

Отчет по лабораторной работе №1

Операционные системы

Клименко Алёна Сергеевна

Российский университет дружбы народов, Москва, Россия

Информация

- Клименко Алёна Сергеевна
- НКАбд-02-2024 № Студенческого билета: 1132246741
- Российский университет дружбы народов
- https://github.com/Alstrr/study_2024-2025_os-intro



Рис. 1: К

Целью данной работы является изучение идеологии и применения средств контроля версий и освоение умения по работе с git.

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

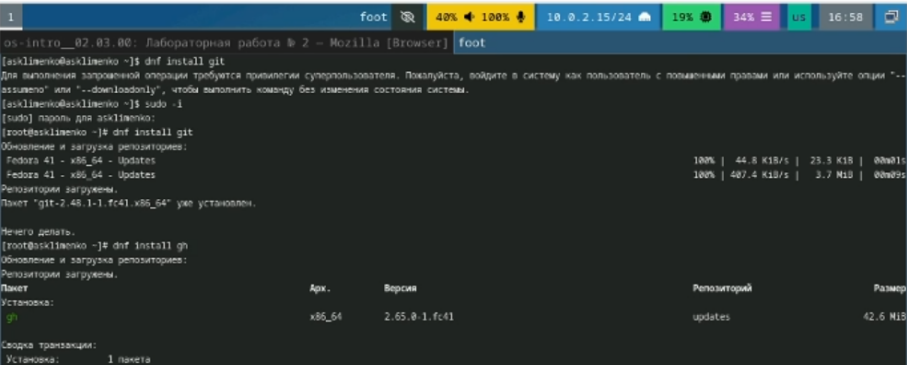
Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.

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 не полную версию изменённых файлов, а производить так называемую дельта-компрессию —

Установим гит командой `dnf install git`, установим gh командой `dnf install gh`

Базовая настройка git.

Открываем терминал. При помощи команд `git config --global user.name` и `git config --global user.email` зададим имя пользователя и адрес электронной почты. При помощи команды `git config --global core.quotepath false` настроим utf-8 в выводе сообщений git. При помощи команды `git config --global init.defaultBranch master` зададим начальной ветке имя master. (рис. 1)



```
1 | foot | 40% | 100% | 10.0.2.15/24 | 19% | 34% | us | 16:58 |
os-intro_02.03.00: Лабораторная работа № 2 – Mozilla [Browser] | foot
[asklimenko@asklimenko ~]$ dnf install git
для выполнения запрошенной операции требуется привилегия суперпользователя. Пожалуйста, войдите в систему как пользователь с полными правами или используйте опции "--assumeno" или "--downloadonly", чтобы выполнить команду без изменения состояния системы.
[asklimenko@asklimenko ~]$ sudo -i
[sudo] пароль для asklimenko:
[root@asklimenko ~]$ dnf install git
Обновление и загрузка репозитория:
Fedora 41 - x86_64 - Updates | 100% | 44.8 KiB/s | 23.3 KiB | 00m01s
Fedora 41 - x86_64 - Updates | 100% | 487.4 KiB/s | 3.7 MiB | 00m09s
Репозитории загружены.
Пакет "git-2.48.1-1.fc41.x86_64" уже установлен.

Нечего делать.
[root@asklimenko ~]$ dnf install gh
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет | Арх. | Версия | Репозиторий | Размер
Установка:
gh | x86_64 | 2.65.0-1.fc41 | updates | 42.6 MiB

Сводка транзакции:
Установка: 1 пакета
```

Создание ssh ключа.

