

Отчёт по лабораторной работе №2

Управление версиями

Иван Горбунов

Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
3 Вывод	17
4 Контрольные вопросы	18

Список иллюстраций

2.1	Загрузка пакетов	7
2.2	Параметры репозитория	8
2.3	rsa-4096	9
2.4	ed25519	10
2.5	GPG ключ	11
2.6	GPG ключ	12
2.7	Параметры репозитория	13
2.8	Связь репозитория с аккаунтом	14
2.9	Загрузка шаблона	15
2.10	Первый коммит	16

Список таблиц

1 Цель работы

Целью данной работы является изучение идеологии и применения средств контроля версий и освоение умений работать с git.

2 Выполнение лабораторной работы

Устанавливаем git, git-flow и gh.

```
imgorbunov@imgorbunov:~$ git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy-fetch]
           [--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]
           [--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=<envvar>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv        Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm        Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect    Use binary search to find the commit that introduced a bug
  diff      Show changes between commits, commit and working tree, etc
  grep      Print lines matching a pattern
```

Рисунок 2.1: Загрузка пакетов

Зададим имя и email владельца репозитория, кодировку и прочие параметры.

```
igorbunov@igorbunov:~$  
igorbunov@igorbunov:~$ git config --global user.name "im-gorgunov"  
igorbunov@igorbunov:~$ git config --global user.email "1032254762@yandex.ru"  
igorbunov@igorbunov:~$ git config --global core.quotepath false  
igorbunov@igorbunov:~$ git config --global init.defaultBranch master  
igorbunov@igorbunov:~$ git config --global core.autorlfi input  
igorbunov@igorbunov:~$ git config --global core.safearlf warn  
igorbunov@igorbunov:~$
```

Рисунок 2.2: Параметры репозитория

Создаем SSH ключи

```
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/imgorbunov/.ssh/id_rsa):  
Created directory '/home/imgorbunov/.ssh'.  
Enter passphrase for "/home/imgorbunov/.ssh/id_rsa" (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/imgorbunov/.ssh/id_rsa  
Your public key has been saved in /home/imgorbunov/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:tNwTN3sVE0gRi2Hf9whBupKJbqF8k8ibAIYzxIfHjvA imgorbunovThe key's randomart image is:  
+---[RSA 4096]----+  
|          +++. |  
| . o     ..+oo 0 |  
| .+ +   . o.+o .o |  
|+=      + = + + oo |  
|=oE.  o S + . o . |  
|.+ o + o . . . |  
| . = *       |  
| . = .       |  
|   o         |  
+---[SHA256]----+  
imgorbunov
```

Рисунок 2.3: rsa-4096

```
imgorbunov@imgorbunov:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/imgorbunov/.ssh/id_ed25519):
Enter passphrase for "/home/imgorbunov/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/imgorbunov/.ssh/id_ed25519
Your public key has been saved in /home/imgorbunov/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:jPSrN0mL0gD9/FmojSkosOPfJdVMxU5PpvPa4lap9nU imgorbunov@imgorbunov
The key's randomart image is:
+--[ED25519 256]--+
|          ..      |
|         .0 0     |
|       . . .0 =   |
|     . . . ++ + . |
|    . . oSo o . |
|   . . o o o. +  |
|  .o . B *.o = .E|
| = . * B.= * ... |
|oo.o.+.. .+o.    |
+---[SHA256]-----+
imgorbunov@imgorbunov:~$
```

Рисунок 2.4: ed25519

Создаем GPG ключ

```
Change (N)ame, (C)oмment, (E)mail or (O)kay/(Q)uit? 0
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.

gpg: /home/imgorбunov/.gnupg/trustdb.gpg: trustdb created
gpg: directory '/home/imgorбunov/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/imgorбunov/.gnupg/openpgp-revocs.d/DF15493301DAD810A70C0C
5EA50B54D0238F3E4F.rev'
public and secret key created and signed.

pub    rsa4096 2026-02-23 [SC]
      DF15493301DAD810A70C0C5EA50B54D0238F3E4F
uid          1032254762@rudn.ru
sub    rsa4096 2026-02-23 [E]

imgorбunov@imgorбunov:~$
```

