

Цель работы

Изучить идеологию и применение средств контроля версий

Объект исследования

системы контроля версий

Предмет исследования

настройка системы контроля версий

Ход работы

1. Создала учётную запись на [github](#)(рис.1)



ibgoloshchapowa

Edit profile

🕒 Joined 7 days ago

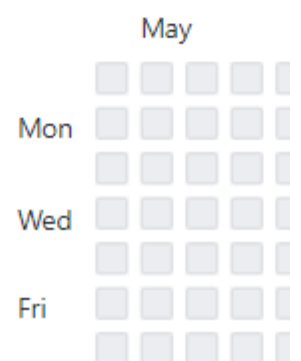
📖 Overview

ProTip! Updating your profile will let others know you.

Popular repositories

[os-intro](#)

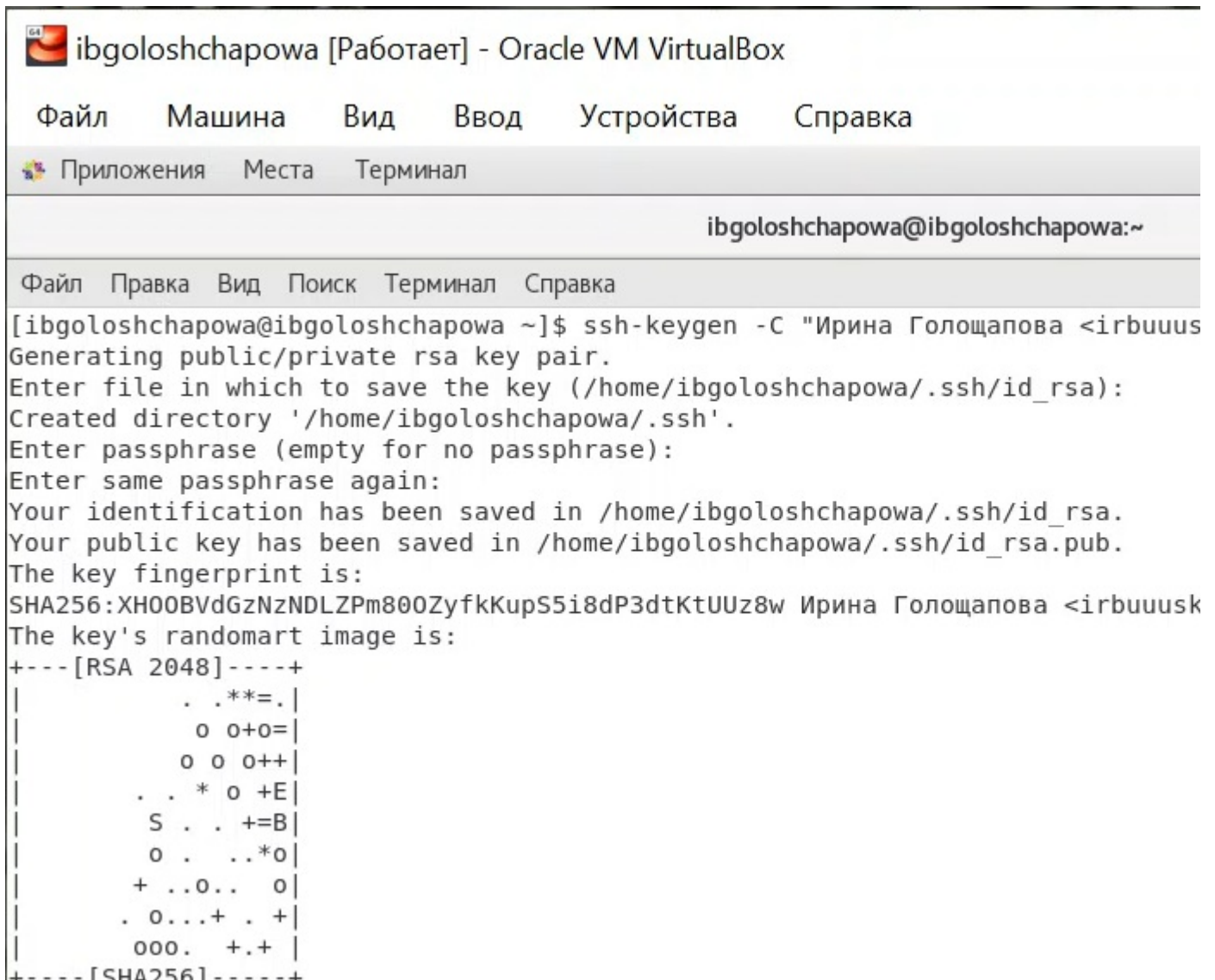
3 contributions in



[Learn how we work](#)

Приступила к настройке системы контроля версий git, как это было описано в указаниях к лабораторной работе с использованием сервера репозитория [github](https://github.com).

2. «Работа с сервером репозитория». Для последующей идентификации пользователя на сервере репозитория я сгенерировала пару ключей (приватный и открытый). Для этого я ввела команду (рис.2), вследствие которой ключи сохранились в каталоге ~/.ssh/.



```
ibgoloshchapowa [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Приложения  Места  Терминал
ibgoloshchapowa@ibgoloshchapowa:~
Файл  Правка  Вид  Поиск  Терминал  Справка
[ibgoloshchapowa@ibgoloshchapowa ~]$ ssh-keygen -C "Ирина Голощанова <irbuuus
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ibgoloshchapowa/.ssh/id_rsa):
Created directory '/home/ibgoloshchapowa/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ibgoloshchapowa/.ssh/id_rsa.
