия метаданных: 5:58:18 назад, Чт 15 фев 2024 14:43:58.
Пакет git-2.43.0-1.fc39.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
root@inneustroeva:~#
```

Рис. 4.1: Установка гита

При помощи команды: `dnf install gh`, устанавливаем gh (рис. 4.2).

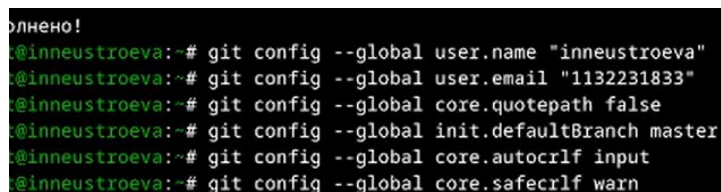


```
Выполнено!
root@inneustroeva:~# dnf install gh
Последняя проверка окончания срока действия метаданных: 5:58:42 назад, Чт 15 фев 2024 14:43:58.
Зависимости разрешены.
```

Рис. 4.2: Установка gh

- Базовая настройка git

Задаем имя и email владельца репозитория, настраиваем utf-8 в выводе сообщений git, задаем имя начальной ветки и параметры (рис. 4.3).



```
Выполнено!
root@inneustroeva:~# git config --global user.name "inneustroeva"
root@inneustroeva:~# git config --global user.email "1132231833"
root@inneustroeva:~# git config --global core.quotepath false
root@inneustroeva:~# git config --global init.defaultBranch master
root@inneustroeva:~# git config --global core.autocrlf input
root@inneustroeva:~# git config --global core.safecrlf warn
```

Рис. 4.3: Базовая настройка git

- Создать ключи ssh и pgp

Создали по алгоритму rsa с ключём размером 4096 бит командой: `ssh-keygen -t rsa -b 4096` (рис. 4.4).

```
root@inneustroeveva:~# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:8b1kH80CCBziRqIqtHxyec8zwmNjGqAF6f0IqxFFpME root@inneustroeveva
The key's randomart image is:
+--[ED25519 256]--+
|o.+ o .|
| E + o .|
|=. o o .|
|=o... . + o .|
|+B.+ . S o = +|
|+oO o o o + o |
|+. o B = . . |
| o = + o |
|+ . |
+----[SHA256]-----+
root@inneustroeveva:~#
```

Рис. 4.4: Создайте ключи ssh

Создали по алгоритму ed25519 командой: `ssh-keygen -t ed25519` (рис. 4.5).

```
ot@inneustroeveva:~# xclip -i < ~/.ssh/id_ed25519.pub
```

Рис. 4.5: Создайте ключи ssh

Сгенерировали ключ командой: `gpg --full-generate-key` и из предложенных опций выбирали: тип RSA and RSA; размер 4096; срок действия 0(срок действия не истекает никогда). Так же GPG запросил Имя и Адрес электронной почты. (рис. 4.6)

```

root@inneustroeva:~# gpg --full-generate-key
gpg (GnuPG) 2.4.3; Copyright (C) 2023 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 '/root/.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

```

Рис. 4.6: Создайте ключа gpg

Далее необходимо настроить учетную запись на <https://github.com> и заполнить основные данные. Я это уже сделала в прошлом семестре.

- **Добавление PGP ключа в GitHub**

Выводим список ключей и копируем отпечаток приватного ключа командой: `gpg --list-secret-keys --keyid-format LONG`. Далее скопировали наш сгенерированный PGP ключ в буфер обмена командой: `gpg --armor --export 3C6993D86C710331 | xclip -sel clip` (рис. 4.7).

```

root@inneustroeva:~# 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  rsa4096/3C6993D86C710331 2024-02-15 [SC]
      2B2EA04D9B05AFF4DE4D4F213C6993D86C710331
uid          [ абсолютно ] inneustroeva <1132231833@pfur.ru>
ssb  rsa4096/7058C3CB1AAAF07A 2024-02-15 [E]

root@inneustroeva:~# gpg --armor --export 3C6993D86C710331

```

Рис. 4.7: Добавление PGP ключ в GitHub

Переходим в настройки GitHub и вставляем полученный ключ в поле ввода (рис. 4.8).

PGP keys

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

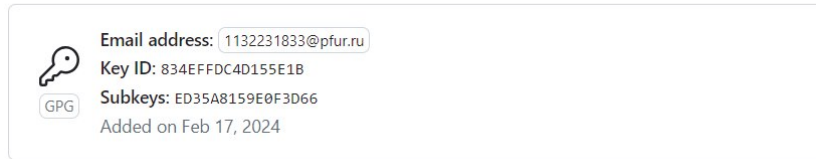


Рис. 4.8: Добавление PGP ключ в GitHub

- **Настройка автоматических подписей коммитов git**

Далее используя введенный email, указываю Git и применяю его при подписи коммитов (рис. 4.9).