создания ключа (рис. 2)

```
1 foot 40% 100% 10.0.2.15/24 45% 35% US 17:08
os-intro_02.03.00: Лабораторная работа № 2 – Mozilla [Browser] foot
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: Alena Klivenko
Адрес электронной почты: 1132246741@pfur.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
  "Alena Klivenko <1132246741@pfur.ru>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход?
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, движение мыши, обращение 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, движение мыши, обращение 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-sevocs.d"
gpg: сертификат отмены записан в "/root/.gnupg/openpgp-sevocs.d/2EDCC29A845A8C0805F29E2128CE98EE98906F82.rev".
открытый и секретный ключи созданы и подписаны.

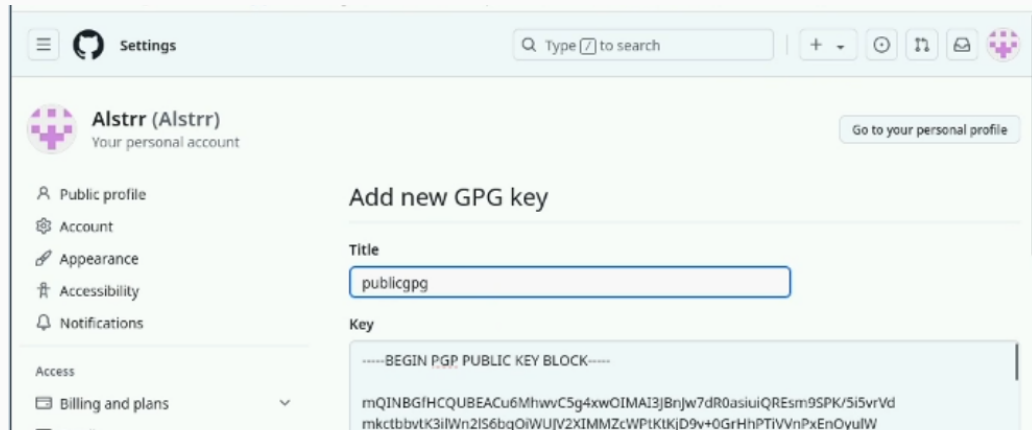
pub rsa4096 2025-03-04 [SC]
    2EDCC29A845A8C0805F29E2128CE98EE98906F82
uid                               Alena Klivenko <1132246741@pfur.ru>
sub rsa4096 2025-03-04 [E]

[root@asklivenko ~]# 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, 0a, 0f, 1u
[keyboard]
-----
-----
```

Добавление gpg-ключа в учетную запись ГитХаб.

Копируем созданный ключ и переносим его на сайт гитхаб в раздел ssh и gpg keys.

Создаем новый ключ, задаем ему название и переносим ключ в поле key, добавляем ключ на сайт. (рис. 3)



The screenshot shows the GitHub 'Settings' page for a user named 'Alstrr (Alstrr)'. The page title is 'Add new GPG key'. The 'Title' field is filled with 'publicgpg'. The 'Key' field contains a GPG public key block starting with '-----BEGIN PGP PUBLIC KEY BLOCK-----' and ending with 'mkctbbvtK3iIWn2IS6bqOiWUjV2XIMMZcWPtKtKjD9v+0GrHhPTiVnPxEnOvuIW'. The left sidebar shows the user's profile and navigation links: Public profile, Account, Appearance, Accessibility, Notifications, Access, and Billing and plans. The top navigation bar includes the GitHub logo, the word 'Settings', a search bar, and several icons.

Settings

Alstrr (Alstrr)
Your personal account

Go to your personal profile

Public profile
Account
Appearance
Accessibility
Notifications

Access

Billing and plans

Add new GPG key

Title

publicgpg

Key

```
-----BEGIN PGP PUBLIC KEY BLOCK-----  
  
mQINBGfHCQUBEACu6MhwvC5g4xwOIMA3jBnJw7dR0asiuiQREsm9SPK/5i5vrVd  
mkctbbvtK3iIWn2IS6bqOiWUjV2XIMMZcWPtKtKjD9v+0GrHhPTiVnPxEnOvuIW
```

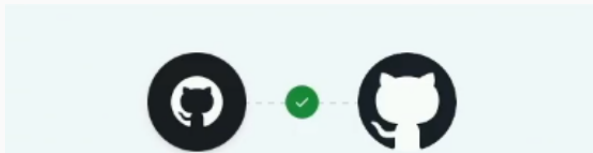
Создание PGP ключа.

создаем шаблон рабочего пространства (рис. 4)

```
[root@asklinenko ~]# mc  
[root@asklinenko ~]# mc  
[root@asklinenko ~]# mkdir -p ~/work/study/2022-2023/"Операционные системы"  
[root@asklinenko ~]# mkdir -p ~/work/study/2024-2025/"Операционные системы"  
[root@asklinenko ~]# cd ~/work/study/2024-2025/"Операционные системы"  
[root@asklinenko Операционные системы]# gh repo create study_2022-2023_os-intro --template-yamadharma/course-directory-student-template --public
```

Рис. 5: Создание шаблона

связываем аккаунт гит с виртуальной машиной (рис. 5)