Your public key has been saved in /home/ibgoloshchapowa/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:ХН00BVdGzNzNDLZPm800ZyfkKupS5i8dP3dtKtUUz8w Ирина Голощанова <irbuusk
The key's randomart image is:
+---[RSA 2048]---+
|      . .**=. |
|      o o+o= |
|      o o o++ |
|      . . * o +E |
|      S . . +=B |
|      o . . .*o |
|      + ..0.. o |
|      . o...+ . + |
|      ooo. .+. |
+-----[SHA256]-----+
```

3. Затем зашла на сайт [github](https://github.com) под своей учетной записью и перешла в меню «GitHub settings» (рис.3). После этого выбрала в боковом меню «SSH-ключи» (рис.4) и нажала кнопку «Добавить ключ» (рис.5):



ibgoloshchapowa

[Edit profile](#)

🕒 Joined 6 days ago

Overview

Repositories

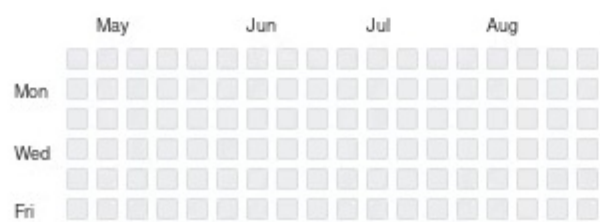
Project

ProTip! Updating your profile with your name, location, and interests helps others find and know you.

Popular repositories

You don't

1 contribution in the last year



Account

Appearance

New

Account security

Billing & plans

Security log

Security & analysis

Emails

Notifications

SSH and GPG keys

Repositories

Packages



ibgoloshchapowa

Your personal account

Account settings

Profile

Account

Appearance

New

Account security

Billing & plans

Security log

SSH keys

There are no SSH keys associated with your account.

Check out our guide to [generating SSH keys](#) or troubleshoot [common S](#)