```
root@inneustroeva:~# git config --global user.signingkey 3C6993D86C710331
root@inneustroeva:~# git config --global commit.gpgsign true
root@inneustroeva:~# git config --global gpg.program $(which gpg2)
root@inneustroeva:~# gh auth login
```

Рис. 4.9: Настройка автоматических подписей коммитов

Потом нам необходимо авторизоваться командой: `gh auth login`, отвечаем на вопросы Утилиты (рис. 4.10).

```
root@inneustroeva:~# gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations on this host? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 92AA-A274
Press Enter to open github.com in your browser...
```

Рис. 4.10: Авторизование

Авторизация прошла успешно (рис. 4.11).

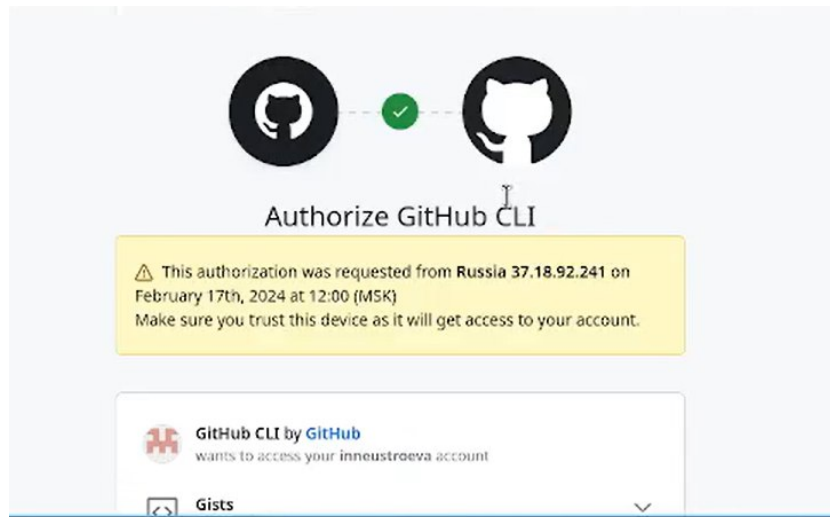


Рис. 4.11: Авторизование

- **Создание Шаблон для рабочего пространства**

Далее нам необходимо создать репозиторий курса на основе шаблона (рис. 4.12).

```
inneustroeva@inneustroeva:~$ mkdir -p ~/work/study/2023-2024/"Operating systems"
inneustroeva@inneustroeva:~$ cd ~/work/study/2023-2024/"Operating systems"
inneustroeva@inneustroeva:~/work/study/2023-2024/Operating systems$ gh repo create studo_2023-2024_os-intro
--template=yamadharma/course-directory-student-template --public
```

Рис. 4.12: Создание рабочего пространства

С помощью команды: `git clone --recursive git@github.com:inneustroeva/study_2022-2023_os-intro.git os-intro`, клонируем нужный нам шаблон из гитхаба (рис. ??).

```
inneustroeva:~/work/study/2023-2024/Operating systems$ git clone --recursive git@github.com:inneustroeva/study_2023-2024_os-intro.git os-
int
Cloning into 'os-intro'...
remote: Enumerating objects: 1, done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 1
Unpacking objects: 100% (1/1)
done.
 authenticity of host 'github.com (140.82.121.4)' can't be established.
 RSA key fingerprint is SHA256:DiY3wvV6TujhbpZisF/zLDABzPKSVHdkz4UvCOqU.
 key is not known by any other names.
 you sure you want to continue connecting (yes/no/[fingerprint])? y
```

Переходим в каталог курса, удаляем лишние файлы, создаем необходимые каталоги (рис. ??).

```
root@inneustroeva:~/work/study/2023-2024/Operating systems$ cd ~/work/study/2023-2024/"Operating systems"/os-intro
root@inneustroeva:~/work/study/2023-2024/Operating systems/os-intro$ rm package.json
rm: удалить обычный файл 'package.json'? y
root@inneustroeva:~/work/study/2023-2024/Operating systems/os-intro$ ls
CHANGELOG.md  config  COURSE  LICENSE  Makefile  README.en.md  README.git-flow.md  README.md  template
root@inneustroeva:~/work/study/2023-2024/Operating systems/os-intro$ echo os-intro COURSE
os-intro COURSE
root@inneustroeva:~/work/study/2023-2024/Operating systems/os-intro$ make
Usage:
  make <target>

Targets:
  list          List of courses
  prepare       Generate directories structure
  submodule     Update submodules
root@inneustroeva:~/work/study/2023-2024/Operating systems/os-intro$
```

Проверяем создание необходимых каталогов командой: `ls`. И начинаю отправ-
лять файлы на гитхаб командой: `git add .` (рис. ??).

