

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

Отчет по лабораторной работе №2:
“Лабораторная работа № 2. Управление версиями”

Дисциплина: Операционные системы
Автор: Шмырин Михаил Сергеевич
Номер группы: НПМбд-02-21
Преподаватель: Кулябов Дмитрий Сергеевич

Москва

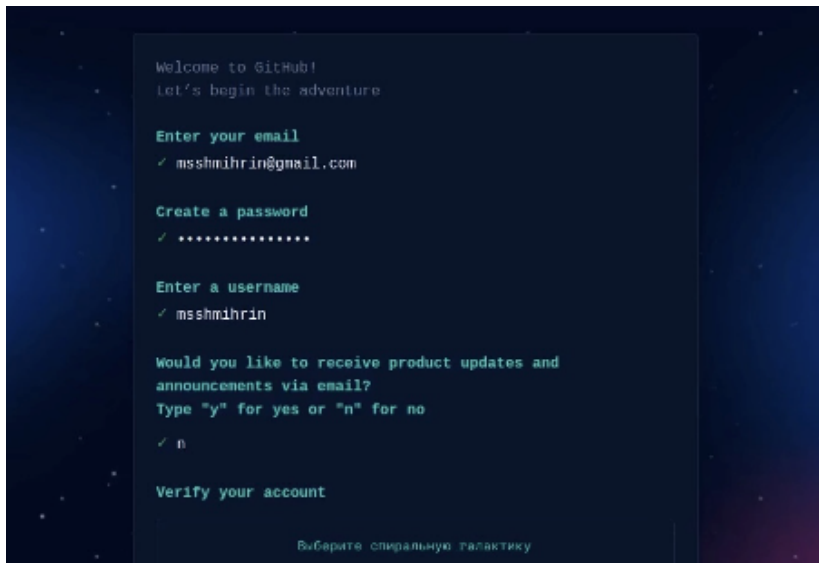
2022г.

2.1. Цель работы

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

2.6. Последовательность выполнения работы

2.6.1. Настройка github и заполнение данных(Рис 1.)



2.6.2. Установка программного обеспечения (Рис 2.)

2.6.2.1. Установка git-flow в Fedora Linux с помощью нескольких команд(Рис 2.)

```
[msshmihrin@msshmihrin ~]$ cd /tmp
[msshmihrin@msshmihrin tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[msshmihrin@msshmihrin tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow-installer.sh
[msshmihrin@msshmihrin tmp]$ chmod +x gitflow-installer.sh
[msshmihrin@msshmihrin tmp]$ sudo ./gitflow-installer.sh install stable

Мы полагаем, что ваш системный администратор изложил вам основы
безопасности. Как правило, всё сводится к трём следующим правилам:

#1) Уважайте частную жизнь других.
#2) Думайте, прежде что-то вводить.
#3) С большой властью приходит большая ответственность.

[sudo] пароль для msshmihrin:
## git-flow no-make installer ##
Installing git-flow to /usr/local/bin
Cloning repo from GitHub to gitflow
Клонирование в «gitflow»...
remote: Enumerating objects: 4270, done.
remote: Total 4270 (delta 0), reused 0 (delta 0), pack-reused 4270
Получение объектов: 100% (4270/4270), 1.74 МиБ | 2.16 МиБ/с, готово.
Определение изменений: 95% (2407/2533)
```

Figure 2: Рис 2.

2.6.2.2. Установка gh в Fedora Linux (Рис 3.)

```
[msshmihrin@msshmihrin tmp]$ sudo dnf install gh
Fedora 35 - x86_64 - Updates                28 kB/s | 20 kB    00:00
Fedora Modular 35 - x86_64 - Updates       37 kB/s | 19 kB    00:00
Зависимости разрешены.
=====
Пакет                Архитектура          Версия                Репозиторий          Размер
=====
Установка:
gh                   x86_64                2.7.0-1.fc35          updates               6.8 M
=====
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 6.8 M
Объем изменений: 32 M
Продолжить? [д/н]: y
Загрузка пакетов:
gh-2.7.0-1.fc35.x86_64.rpm                8.0 MB/s | 6.8 MB    00:00
-----
Общий размер                5.0 MB/s | 6.8 MB    00:01
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка      : 1/1
Установка       : gh-2.7.0-1.fc35.x86_64 1/1
Запуск скрипта : gh-2.7.0-1.fc35.x86_64 1/1
Проверка        : gh-2.7.0-1.fc35.x86_64 1/1

Установлен:
gh-2.7.0-1.fc35.x86_64

Всего выполнено:
```