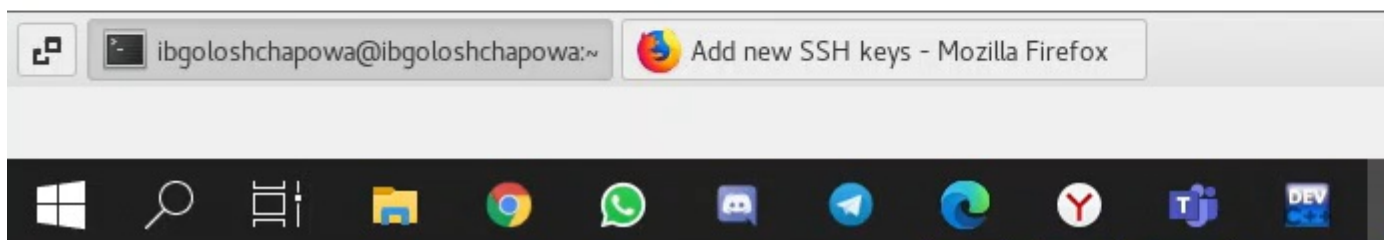
GPG keys

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

4. Перешла в терминал и набрала следующую команду (рис.6), после чего скопировала из локальной консоли ключ в буфер обмена (рис.7):

```
[ibgoloshchapowa@ibgoloshchapowa ~]$ cat ~/.ssh/id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDSVzdGhnwHyjSvnDBmENcm0G6gBTZNZ06MoVeh9  
9ZIC0oaylU660rdUNF5AXs4rX3hWEbBAsdPpULYJMw9w6VSDbcpNBRKBx1ednBPYqgb8YsXLqb8UT  
[ibgoloshchapowa@ibgoloshchapowa ~]$
```





ibgoloshchapowa

Your personal account

Account settings

Profile

Account

Appearance

New

Account security

Billing & plans

Security log

Security & analysis

Emails

Notifications

SSH and GPG keys

SSH keys / Add new

Title

Key

ssh-rsa

```
AAAAB3NzaC1yc2EAAAADAQABAAQDSVzdGhn  
qzhLyMPXBRRDGer+71B9njKZM1yEPozLqIhF1ki+THy  
mpEtujuToNu2hL0K2a103E2P31d6Vs6THsOUCv9ZIC  
bcpNBRKBx1ednBPYqgb8YsXLqb8UTD7V2Qci7AXqKz  
/n+RqPnSygx3YO+d9GUghxlymfQ3RavsqHRZH6pAz  
<irbuuuska@yandex>
```



Add SSH key

В итоге мы добавили новый ключ – «key1» (рис.8):




ibgoloshchapowa

Your personal account

Account settings

Profile

Account

Appearance 

New

Account security

Billing & plans

Security log

Security & analysis

Emails

Notifications

SSH and GPG keys

SSH keys

This is a list of SSH keys associated with your account. Remove



SSH

key1

SHA256: XH00BVdGzNzNDLZPm800ZyfkKupS5i8dP3dtKtU

Added on 28 Apr 2021

Never used — Read/write

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH](#)

GPG keys

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

5. Создала структуру каталога лабораторных работ согласно пункту М.2 (рис.9)

```
[ibgoloshchapowa@ibgoloshchapowa ~]$ cd
[ibgoloshchapowa@ibgoloshchapowa ~]$ mkdir work
[ibgoloshchapowa@ibgoloshchapowa ~]$ cd work
[ibgoloshchapowa@ibgoloshchapowa work]$ mkdir 2020-2021
[ibgoloshchapowa@ibgoloshchapowa work]$ cd 2020-2021
[ibgoloshchapowa@ibgoloshchapowa 2020-2021]$ mkdir Операционные_системы
[ibgoloshchapowa@ibgoloshchapowa 2020-2021]$ cd Операционные_системы
[ibgoloshchapowa@ibgoloshchapowa Операционные_системы]$ mkdir laboratory
[ibgoloshchapowa@ibgoloshchapowa Операционные_системы]$
```

6. Создала на github репозиторий os-intro (рис.10)

Create a new repository

A repository contains all project files, including the revision history. [Import a repository.](#)

Owner *



ibgoloshchapowa ▾

Repository name *

os-intro

Great repository names are short and easy to remember. os-intro is available. Need inspiration?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