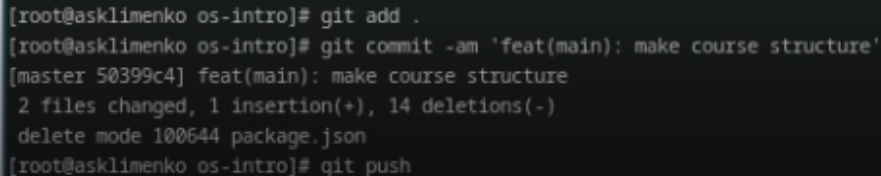


создаем папки для лабараторных работ (рис. 6)

```
[root@asklimenko ~]# cd ~/work/study/2024-2025/Операционные\ системы/os-intro/
[root@asklimenko os-intro]# ls
CHANGELOG.md config COURSE LICENSE Makefile package.json README.en.md README.git-flow.md README.md template
[root@asklimenko os-intro]# rm package.json
rm: удалить обычный файл 'package.json'? y
[root@asklimenko os-intro]# echo os-intro > COURSE
[root@asklimenko os-intro]#
```

Рис. 7: Новый ключ PGP

выгружаем информацию в git (рис. 7)

A terminal window with a dark background and light gray text. The text shows a sequence of git commands and their output. The first command is 'git add .' which is partially visible. The second command is 'git commit -am 'feat(main): make course structure''. The output shows the commit hash '50399c4', the commit message 'feat(main): make course structure', and a summary of changes: '2 files changed, 1 insertion(+), 14 deletions(-)'. It also shows a file deletion: 'delete mode 100644 package.json'. The third command is 'git push', which is followed by a cursor.

```
[root@asklimenko os-intro]# git add .  
[root@asklimenko os-intro]# git commit -am 'feat(main): make course structure'  
[master 50399c4] feat(main): make course structure  
 2 files changed, 1 insertion(+), 14 deletions(-)  
  delete mode 100644 package.json  
[root@asklimenko os-intro]# git push  
█
```

Рис. 8: Настройка необходимых подписей коммитов

Создание и настройка репозитория курса.

Отправляем файлы первой лабораторной работы на сервер. (рис. 8)

The screenshot shows a GitHub repository page for 'study_2024-2025_os-intro' by user 'Alstrr'. The repository is public and was generated from a template. The file list shows an initial commit 25 minutes ago with files: config, template, .gitattributes, .gitignore, and .gitmodules. The right sidebar shows the repository has no description, 0 stars, 1 watcher, and 0 forks.

Alstrr / study_2024-2025_os-intro

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

study_2024-2025_os-intro Public

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

Pin Unwatch 1 Fork 0 Star 0

master Go to file Code

Alstrr	Initial commit	de48c3b · 25 minutes ago	1 Commit
config	Initial commit	25 minutes ago	
template	Initial commit	25 minutes ago	
.gitattributes	Initial commit	25 minutes ago	
.gitignore	Initial commit	25 minutes ago	
.gitmodules	Initial commit	25 minutes ago	

About

No description, website, or topics provided.

- Readme
- CC-BY-4.0 license
- Activity
- 0 stars
- 1 watching
- 0 forks

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

Система контроля версий — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями 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. Системы контроля версий (Version Control System, VCS) применяются для:

- Хранение полной истории изменений
- причин всех производимых изменений
- Откат изменений, если что-то пошло не так
- Поиск причины и ответственного за появления ошибок в программе
- Совместная работа группы над одним проектом
- Возможность изменять код, не мешая работе других пользователей

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

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

Централизованные VCS (Subversion; CVS; TFS; VAULT; AccuRev): Одно основное хранилище всего проекта. Каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет и, затем, добавляет свои изменения обратно. Децентрализованные VCS (Git; Mercurial; Bazaar): У каждого пользователя свой вариант (возможно не один) репозитория. Присутствует возможность добавлять и забирать изменения из любого репозитория. В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

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

Сначала создаем и подключаем удаленный репозиторий. Затем по мере изменения проекта отправлять эти изменения на сервер.

5. Опишите порядок работы с общим хранилищем VCS.

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.

6. Каковы основные задачи, решаемые инструментальным средством git?

Первая — хранить информацию о всех изменениях в вашем коде, начиная с самой первой строки, а вторая — обеспечение удобства командной работы над кодом.

7. Назовите и дайте краткую характеристику командам 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 изменений: `git diff`
- сохранение текущих изменений: – добавить все изменённые

17/20

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

`git push -all (push origin master/любой branch)`

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

Ветвление («ветка», branch) — один из параллельных участков истории в одном хранилище, исходящих из одной версии (точки ветвления). [3] • Обычно есть главная ветка (master), или ствол (trunk). • Между ветками, то есть их концами, возможно слияние. Используются для разработки новых функций.

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

В результате выполнения данной лабораторной работы я приобрела необходимые навыки работы с гит, научилась созданию репозитория, gpg и ssh ключей, настроила каталог курса и авторизовалась в gh.