Figure 3: Рис 3.

2.6.3. Базовая настройка git (Рис 4.)

```
[msshmihrin@msshmihrin tmp]$ git config --global user.name "Mikhail Shmyrin"  
[msshmihrin@msshmihrin tmp]$ git config --global user.name "msshmihrin@gmail.com"  
[msshmihrin@msshmihrin tmp]$ git config --global core.quotepath false  
[msshmihrin@msshmihrin tmp]$ git config --global unit.defaultBranch master  
[msshmihrin@msshmihrin tmp]$ git config --global core.autocrlf input  
[msshmihrin@msshmihrin tmp]$ git config --global core.safecrlf warn  
[msshmihrin@msshmihrin tmp]$ ssh-keygen -t rsa -b 4096
```

Figure 4: Рис 4.

2.6.4. Создайте ключи ssh (Рис 5.)

```
[msshmihrin@msshmihrin tmp]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/msshmihrin/.ssh/id_rsa): ssh-keygen -t ed25519
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ssh-keygen -t ed25519
Your public key has been saved in ssh-keygen -t ed25519.pub
The key fingerprint is:
SHA256:mg++hqeDTNmFzIW0v4nDMs6TBjvcYDvs2v0E VWORafQ msshmihrin@msshmihrin
The key's randomart image is:
+---[RSA 4096]-----+
|      .oo      |
|      +o       |
|      .+ E      |
|      o .       |
|      o.. .S     |
| .oooo oo.      |
| +==+.+++      |
| oBXo=o0+      |
| +==*+=o.      |
| +-----[SHA256]-----+
```

2.6.5. Создайте ключи pgr (Рис 6.)

```
[msshmihrin@msshmihrin tmp]$ gpg --full-generate-key
gpg (GnuPG) 2.3.2; Copyright (C) 2021 Free Software Foundation, Inc.
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/msshmihrin/.gnupg'
gpg: создан щит с ключами '/home/msshmihrin/.gnupg/pubring.kbx'
Выберите тип ключа:
```

- (1) RSA and RSA
- (2) DSA and Elgamal
- (3) DSA (sign only)
- (4) RSA (sign only)
- (5) ECC (sign and encrypt)