Рисунок 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

```
imgorbunov@imgorbunov:~$  
imgorbunov@imgorbunov:~$ gpg --list-secret-keys --keyid-format LONG  
gpg: checking the trustdb  
gpg: marginals needed: 3 completes needed: 1 trust model: pgp  
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u  
[keyboxd]  
-----  
sec rsa4096/A50B54D0238F3E4F 2026-02-23 [SC]  
      DF15493301DAD810A70C0C5EA50B54D0238F3E4F  
uid          [ultimate] 1032254762@rudn.ru  
ssb rsa4096/AF3B2E439FF7A16F 2026-02-23 [E]  
  
imgorbunov@imgorbunov:~$ gpg --armor --export A50B54D0238F3E4F | xclip -sel clip  
imgorbunov@imgorbunov:~$
```

Рисунок 2.6: GPG ключ

Настройка автоматических подписей коммитов git

SSH keys

New SSH key

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

Authentication keys



1

SHA256:c/rJ7dUtDBeHfcYC2rB1zEyZy0fBxVGYUPnFkXv+QA



Added on Dec 27, 2025

Last used within the last 2 months — Read/write

Delete

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

GPG keys

New GPG key

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



1

Email addresses:

Key ID: A50B54D0238F3E4F



Subkeys: AF3B2E439F57A16F

Added on Feb 23, 2026

Delete

Рисунок 2.7: Параметры репозитория

Настройка gh

```
imgorunov@imgorunov: ~$ gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? SSH
? Upload your SSH public key to your GitHub account? /home/imgorunov/.ssh/id_rsa.pub
? Title for your SSH key: GitHub CLI
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 4F49-089A
Press Enter to open https://github.com/login/device in your browser...
✓ Authentication complete.
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Uploaded the SSH key to your GitHub account: /home/imgorunov/.ssh/id_rsa.pub
✓ Logged in as im-gorgunov
imgorunov@imgorunov: ~$
```

Рисунок 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация

```
remote: Total 287 (delta 125), reused 234 (delta 72), pack-reused 0 (from 0)
Receiving objects: 100% (287/287), 785.00 KiB | 3.83 MiB/s, done.
Resolving deltas: 100% (125/125), done.

Submodule path 'template/presentation': checked out '60a6091ebedbb01a8d98f2c5fae0f98dc63ad031'
Submodule path 'template/report': checked out '708b8c51acd94d4b78e7634198f221e5decb6355'

imgorbunov@imgorbunov:~/work/study/2025-2026/Операционные системы$ 
imgorbunov@imgorbunov:~/work/study/2025-2026/Операционные системы$ cd 2026-1--study--os-intro/
imgorbunov@imgorbunov:~/work/study/2025-2026/Операционные системы/2026-1--study--os-intro$ echo os-intro > COURSE
imgorbunov@imgorbunov:~/work/study/2025-2026/Операционные системы/2026-1--study--os-intro$ make prepare
Synchronizing submodule url for 'template/report'
Synchronizing submodule url for 'template/presentation'
Synchronizing submodule url for 'template/presentation'
Synchronizing submodule url for 'template/report'

imgorbunov@imgorbunov:~/work/study/2025-2026/Операционные системы/2026-1--study--os-intro$ ls
CODE_OF_CONDUCT.md  labs      package.json  project-personal  README.md
CONTRIBUTING.md    LICENSE   prepare       README.en.md     SECURITY.md
COURSE             Makefile  presentation  README.git-flow.md template
imgorbunov@imgorbunov:~/work/study/2025-2026/Операционные системы/2026-1--study--os-intro$
```