7. Инициализировала системы git (рис.11), создала заготовку для файла README.md, сделала первый коммит и выложила на github (рис.12-14)

```
[ibgoloshchapowa@ibgoloshchapowa 2020-2021]$ cd операционные_системы
[ibgoloshchapowa@ibgoloshchapowa Операционные_системы]$ mkdir laboratory
[ibgoloshchapowa@ibgoloshchapowa Операционные_системы]$ cd laboratory
[ibgoloshchapowa@ibgoloshchapowa laboratory]$ git init
Initialized empty Git repository in /home/ibgoloshchapowa/work/2020-2021/Опер
ory/.git/
```

```
[ibgoloshchapowa@ibgoloshchapowa .git]$ cd
[ibgoloshchapowa@ibgoloshchapowa ~]$ mc
```

```
[ibgoloshchapowa@ibgoloshchapowa .git]$ echo "# Лабораторные работы" >> README
```

```

[ibgoloshchapowa@ibgoloshchapowa .git]$ git add README.md
[ibgoloshchapowa@ibgoloshchapowa .git]$ git commit -am "first commit"
[master (root-commit) 32e625b] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
[ibgoloshchapowa@ibgoloshchapowa .git]$ git branch -M main
[ibgoloshchapowa@ibgoloshchapowa .git]$ git remote add origin git@github.com:
.git
[ibgoloshchapowa@ibgoloshchapowa .git]$ git push -u origin main

Counting objects: 3, done.
Writing objects: 100% (3/3), 275 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:ibgoloshchapowa/os-intro.git
* [new branch]      main -> main
Branch main set up to track remote branch main from origin.
[ibgoloshchapowa@ibgoloshchapowa .git]$ █

```

8. Добавила файл лицензии (рис.15):

```

[ibgoloshchapowa@ibgoloshchapowa laboratory]$ wget https://creativecommons.org/licenses/by/4.0/legalcode
--2021-04-29 22:25:27-- https://creativecommons.org/licenses/by/4.0/legalcode
Распознаётся creativecommons.org (creativecommons.org)... 104.20.151.16, 172.
Подключение к creativecommons.org (creativecommons.org)|104.20.151.16|:443...
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: нет данных [text/plain]
Сохранение в: «LICENSE»

[  <=>
] 18 657      52,7KB/s

2021-04-29 22:25:28 (52,7 KB/s) - «LICENSE» сохранён [18657]

[ibgoloshchapowa@ibgoloshchapowa laboratory]$ █

```

9. Добавила шаблон игнорируемых файлов. Просмотрела список имеющихся шаблонов (рис.16)

```
[ibgoloshchapowa@ibgoloshchapowa laboratory]$ curl -L -s https://www.gitignor
lc,lc-bitrix,a-frame,actionscript,ada
adobe,advancedinstaller,adventuregamestudio,agda,al
alteraquartusii,altium,amplify,android,androidstudio
angular,anjuta,ansible,apachecordova,apachehadoop
appbuilder,appcelerator titanium,appcode,appcode+all,appcode+iml
appengine,aptanastudio,arcanist,archive,archives
archlinuxpackages,aspnetcore,assembler,ate,atmelstudio
ats,audio,automationstudio,autotools,autotools+strict
awr,azurefunctions,backup,ballerina,basercms
basic,batch,bazaar,bazel,bitrise
bitrix,bittorrent,blackbox,bloop,bluej
bookdown,bower,bricxcc,buck,c
c++,cake,cakephp,cakephp2,cakephp3
calabash,carthage,certificates,ceylon,cfwheels
chefcookbook,chocolatey,clean,clion,clion+all
clion+iml,clojure,cloud9,cmake,cocoapods
cocos2dx,cocoscreator,code,code-java,codeblocks
codecomposerstudio,codeigniter,codeio,codekit,codesniffer
coffeescript,commonlisp,compodoc,composer,compressed
compressedarchive,compression,coq,concrete5,coq
cordova,craftcms,crashlytics,crbasic,crossbar
crystal,cs-cart,csharp,cuda,cvs
cypressio,d,dart,darteditor,data
database,datarecovery,dbeaver,defold,delphi
dframe,diff,direnv,diskimage,django
dm,docfx,docpress,docz,dotenv
dotfilessh,dotnetcore,dotsettings,dreamweaver,dropbox
drupal,drupal7,drupal8,e2studio,eagle
easybook,eclipse,eiffelstudio,elasticbeanstalk,elisp
elixir,elm,emacs,ember,ensime
episerver,erlang,espresso,executable,exercism
expressionengine,extjs,fancy,fastlane,finale
firebase,flashbuilder,flask,flatpak,flex
flexbuilder,floobits,flutter,font,fontforge
forcedotcom,forgegradle,fortran,freepascal,fsharp
```

