

# Лабараторная работа 1

Простейший шаблон

---

Хайманов А.С.

30 июля 1999

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Хайманов А.С
- студент
- Российский университет дружбы народов
- 1132243810@pfur.ru

## Вводная часть

---

- Управление версиями
- Совместная работа
- Поддержка распределенной разработки ## Цели и задачи
- Изучить применение средств контроля версии
- освоить умения по работе с гит

- Указанные в соответствии с лабораторной работой



# Установка и настройка git

Для выполнения запрошенной операции требуются привилегии суперпользователя. Пожалуйста, войдите в систему как пользователь `askhayjmanov` или `--downloadonly`, чтобы выполнить команду без изменения состояния системы.

```
[askhayjmanov@askhayjmanov ~]$ sudo -i
[sudo] пароль для askhayjmanov:
[root@askhayjmanov ~]# dnf install git
```

Обновление и загрузка репозитория:

```
Fedora 41 - x86_64 - Updates
Fedora 41 - x86_64 - Updates
```

Репозитории загружены.

Пакет `git-2.48.1-1.fc41.x86_64` уже установлен.

Нечего делать.

```
[root@askhayjmanov ~]# dnf install gh
```

Обновление и загрузка репозитория:

Репозитории загружены.

Пакет	Арх.	Версия
Установка:	x86_64	2.65.0-1.fc41

Сводка транзакции:

Установка: 1 пакета

Общий размер входящих пакетов составляет 10 MiB. Необходимо загрузить 10 MiB.

После этой операции будут использоваться дополнительные 43 MiB (установка 43 MiB, удаление 0 B).

Is this ok [y/N]: y

```
[1/1] gh-2.65.0-1.fc41.x86_64
-----
[1/1] Total
```

Выполнение транзакции

```
[1/3] Проверить файлы пакета
[2/3] Подготовить транзакцию
[3/3] Установка gh-2.65.0-1.fc41.x86_64
```

Завершено!

```
[root@askhayjmanov ~]# git config --global user.name "Aslanbek Haymanov"
[root@askhayjmanov ~]# git config --global user.email "vladikavkas1999@gmail.com"
[root@askhayjmanov ~]# git config --global core.quotePath false
[root@askhayjmanov ~]# git config --global init.defaultBranch master
[root@askhayjmanov ~]# git config --global core.autocrlf input
[root@askhayjmanov ~]# git config --global core.safecrlf warn
[root@askhayjmanov ~]# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
```

### Интеграция мыши ...

Репозиторий	Размер
updates	42.6 MiB

100%	28.1 KiB/s	24.8 KiB	00m01s
100%	1.8 MiB/s	2.7 MiB	00m01s

100%	10.2 MiB/s	10.3 MiB	00m01s
100%	7.1 MiB/s	10.3 MiB	00m01s
100%	45.8 B/s	1.0 B	00m00s
100%	3.6 B/s	1.0 B	00m00s
100%	65.2 MiB/s	42.7 MiB	00m01s



# Создание ключей

Все верно? (y/N) y

GnuPG должен составить идентификатор пользователя для идентификации ключа.

Ваше полное имя: Aslanbek

Адрес электронной почты: vladikavkas1999@gmail.com

Примечание:

Вы выбрали следующий идентификатор пользователя:

"Aslanbek <vladikavkas1999@gmail.com>"

Оменить (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, движения мыши, обращения 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, движения мыши, обращения 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/opensp-gevocs.d'

gpg: сертификат отзыва записан в '/root/.gnupg/opensp-gevocs.d/D8909FC55206F4A743E8C4F222123A3F0760982C.rev'.  
открытый и секретный ключи созданы и подписаны.

pub rsa4096 2025-02-28 [SC]

D8909FC55206F4A743E8C4F222123A3F0760982C

uid Aslanbek <vladikavkas1999@gmail.com>

sub rsa4096 2025-02-28 [E]

[root@askhayjmanov ~]# 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, 0m, 0f, 1u

[keyboxd]

-----

sec rsa4096/22123A3F0760982C 2025-02-28 [SC]

D8909FC55206F4A743E8C4F222123A3F0760982C

uid [ абсолютно ] Aslanbek <vladikavkas1999@gmail.com>

ssb rsa4096/8B50829C160A1D47 2025-02-28 [E]

[root@askhayjmanov ~]# gpg --armor --export <PGP Fingerprint> | xclip -sel clip

-bash: синтаксическая ошибка рядом с неожиданным маркером «|»

[root@askhayjmanov ~]# gpg --armor --export <PGP Fingerprint> | xclip -sel cl

Интеграция мыши ...