```
root@inneustroeva:~/work/study/2023-2024/Operating systems/os-intro# make prepare
root@inneustroeva:~/work/study/2023-2024/Operating systems/os-intro# ls
CHANGELOG.md  COURSE  LICENSE  prepare  project-personal  README.git-flow.md  template
config        labs   Makefile  presentation  README.en.md      README.md
root@inneustroeva:~/work/study/2023-2024/Operating systems/os-intro# git status
Текущая ветка: master
Эта ветка соответствует «origin/master».

Изменения, которые не в индексе для коммита:
(используйте «git add/xm <файл>...», чтобы добавить или удалить файл из индекса)
(используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
изменено:    COURSE
удалено:     package.json

Неотслеживаемые файлы:
(используйте «git add <файл>...», чтобы добавить в то, что будет включено в коммит)
labs/
prepare
presentation/
project-personal/

индекс пуст (используйте «git add» и/или «git commit -a»)
root@inneustroeva:~/work/study/2023-2024/Operating systems/os-intro# git add .
root@inneustroeva:~/work/study/2023-2024/Operating systems/os-intro#
```

Пишем команду: `git commit -am 'feat(main): make course structure'` (рис. ??).

```
новый файл:   project-personal/stage5/report/pandoc/filters/pandocxnos/__init__.py
новый файл:   project-personal/stage5/report/pandoc/filters/pandocxnos/core.py
новый файл:   project-personal/stage5/report/pandoc/filters/pandocxnos/main.py
новый файл:   project-personal/stage5/report/pandoc/filters/pandocxnos/pandocattributes.py
новый файл:   project-personal/stage5/report/report.md
новый файл:   project-personal/stage6/presentation/Makefile
новый файл:   project-personal/stage6/presentation/image/kulyabov.jpg
новый файл:   project-personal/stage6/presentation/presentation.md
новый файл:   project-personal/stage6/report/Makefile
новый файл:   project-personal/stage6/report/bib/cite.bib
новый файл:   project-personal/stage6/report/image/placeimg_800_600_tech.jpg
новый файл:   project-personal/stage6/report/pandoc/csl/gost-x-7-0-5-2008-numeric.csl
новый файл:   project-personal/stage6/report/pandoc/filters/pandoc_eqnos.py
новый файл:   project-personal/stage6/report/pandoc/filters/pandoc_fignos.py
новый файл:   project-personal/stage6/report/pandoc/filters/pandoc_secnos.py
новый файл:   project-personal/stage6/report/pandoc/filters/pandoc_tablenos.py
новый файл:   project-personal/stage6/report/pandoc/filters/pandocxnos/__init__.py
новый файл:   project-personal/stage6/report/pandoc/filters/pandocxnos/core.py
новый файл:   project-personal/stage6/report/pandoc/filters/pandocxnos/main.py
новый файл:   project-personal/stage6/report/pandoc/filters/pandocxnos/pandocattributes.py
новый файл:   project-personal/stage6/report/report.md
t@inneustroeva:~/work/study/2023-2024/Operating systems/os-intro# git commit -am 'feat(main): make course structure'
```

С помощью команды: `git push` отправляем файлы на гитхаб (рис. ??).

```
Файл  Машина  Вид  Ввод  Устройства  Справка
Терминал - inneustroeva@inneustroeva:~$
Файл  Правка  Вид  Терминал  Вкладки  Справка
create mode 100644 project-personal/stage5/presentation/presentation.md
create mode 100644 project-personal/stage5/report/Makefile
create mode 100644 project-personal/stage5/report/bib/cite.bib
create mode 100644 project-personal/stage5/report/image/placeimg_800_600_tech.jpg
create mode 100644 project-personal/stage5/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 project-personal/stage5/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 project-personal/stage5/report/pandoc/filters/pandoc_fignos.py
create mode 100755 project-personal/stage5/report/pandoc/filters/pandoc_secnos.py
create mode 100755 project-personal/stage5/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 project-personal/stage5/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 project-personal/stage5/report/pandoc/filters/pandocxnos/core.py
create mode 100644 project-personal/stage5/report/pandoc/filters/pandocxnos/main.py
create mode 100644 project-personal/stage5/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 project-personal/stage5/report/report.md
create mode 100644 project-personal/stage6/presentation/Makefile
create mode 100644 project-personal/stage6/presentation/image/kulyabov.jpg
create mode 100644 project-personal/stage6/presentation/presentation.md
create mode 100644 project-personal/stage6/report/Makefile
create mode 100644 project-personal/stage6/report/bib/cite.bib
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
root@inneustroeva:~/work/study/2023-2024/Operating systems/os-intro# git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 341.45 Киб | 1.89 Миб/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 github.com:inneustroeva/study_2023-2024_os-intro.git
 9d797f6..e607a76  master -> master
root@inneustroeva:~/work/study/2023-2024/Operating systems/os-intro#
```

Заходим на гитхаб и проверяем наши файлы (рис. ??).

Most Visited Fedora Docs Fedora Magazine Fedora Project User Communities Red Hat Free Content

inneustroeva / study_2023-2024_os-intro

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

study_2023-2024_os-intro Public

generated from [yamadharma/course-directory-student-template](#)

master 1 Branch 0 Tags

Go to file

code

inneustroeva feat(main): make course structure	e607a76 · 11 minutes ago	2 Commits
config	Initial commit	5 days ago
labs	feat(main): make course structure	11 minutes ago
presentation	feat(main): make course structure	11 minutes ago
project-personal	feat(main): make course structure	11 minutes ago
template	Initial commit	5 days ago
__init__.py	Initial commit	5 days ago

About

No description, website, or topics provided.

Readme

CC-BY 4.0 license

Activity

0 stars

1 watching

0 forks

Releases

5 Выводы

Я научилась работать с системой контроля версий гит и освоить умения по работе с git, сделала базовую настройку.

6 Ответы на контрольные вопросы

1 Что такое системы контроля версий (VCS) и для решения каких задач они предназначены? Это программное обеспечение для облегчения работы с изменяющейся информацией. VCS позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

2 Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. Хранилище (repository), или репозиторий, — место хранения всех версий и служебной информации. Commit («[трудовой] вклад», не переводится) — синоним версии; процесс создания новой версии. История — место, где сохраняются все коммиты, по которым можно посмотреть данные о коммитах. Рабочая копия — текущее состояние файлов проекта, основанное на версии, загруженной из хранилища.

3 Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида. Централизованные VCS: одно основное хранилище всего проекта и каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет и, затем, добавляет свои изменения обратно. Децентрализованные VCS: у каждого пользователя свой вариант (возможно не один) репозитория.

4 Опишите действия с VCS при единоличной работе с хранилищем. Для начала создаём удаленный репозиторий и подключаем его к основному. Затем внесим изменения в проект посредством локального репозитория и отправляем изменения на сервер.

5 Опишите порядок работы с общим хранилищем VCS.

6 Каковы основные задачи, решаемые инструментальным средством git? Git — это система управления версиями. У Git две основных задачи: первая — хранить информацию о всех изменениях в вашем коде, начиная с самой первой строчки, а вторая — обеспечение удобства командной работы над кодом.

7 Назовите и дайте краткую характеристику командам git. git -version (Проверка версии Git) git init (Инициализировать ваш текущий рабочий каталог как Git-репозиторий) git clone <https://www.github.com/username/repo-name> (Склонировать существующий удаленный Git-репозиторий) git remote (Просмотреть список текущих удаленных репозиториях Git) git remote -v (Для более подробного вывода) git add my_script.py (Можете указать в команде конкретный файл). git add . (Позволяет охватить все файлы в текущем каталоге, включая файлы, чье имя начинается с точки) git commit -am "Commit message" (Вы можете сжать все индексированные файлы и отправить коммит). git branch (Просмотреть список текущих веток можно с помощью команды branch) git -help (Чтобы узнать больше обо всех доступных параметрах и командах) git push origin master (Передать локальные коммиты в ветку удаленного репозитория).

8 Приведите примеры использования при работе с локальным и удаленным репозиториями.

9 Что такое и зачем могут быть нужны ветви (branches)? Ветки нужны, чтобы несколько программистов могли вести работу над одним и тем же проектом или даже файлом одновременно, при этом не мешая друг другу. Кроме того, ветки используются для тестирования экспериментальных функций: чтобы не повредить основному проекту, создается новая ветка специально для экспериментов. 10 Как и зачем можно игнорировать некоторые файлы при commit? Игнорируемые файлы — это, как правило, артефакты сборки и файлы, генерируемые машиной из исходных файлов в вашем репозитории, либо файлы, которые по какой-либо иной причине не должны попадать в коммиты.