10. Скачала шаблон для C (рис.17)

```
cyc,cyrings,cyres,cyres-composer,amberace
unity,unrealengine,vaadin,vagrant,valgind
vapor,venv,vertx,video,vim
virtualenv,virtuoso,visualstudio,visualstudiocode,vivado
vlab,vs,vscodex,vue,vuejs
vvvv,waf,wakanda,web,webmethods
webstorm,webstorm+all,webstorm+iml,werckercli,windows
wintersmith,wordpress,wyam,xamarinstudio,xcode
xcodeinjection,xilinx,xilinxise,xilinxvivado,xill
xojo,xtext,y86,yarn,yeoman
yii,yii2,zendframework,zephir,zig
zsh,zukencr8000[ibgoloshchapowa@ibgoloshchapowa laboratory]$ curl -L -s http
```

11. Добавила новые файлы, выполнила коммит и отправила на github (рис.18)

```
[ibgoloshchapowa@ibgoloshchapowa laboratory]$ git add .
[ibgoloshchapowa@ibgoloshchapowa laboratory]$ git commit -am "license"
[master (root-commit) 906cf01] license
 2 files changed, 455 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 LICENSE
[ibgoloshchapowa@ibgoloshchapowa laboratory]$
```

```
[ibgoloshchapowa@ibgoloshchapowa laboratory]$ git push
fatal: No configured push destination.
Either specify the URL from the command-line or configure a remote repository

    git remote add <name> <url>

and then push using the remote name

    git push <name>

[ibgoloshchapowa@ibgoloshchapowa laboratory]$
```

12. Инициализировала git-flow (префикс для ярлыков установила в v) (рис.19). Также проверила, что нахожусь на ветке develop.

```
[ibgoloshchapowa@ibgoloshchapowa laboratory]$ git flow init
Which branch should be used for bringing forth production releases?
- master
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [] v
[ibgoloshchapowa@ibgoloshchapowa laboratory]$ git branch
* develop
  master
[ibgoloshchapowa@ibgoloshchapowa laboratory]$
```

13. Создала релиз с версией 1.0.0 (рис.20)


```
[ibgoloshchapowa@ibgoloshchapowa laboratory]$ git flow release start 1.0.0
Switched to a new branch 'release/1.0.0'

Summary of actions:
- A new branch 'release/1.0.0' was created, based on 'develop'
- You are now on branch 'release/1.0.0'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish '1.0.0'
```

14. Записала версию, добавила в индекс, залила релизную ветку в основную ветку и отправила данные на github (рис.21).

```
[ibgoloshchapowa@ibgoloshchapowa laboratory]$ echo "1.0.0" >> VERSION
[ibgoloshchapowa@ibgoloshchapowa laboratory]$ git add .
[ibgoloshchapowa@ibgoloshchapowa laboratory]$ git commit -am "chare(main): ad
[release/1.0.0 562f4fb] chare(main): add version
1 file changed, 1 insertion(+)
create mode 100644 VERSION
[ibgoloshchapowa@ibgoloshchapowa laboratory]$ git flow release finish 1.0.0
Switched to branch 'main'
Merge made by the 'recursive' strategy.
VERSION | 1 +
1 file changed, 1 insertion(+)
create mode 100644 VERSION
Switched to branch 'develop'
Merge made by the 'recursive' strategy.
VERSION | 1 +
1 file changed, 1 insertion(+)
create mode 100644 VERSION
Deleted branch release/1.0.0 (was 562f4fb).

Summary of actions:
- Latest objects have been fetched from 'origin'
- Release branch has been merged into 'main'
- The release was tagged 'v1.0.0'
- Release branch has been back-merged into 'develop'
- Release branch 'release/1.0.0' has been deleted
```

15. Обновила github (рис.22-23)

<> Code

! Issues

🔗 Pull requests

▶ Actions

🔗 develop had recent pushes 7 minutes ago

🔗 main

🔗 2 branches

🏷️ 1 tag

Switch branches/tags

Find or create a branch...

Branches

Tags

✓ main

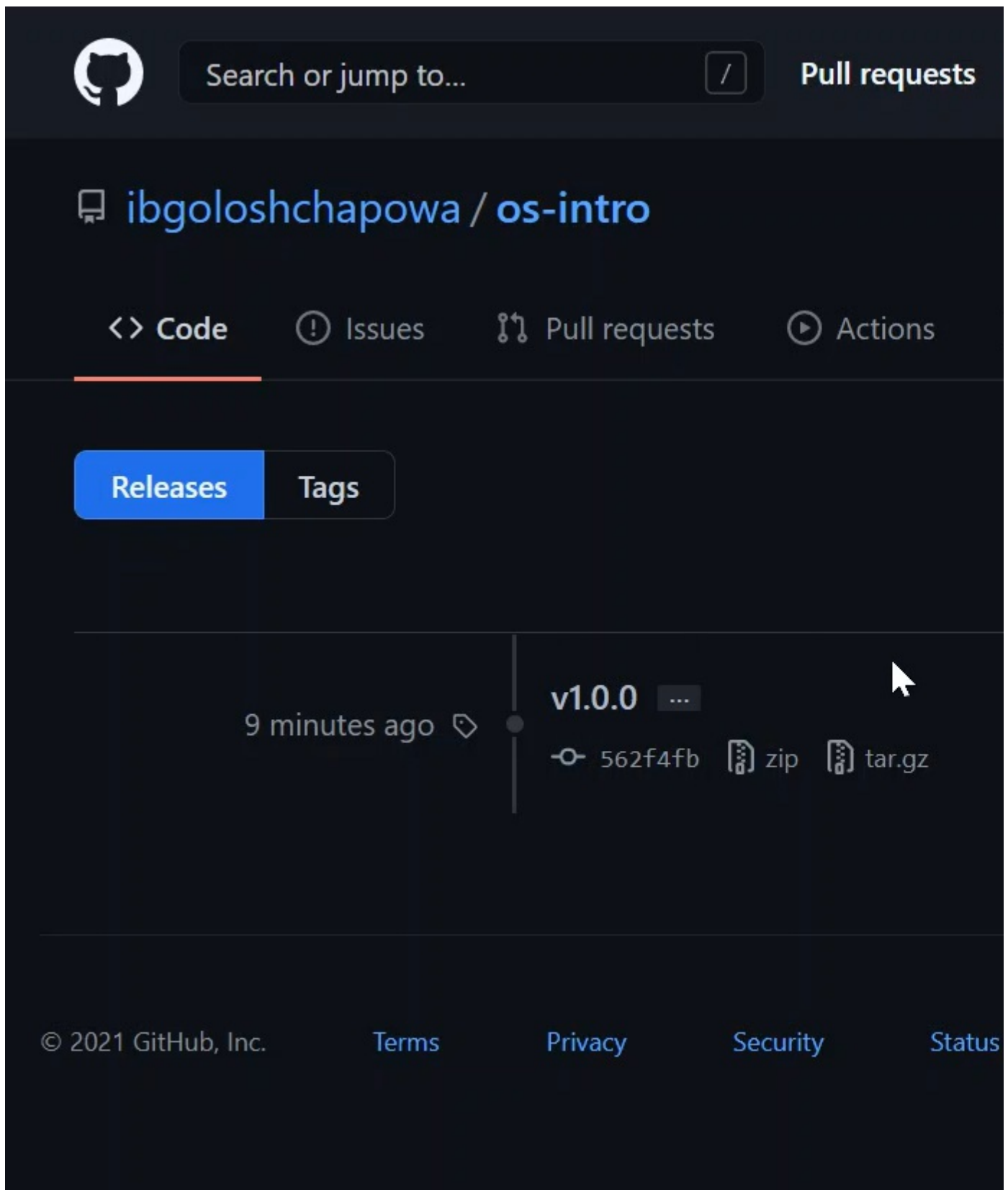
default

develop

[View all branches](#)

README.md

Лабораторные работы



Вывод

В ходе лабораторной работы я изучила идеологию и применение средств контроля версий (VCS). Научилась настраивать систему контроля версий с использованием сервера репозитория на ПК.

Ответы на контрольные вопросы

1. Что такое системы контроля версий и для решения каких задач они предназначены?

Ответ: Система контроля версий (Version Control System, VCS) представляет собой программное обеспечение, которое позволяет отслеживать изменения в документах, при необходимости производить их откат, определять, кто и когда внес исправления и т.п. Также важно отметить, что системы контроля версий применяются и для хранения исходного кода, для хранения файлов совершенно любого типа.

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

Ответ: Хранилище (repository) – место хранения файлов и их версий, служебной информации. Commit – процесс создания новой версии; иногда синоним версии. История – одна из самых важных частей git, где сохраняются все коммиты, по которым можно посмотреть автора коммита, commite message, дату коммита и его хэш. Рабочая копия – текущее состояние файлов проекта (любой версии), полученных из хранилища и, возможно, измененных.

3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

Ответ: В централизованной системе управления версиями код хранится на сервере. Все разработчики в команде имеют доступ к централизованному хранилищу. Однако у разработчика нет своего локального репозитория, а есть только копия данных с сервера. В случае выхода из строя сервера команда не сможет продолжить работу над проектом. Еще один минус использования централизованной системы управления версиями заключается в сложности доступа к одному ресурсу одновременно несколькими разработчиками.

Примеры централизованных VCS: CVS (Система одновременных версий), Subversion (SVN).

В децентрализованной (распределенной) системе управления версиями, кроме версии репозитория на сервере, у каждого разработчика есть свой репозиторий. 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.

Примеры распределенных VCS: Git, Mercurial.

4. Опишите действия с VCS при единоличной работе с хранилищем.

Ответ: На самом деле действий может быть очень много. Например, можно делать commit, хранить файлы проекта в репозитории, делать релизы

5. Опишите порядок работы с общим хранилищем VCS.

Ответ: Пользователь перед началом работы с помощью определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. Можно объединить (слить) изменения, сделанные разными участниками (сделать это автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения.

6. Каковы основные задачи, решаемые инструментальным средством git?

Ответ: У Git есть две основные задачи: хранить информацию обо всех изменениях в коде, начиная с самой первой строчки, и обеспечить удобства командной работы над кодом.

7. Назовите и дайте краткую характеристику командам git.

Ответ:

- 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. Приведите примеры использования при работе с локальным и удалённым репозиториями.

Ответ:

- Если в проекте закрыт исходный код, то здесь удобно использовать локальный репозиторий. Также локальный репозиторий удобен для одиночного проекта.
- Проекты с открытым кодом или выполняемые группой людей удобно хранить на удалённом репозитории

9. Что такое и зачем могут быть нужны ветви (branches)?

Ответ: 'Git branch' – это команда для управления ветками в репозитории Git. Ветка – это просто «скользящий» указатель на один из коммитов. Когда мы создаём новые коммиты, указатель ветки автоматически сдвигается вперёд, к вновь созданному коммиту. Ветки используются для разработки одной части функционала изолированно от других. Каждая ветка представляет собой отдельную копию кода проекта. Ветки позволяют одновременно работать над разными версиями проекта.

10. Как и зачем можно игнорировать некоторые файлы при commit?

Ответ: Игнорируемые файлы – это, как правило, специфичные для платформы файлы или автоматически созданные файлы из систем сборки. .gitignore – это простой текстовый файл, в каждой строке которого содержится шаблон файла или каталога, который необходимо проигнорировать. Строки, начинающиеся со знака #, являются комментариями и игнорируются. Пустые строки могут быть использованы для улучшения читабельности файла и группировки связанных строк шаблонов.