2.6.6. Добавление PGP ключа в GitHub (Рис 7. и Рис 8. и Рис 9.)

```
ewe23pgg: /home/msshmihrin/.gnupg/trustdb.gpg: создана таблица доверия
pgp: ключ 0D17016924F71787 помечен как абсолютно доверенный
pgp: создан каталог '/home/msshmihrin/.gnupg/openpgp-revocs.d'
pgp: сертификат отзыва записан в '/home/msshmihrin/.gnupg/openpgp-revocs.d/AE211AED2033E9D5479506180D17016924F71787.rev'.
открытый и секретный ключи созданы и подписаны.
```

```
pub  rsa4096 2022-04-23 [SC]
     AE211AED2033E9D5479506180D17016924F71787
uid      Mikhail Shmyrin <msshmihrin@gmail.com>
sub  rsa4096 2022-04-23 [E]

[msshmihrin@msshmihrin tmp]$ gpg --list-secret-keys --keyid-format LONG
pgp: проверка таблицы доверия
pgp: marginals needed: 3  completes needed: 1  trust model: pgp
pgp: глубина: 0  достоверных: 1  подписанных: 0  доверие: 0-, 0q, 0n, 0m, 0f, 1u
/home/msshmihrin/.gnupg/pubring.kbx
-----
sec  rsa4096/0D17016924F71787 2022-04-23 [SC]
     AE211AED2033E9D5479506180D17016924F71787
uid      [  абсолютно ] Mikhail Shmyrin <msshmihrin@gmail.com>
ssb  rsa4096/6FF0D20A4A27564A 2022-04-23 [E]
```

```
[msshmihrin@msshmihrin tmp]$ gpg --armor --export 0D17016924F71787 | xclip -sel clip
```

```
mQINBG3KJHEBEADhxwEokad8V34KAb5I8Zz2GDeazMyASv6lm8tDFXm1NvHy8oQh
1hGaGzp7LvyAruSgcjo5Z3gPKvpXdrJ9AijymB1Pimyej30DhzQVTutmf2cn1nv
QK51IUP0LUUwRrAE1VGH6uu12VHbtTos72oBcYFP20zQyh1oVDvy+9fwIAUnP1
fFMjnmJ7xpKSEURZHKheB0rsgPiHj4p2y6P9g3CP3V452hh5pMRA+sGdf85pS/
QxtrFTwS14B0AgznGGIzSJXUrwDZCZan8TqvUEKqeoPAQBvE18d2HZm28BYCQM
wWqNzW9uG5GySC9HFR+mr1aL383uf6ymfPoUCPgk3fYVpp3mUdoVnTf/6KJ3BqmEW7
JnkLLKanH/a3xfqgmKTdw9XbesukMSeV1rPsA+arzy0rzsxbxUatrBYRgrBi77m
7fr1Ub4fGEe2NDv9y8xKi1kQc9wX66g0Kp7EfWzqfNbVw2iLfoVZtKhfN8ZftQ+T
1t+BA+mhMIxgln1oR+kmpRozLjquQHbVv8z5+E6NFGwcGJ0QmSKE+pP85Bz1bYt1l
g5qV41Pvk6y3JId5SM4BkQomLxjfe4T9HjoGUCxXfikaXm5Bfo+EvPdYLwM4Qo/p
cA9eEgZxgrBxbp3x4DHa5ZWn/M5S05jo8AA5BknR173sTIQs9Ie0JEJAwARAQAB
CZNaWtoWVlsIFNobXlyYW4gPG1zc2htawhyaW5AZ21haWwUy29tPocKUGtQAQgA
PB9YhBk4Gu0gM+nVR5UGGA0XAwkk9xeHBQ31ZCRxAhsDBQsJCAcCAyICAQYVCgkI
CwIEFgIDAQIEBwIXgAAKCRANFwFpJPCxH5N2D/4yEwpUK5Vq/ccmLj5+GD24u6xc
je0x//Z8Tnw4iagNuteWjNs1I2EqGNYVfootLxZjUQM2L9+nbQqNzgJcUvXA6fk4
9tIQo/qLWw2W6LPcc6QD8rKhfJubr/Wsio+oW5wFH9Hcu70PevA0XmBbJkPS6nL
8Kb56Yy/hFYwM+4oFeaSIAX1xc0KyW6ym7qf2o0CJ7NT0Y5MSyW0DoYNCKaqjzPb
```


И вставляем его на гитхаб (Рис 10.)

GPG keys / Add new

Key

```
0z5aBU0vsJDBVslPVm10SU7OjUKRPuiXNMbIJ9+Zl62of1QnpUbh9NByXUSELbz
Ez7nFlxwtw3N2VfXlJa7MBqbb03EsleMIEGzZYHgHh3FvQJSNRoyT0BqV9+r2apl
AAqTV978SWgs/XQzY7qexPvUG+H6MXGYZhrNmMR3Gh+J1B160s8bGBMCL+TB9cR
QEFBOEGalmI6LTJ/QqiQ2PixMASVK7uXLm0Odyawe94AvSi0I6ohuwgHNSy2wKAi
bgvoHqGXnahQTE8MIIPiZzPys/GkMT0C2Wkf0JUy2QyQmQLHuF3Mhm8Dvw9k9M8M
YD/7N8Hr
~IGuy
-----END PGP PUBLIC KEY BLOCK-----
```

Add GPG key

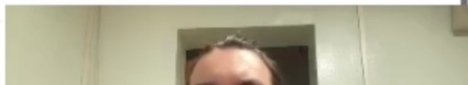


Figure 5: Рис 10.

