

```

---
## Front matter
title: "Отчет лабораторная работа №2"
subtitle: "Операционные системы"
author: "Шабакова Карина Баировна"

## Generic options
lang: ru-RU
toc-title: "Содержание"

## Bibliography
bibliography: bib/cite.bib
csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl

## Pdf output format
toc: true # Table of contents
toc-depth: 2
lof: true # List of figures
lot: true # List of tables
fontsize: 12pt
linestretch: 1.5
papersize: a4
documentclass: scrreprt
## I18n polyglossia
polyglossia-lang:
  name: russian
  options:
    - spelling=modern
    - babelshorthands=true
polyglossia-otherlangs:
  name: english
## I18n babel
babel-lang: russian
babel-otherlangs: english
## Fonts
mainfont: IBM Plex Serif
romanfont: IBM Plex Serif
sansfont: IBM Plex Sans
monofont: IBM Plex Mono
mathfont: STIX Two Math
mainfontoptions: Ligatures=Common,Ligatures=TeX,Scale=0.94
romanfontoptions: Ligatures=Common,Ligatures=TeX,Scale=0.94
sansfontoptions: Ligatures=Common,Ligatures=TeX,Scale=MatchLowercase,Scale=0.94
monofontoptions: Scale=MatchLowercase,Scale=0.94,FakeStretch=0.9
mathfontoptions:
## Biblatex
biblatex: true
biblio-style: "gost-numeric"
biblatexoptions:
  - parenttracker=true
  - backend=biber
  - hyperref=auto
  - language=auto
  - autolang=other*
  - citestyle=gost-numeric
## Pandoc-crossref LaTeX customization
figureTitle: "Рис."
tableTitle: "Таблица"
listingTitle: "Листинг"
lofTitle: "Список иллюстраций"
lotTitle: "Список таблиц"
lolTitle: "Листинги"
## Misc options
indent: true

```

```
header-includes:
- \usepackage[indentfirst]
- \usepackage{float} # keep figures where there are in the text
- \floatplacement{figure}{H} # keep figures where there are in the text
---
```

#### # Цель работы

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

#### # Задание

- 1.Выполнение лабораторной работы
- 2.Выводы
- 4.Контрольные вопросы
- 3.Список литературы

# Выполнение лабораторной работы  
Установим git: (рис. [-@fig:001]).

![1](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/1.jpg){#fig:001 width=70%}

Установка gh (рис. [-@fig:002]).

![2](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/2.jpg){#fig:002 width=70%}

Задала имя и email владельца репозитория:(рис. [-@fig:003]).

![3](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/3.jpg){#fig:003 width=70%}

Настраиваю utf-8 в выводе сообщений git: (рис. [-@fig:004]).

![4](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/4.png){#fig:004 width=70%}

Задала имя начальной ветки (будем называть её master):(рис. [-@fig:005]).

![5](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/5.png){#fig:005 width=70%}

Параметр autocrlf: (рис. [-@fig:006]).

![6](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/6.jpg){#fig:006 width=70%}

Параметр safecrlf:(рис. [-@fig:007]).

![7](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/7.jpg){#fig:007 width=70%}

по алгоритму rsa с ключём размером 4096 бит: (рис. [-@fig:008]).

![8](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/8.jpg){#fig:008 width=70%}

по алгоритму ed25519: (рис. [-@fig:009]).

![9](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/9.jpg){#fig:009 width=70%}

Генерирую ключ(рис. [-@fig:010]).

![10](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/10.jpg){#fig:010 width=70%}

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. [-@fig:001]).

![Название рисунка](image/placeimg\_800\_600\_tech.jpg){#fig:001 width=70%}

Вывожу список ключей и копируем отпечаток приватного ключа:(рис. [-@fig:011]).

![11](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/11.jpg){#fig:011 width=70%}

Скопируйте ваш сгенерированный PGP ключ в буфер обмена: (рис. [-@fig:012]).

![12](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/12.png){#fig:012 width=70%}

Перехожу в настройки GitHub, нажала на кнопку New GPG key и вставила полученный ключ в поле ввода. (рис. [-@fig:013]).

![13](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/13.jpg){#fig:013 width=70%}

Используя введённый email, укажите Git применять его при подписи коммитов: (рис. [-@fig:014]).

![14](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/14.jpg){#fig:014 width=70%}

Для начала авторизовалась(рис. [-@fig:015]).

![15](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/15.jpg){#fig:015 width=70%}

Авторизовалась через браузер. (рис. [-@fig:016]).

![16](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/16.jpg){#fig:016 width=70%}

Создала шаблон рабочего пространства для 2024-2025 учебного года и предмета «Операционные системы» (код предмета os-intro) создание репозитория примет следующий вид:(рис. [-@fig:017]). (рис. [-@fig:018]).

![17](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/17.jpg){#fig:017 width=70%}

![18](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/18.jpg){#fig:018 width=70%}

Перешла в каталог курса:(рис. [-@fig:019]).

![19](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/19.png){#fig:019 width=70%}

Удалила лишние файлы и Создала необходимые каталоги:.(рис. [-@fig:020]).

![20](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/20.jpg){#fig:020 width=70%}

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. [-@fig:021]).

![21](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/21.jpg){#fig:021 width=70%}

Отправила файлы на сервер:(рис. [-@fig:022]).

![22](/home/kbshabakova/work/study/2024-2025/Операционные системы/os-intro/labs/lab02/report/image/22.jpg){#fig:022 width=70%}

## # Выводы

При выполнении данной лабораторной работы я изучила идеологию и применение средств контроля версий, освоила умение по работе с git.

## # 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.

# Список литературы{.unnumbered}

::: {#refs}

1. Лабораторная работа №2[Электронный ресурс]

URL: <https://esystem.rudn.ru/mod/page/view.php?id=1224371>

:::