# Настройка каталога курса

The screenshot shows a web browser window with the GitHub repository page for 'yamadharma'. The repository is titled 'os-intro\_\_38.0 x' and 'yamadharma/c x'. The file list includes 'config', 'template', '.gitattributes', '.gitignore', '.gitmodules', 'CHANGELOG.md', 'COURSE', 'LICENSE', 'Makefile', and 'README.en.md'. A 'Clone' button is visible, and a message states: 'You don't have any public SSH keys in your GitHub account. You can [add a new public key](#), or try cloning this repository via HTTPS.' Below this, there is a field for the repository URL: 'git@github.com:yamadharma' and a 'Clone' button. A 'Download ZIP' button is also present.

The terminal window on the right shows the following commands and output:

```
-4, --ipv4          использовать только IPv4 адреса
-6, --ipv6          использовать только IPv6 адреса
--[no-]filter <аргументы>
                    фильтрация объектов
--[no-]also-filter-submodules
                    apply partial clone filters to submodules
--[no-]remote-submodules
                    any cloned submodules will use their remote-tracking branch
--[no-]sparse       initialize sparse-checkout file to include only files at root
--[no-]bundle-uri <uri>
                    a URI for downloading bundles before fetching from origin
remote

[askhayjmanov@askhayjmanov Операционные системы]$ git clone --recursive git@github.com:yamadharma/course-directory-student-template.git/study_2022-2023_os-intro.git os-intro
Клонирование в «os-intro»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+D1Y3wvV6TujJhbpZ1sF/zLDABzPMSvHkKz4UvCoqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
git@github.com: Permission denied (publickey).
fatal: Не удалось прочитать из внешнего репозитория.

Удостоверьтесь, что у вас есть необходимые права доступа
и репозиторий существует.
[askhayjmanov@askhayjmanov Операционные системы]$ sudo -i
[sudo] пароль для askhayjmanov:
[root@askhayjmanov ~]# cd ~/work/study/2022-2023/"Операционные системы"
-bash: cd: /root/work/study/2022-2023/Операционные системы: Нет такого файла или каталога
[root@askhayjmanov ~]# mkdir -p ~/work/study/2022-2023/"Операционные системы"
[root@askhayjmanov ~]# cd ~/work/study/2022-2023/"Операционные системы"
[root@askhayjmanov Операционные системы]# git clone --recursive git@github.com:yamadharma/course-directory-student-template.git/study_2022-2023_os-intro.git os-intro
Клонирование в «os-»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+D1Y3wvV6TujJhbpZ1sF/zLDABzPMSvHkKz4UvCoqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

- Системы контроля версий (VCS) — это инструменты для отслеживания изменений в файлах и управления проектами. Они позволяют сохранять различные версии файлов, что упрощает возврат к предыдущим состояниям и совместную работу.

- Хранилище (Repository): Место, где хранятся файлы проекта и их история изменений.
- Commit: Операция, сохраняющая изменения в хранилище с описанием внесённых изменений.
- История (History): Последовательность всех коммитов, показывающая, как проект развивался.
- Рабочая копия (Working Copy): Локальная версия файлов, с которой пользователь работает.

- Централизованные VCS: Один центральный сервер, к которому обращаются все пользователи. Примеры: Subversion (SVN), CVS.
- Децентрализованные VCS: Каждый пользователь имеет полную копию репозитория. Примеры: Git, Mercurial.

- Создание локального репозитория.
- Внесение изменений в рабочую копию.
- Сохранение изменений с помощью commit.
- Просмотр истории изменений.
- Возврат к предыдущим версиям.

- Клонирование удалённого репозитория.
- Создание новой ветки.
- Внесение изменений и commit.
- Синхронизация изменений (push).
- Получение изменений от других (pull).

- Отслеживание изменений.
- Управление версиями.
- Слияние изменений.
- Работа с ветками.
- Возврат к предыдущим версиям.



- `git init`: Создает новый репозиторий.
- `git clone`: Клонировать удалённый репозиторий.
- `git add`: Добавляет изменения в индекс.
- `git commit`: Сохраняет изменения.
- `git push`: Отправляет изменения в удалённый репозиторий.
- `git pull`: Получает изменения из удалённого репозитория.
- `git branch`: Управляет ветками.

- Локальный репозиторий: `git init` `git add .` `git commit -m "Initial commit"`
- Удалённый репозиторий: `git clone https://github.com/user/repo.git` `git push origin main`

Ветви позволяют создавать параллельные линии разработки, что полезно для работы над новыми функциями или исправлениями, не затрагивая основную ветку.

---



Файл `.gitignore` используется для указания файлов и папок, которые Git должен игнорировать, чтобы избежать случайного коммита ненужных файлов. Это помогает поддерживать чистоту репозитория.

В процессе выполнения лабораторной работы 2 освоил умения по работе с гит

- Запоминается последняя фраза. © Штирлиц
- Главное сообщение, которое вы хотите донести до слушателей
- Избегайте использовать последний слайд вида *Спасибо за внимание*