2.6.7. Настройка автоматических подписей коммитов git(Рис 11.)

```
[msshmihrin@msshmihrin tmp]$ git config --global user.signingkey 0D17016924F71787  
[msshmihrin@msshmihrin tmp]$ git config --global commit.gpgsign true  
[msshmihrin@msshmihrin tmp]$ git config --global gpg.program $(which gpg2)
```

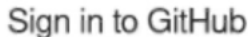
Figure 6: Рис 11.

2.6.8. Настройка gh (авторизация через браузер)(Рис 12-15.)

```
[msshmihrin@msshmihrin tmp]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? 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: D94F-287D
Press Enter to open github.com in your browser...
restorecon: SELinux: Could not get canonical path for /home/msshmihrin/.mozilla/firefox/* restorecon: No such file or directory.
[
[msshmihrin@msshmihrin tmp]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? 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: D94F-287D
Press Enter to open github.com in your browser...
restorecon: SELinux: Could not get canonical path for /home/msshmihrin/.mozilla/firefox/* restorecon: No such file or directory.
```



Username or email address

msshmihrin

Password

[Forgot password?](#)

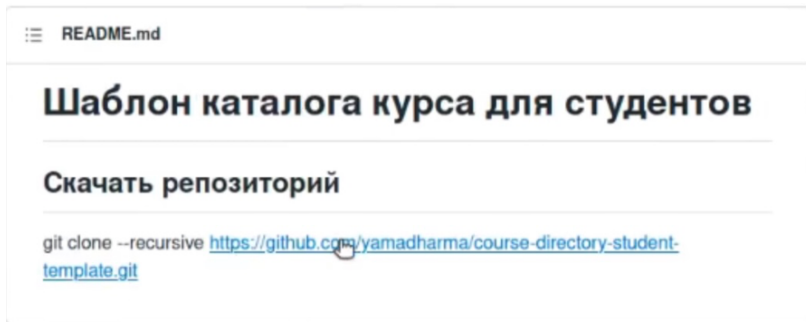
●●●●●●●●●●●●●●●●

2.6.9. Шаблон для рабочего пространства(Рис 16.)

```
? How would you like to authenticate GitHub CLI? Login with a web browser
! First copy your one-time code: D94F-287D
Press Enter to open github.com in your browser...
restorecon: SELinux: Could not get canonical path for /home/msshmihrin/.mozilla/firefox/* restorecon: No such file or directory.
✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as msshmihrin
```

Figure 7: Рис 16.

2.6.9.1. Сознание репозитория курса на основе шаблона(Рис 17-18)



```
[msshmihrin@msshmihrin ~]$ mkdir -p ~/work/study/2021-2022/"Операционные системы"  
[msshmihrin@msshmihrin ~]$ cd ~/work/study/2021-2022/"Операционные системы"  
[msshmihrin@msshmihrin Операционные системы]$ gh repo create study_2021-2022_o  
s-intro --template=yamadharma/course-directory-student-template --public  
GraphQL: Could not clone: Name already exists on this account (cloneTemplateRe  
pository)
```

2.6.9.2. Настройка каталога курса(Рис 19-20)

```
[msshmihrin@msshmihrin Операционные системы]$ git clone --recursive https://github.com/msshmihrin/study_2021-2022_os-intro.git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 20 (delta 2), reused 15 (delta 2), pack-reused 0
Получение объектов: 100% (20/20), 12.49 Киб | 3.12 Миб/с, готово.
Определение изменений: 100% (2/2), 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 «/home/msshmihrin/work/study/2021-2022/Операционные системы/os-intro/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 | 1.36 Миб/с, готово.
Определение изменений: 100% (9/9), готово.
Клонирование в «/home/msshmihrin/work/study/2021-2022/Операционные системы/os-intro/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 | 1.35 Миб/с, готово.
Определение изменений: 100% (31/31), готово.
Подмодуль по пути «template/presentation»: забрано состояние «3eae6b7586f8a9aded2b596cd1018e625b228b93»
Подмодуль по пути «template/report»: забрано состояние «df7b2ef80f8def3b9a496f8695277469a1a7842a»
```

```
[msshmihrin@msshmihrin Операционные системы]$ cd ~/work/study/2021-2022/"Операционные системы"/os-intro
[msshmihrin@msshmihrin os-intro]$ rm package.json
[msshmihrin@msshmihrin os-intro]$ make COURSE=os-intro
[msshmihrin@msshmihrin os-intro]$ git add .
[msshmihrin@msshmihrin os-intro]$ git commit -am 'feat(main): make course structure'
Author identity unknown
```

*** Пожалуйста, скажите мне кто вы есть.

Запустите

```
git config --global user.email "you@example.com"
git config --global user.name "Ваше Имя"
```

для указания идентификационных данных аккаунта по умолчанию.

Пропустите параметр --global для указания данных только для этого репозитория.

2.7. Вывод

Я изучил идеологию и применение контроля версий.

2.8. Контрольные вопросы

2.8.1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями. Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом.

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

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять неполную версию изменённых файлов, а производить так называемую дельта-компрессию—сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных. Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с

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

Централизованные системы — это системы, которые используют архитектуру клиент / сервер, где один или несколько клиентских узлов напрямую подключены к центральному серверу. Пример - Wikipedia. В децентрализованных системах каждый узел принимает свое собственное решение. Конечное поведение системы является совокупностью решений отдельных узлов. Пример — Bitcoin. В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером.

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

Создадим локальный репозиторий. Сначала сделаем предварительную конфигурацию, указав имя и email владельца репозитория: `git config --global user.name "Имя Фамилия"` `git config --global user.email "work@mail"` и настроив utf-8 в выводе сообщений `git config --global quotepath false` Для инициализации локального репозитория, расположенного, например, в каталоге `~/tutorial`, необходимо ввести в командной строке: `cd mkdir tutorial cd tutorial git init`

2.8.5. Опишите порядок работы с общим хранилищем VCS.

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый): `ssh-keygen -C "Имя Фамилия work@mail"` Ключи сохраняются в каталоге `~/.ssh/`. Скопировав из локальной консоли ключ в буфер обмена `cat ~/.ssh/id_rsa.pub | xclip -sel clip` вставляем ключ в появившееся на сайте поле.

2.8.6. Каковы основные задачи, решаемые инструментом git?

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.

2.8.7. Назовите и дайте краткую характеристику командам git.

Основные команды git: Наиболее часто используемые команды git: – 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 изменения: 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

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

Использования git при работе с локальными репозиториями
(добавления текстового документа в локальный репозиторий): `git
add hello.txt git commit -am 'Новый файл'`

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

Проблемы, которые решают ветки git: • нужно постоянно создавать архивы с рабочим кодом • сложно “переключаться” между архивами • сложно перетаскивать изменения между архивами • легко что-то напутать или потерять

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

Во время работы над проектом так или иначе могут создаваться файлы, которые не требуется добавлять в последствии в репозиторий. Например, временные файлы, создаваемые редакторами, или объектные файлы, создаваемые компиляторами. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл .gitignore с помощью сервисов. Для этого сначала нужно получить списки меняющихся шаблонов: `curl -L -s https://www.gitignore.io/api/list` Затем скачать шаблон, например, для С и С++ `curl -L -s https://www.gitignore.io/api/c » .gitignore` `curl -L -s https://www.gitignore.io/api/c++ » .gitignore`

2.9. Вывод

Я изучил идеологию и применение контроля версий.