Рисунок 2.9: Загрузка шаблона

Подготовка репозитория и коммит изменений

```
create mode 100644 project-personal/stage06/report/_extensions/yamadharma/minited-quarto/_extension.yml
l
create mode 100644 project-personal/stage06/report/_extensions/yamadharma/minited-quarto/minited-quarto.lua
create mode 100644 project-personal/stage06/report/_quarto.yml
create mode 100644 project-personal/stage06/report/_resources/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 project-personal/stage06/report/_resources/tex/preamble.tex
create mode 100644 project-personal/stage06/report/bib/cite.bib
create mode 100644 project-personal/stage06/report/image/solvay.jpg
create mode 100644 project-personal/stage06/report/os-intro-project-personal-stage06-report.qmd
imgorbunov@imgorbunov:~/work/study/2025-2026/Операционные системы/2026-1--study--os-intro$ git push
Enumerating objects: 109, done.
Counting objects: 100% (109/109), done.
Delta compression using up to 8 threads
Compressing objects: 100% (90/90), done.
Writing objects: 100% (106/106), 705.57 KiB | 4.70 MiB/s, done.
Total 106 (delta 41), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (41/41), completed with 1 local object.
To github.com:im-gorgunov/2026-1--study--os-intro.git
  1cb5102..836d5d5  master -> master
imgorbunov@imgorbunov:~/work/study/2025-2026/Операционные системы/2026-1--study--os-intro$
```

Рисунок 2.10: Первый коммит

3 Вывод

Мы приобрели практические навыки работы с сервисом github.

4 Контрольные вопросы

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

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

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

- хранилище - пространство на накопителе где расположен репозиторий
- commit - сохранение состояния хранилища
- история - список изменений хранилища (коммитов)
- рабочая копия - локальная копия сетевого репозитория, в которой работает программист. Текущее состояние файлов проекта, основанное на версии, загруженной из хранилища (обычно на последней)

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

Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере. Доступ к нему осуществлялся через специальное клиентское приложение. В качестве примеров таких программных продуктов можно привести CVS, Subversion.

Распределенные системы контроля версий (Distributed Version Control System, DVCS) позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой. 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. При этом сервер, чаще всего, выбирается условно, т.к. в большинстве DVCS нет такого понятия как «выделенный сервер с центральным репозиторием».

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

Один пользователь работает над проектом и по мере необходимости делает коммиты, сохраняя определенные этапы.

5. Опишите порядок работы с общим хранилищем VCS.

Несколько пользователей работают каждый над своей частью проекта. При этом каждый должен работать в своей ветки. При завершении работы ветка пользователя слиивается с основной веткой проекта.

6. Каковы основные задачи, решаемые инструментальным средством git?

- Ведение истории версий проекта: журнал (log), метки (tags), ветвления (branches).
- Работа с изменениями: выявление (diff), слияние (patch, merge).
- Обеспечение совместной работы: получение версии с сервера, загрузка обновлений на сервер.

7. Назовите и дайте краткую характеристику командам git.

- `git config` - установка параметров
- `git status` - полный список изменений файлов, ожидающих коммита
- `git add .` - сделать все измененные файлы готовыми для коммита.
- `git commit -m «[descriptive message]»` - записать изменения с заданным сообщением.
- `git branch` - список всех локальных веток в текущей директории.
- `git checkout [branch-name]` - переключиться на указанную ветку и обновить рабочую директорию.
- `git merge [branch]` – соединить изменения в текущей ветке с изменениями из заданной.
- `git push` - запушить текущую ветку в удаленную ветку.
- `git pull` - загрузить историю и изменения удаленной ветки и произвести слияние с текущей веткой.

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

- `git remote add [имя] [url]` – добавляет удалённый репозиторий с заданным именем;
- `git remote remove [имя]` – удаляет удалённый репозиторий с заданным именем;
- `git remote rename [старое имя] [новое имя]` – переименовывает удалённый репозиторий;

- `git remote set-url [имя] [url]` – присваивает репозиторию с именем новый адрес;
- `git remote show [имя]` – показывает информацию о репозитории.

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

Ветвление – это возможность работать над разными версиями проекта: вместо одного списка с упорядоченными коммитами история будет расходиться в определённых точках. Каждая ветвь содержит легковесный указатель HEAD на последний коммит, что позволяет без лишних затрат создать много веток. Ветка по умолчанию называется `master`, но лучше назвать её в соответствии с разрабатываемой в ней функциональностью.

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

Зачастую нам не нужно, чтобы Git отслеживал все файлы в репозитории, потому что в их число могут входить: