

Отчёт по лабораторной работе №2

Операционные системы

Карпова Анастасия Александровна

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Установка ПО	6
3.2	Базовая настройка git	6
3.3	Создание ключей ssh	6
3.4	Создание ключей pgr	7
3.5	Добавление PGP ключа в GitHub	7
3.6	Настройка автоматических подписей коммитов git	7
3.7	Настройка gh	8
3.8	Создание репозитория курса на основе шаблона	8
3.9	Настройка каталога курса	8
3.10	Ответы на контрольные вопросы	9
4	Выводы	12
	Список литературы	13

Список иллюстраций

1 Цель работы

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

2 Задание

1. Установка ПО
2. Базовая настройка git
3. Создание ключей ssh
4. Создание ключей pgr
5. Добавление ключа PGP на гитхаб
6. Настройка автоматических подписей коммитов git
7. Настройка gh
8. Шаблон для рабочего пространства
9. Ответы на контрольные вопросы

3 Выполнение лабораторной работы

3.1 Установка ПО

Устанавливаю гит с помощью `dnf install git`

Устанавливаю gh с помощью `dnf install gh`

3.2 Базовая настройка git

Задаю имя и email владельца репозитория с помощью `git config --global user.name "Name Surname"` и `git config --global user.email "work@mail"`

Настраиваю utf-8 в выводе сообщений git с помощью `git config --global core.quotePath false`

Задаю имя начальной ветки с помощью `git config --global init.defaultBranch master`

Параметр `autocrlf` с помощью `git config --global core.autocrlf input` Параметр `safecrlf` с помощью `git config --global core.safecrlf warn`

3.3 Создание ключей ssh

Создаю ssh ключ по алгоритму `rsa` с ключём размером 4096 бит:

```
ssh-keygen -t rsa -b 4096
```

по алгоритму `ed25519`:

```
ssh-keygen -t ed25519
```

3.4 Создание ключей gpg

Генерирую ключ

```
gpg --full-generate-key
```

Из предложенных опций выбираю:

тип RSA and RSA;

размер 4096;

выберите срок действия; значение по умолчанию – 0 (срок действия не истекает никогда)

GPG запросит личную информацию, которая сохранится в ключе:

Имя (не менее 5 символов).

Адрес электронной почты.

При вводе email убеждаюсь, что он соответствует адресу, используемому на GitHub.

Комментарий. Можно ввести что угодно или нажать клавишу ввода, чтобы оставить это пол

3.5 Добавление PGP ключа в GitHub

Вывожу список ключей и копируем отпечаток приватного ключа:

```
gpg --list-secret-keys --keyid-format LONG
```

Отпечаток ключа — это последовательность байтов, используемая для идентификации более длинного, по сравнению с самим отпечатком ключа.

Копирую мой сгенерированный PGP ключ в буфер обмена:

```
gpg --armor --export | xclip -sel clip
```

Перехожу в настройки GitHub (<https://github.com/settings/keys>), нажимаю на кнопку New GPG key и вставляю полученный ключ в поле ввода.

3.6 Настройка автоматических подписей коммитов git

Используя введенный email, указываю Git применять его при подписи коммитов:

```
git config --global user.signingkey git config --global commit.gpgsign true git config  
--global gpg.program $(which gpg2)
```

3.7 Настройка gh

Для начала необходимо авторизоваться

```
gh auth login
```

Утилита задаст несколько наводящих вопросов. Авторизоваться можно через браузер.

3.8 Создание репозитория курса на основе шаблона

Необходимо создать шаблон рабочего пространства.

Например, для 2022–2023 учебного года и предмета «Операционные системы» (код предмета os-intro) создание репозитория примет следующий вид:

```
mkdir -p ~/work/study/2022-2023/“Операционные системы” cd ~/work/study/2022-  
2023/“Операционные системы” gh repo create study_2022-2023_os-intro  
--template=yamadharma/course-directory-student-template --public git clone  
--recursive git@github.com:/study_2022-2023_os-intro.git os-intro
```

3.9 Настройка каталога курса

Перехожу в каталог курса:

```
cd ~/work/study/2022-2023/“Операционные системы”/os-intro
```

Удаляю лишние файлы:

```
rm package.json
```

Создаю необходимые каталоги:

```
echo os-intro > COURSE make prepare
```

Отправляю файлы на сервер:


```
git add . git commit -am 'feat(main): make course structure' git push
```

3.10 Ответы на контрольные вопросы

Что такое системы контроля версий (VCS) и для решения каких задач они предназначены? - Система контроля версий (Version Control System, VCS) представляет собой программное обеспечение, которое позволяет отслеживать изменения в документах, при необходимости производить их откат, определять, кто и когда внес исправления и т. п.

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.

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.

Сначала создается и подключается удаленный репозиторий, затем по мере изменения проекта эти изменения отправляются на сервер.

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.

Хранение информации о всех изменениях в вашем коде, обеспечение удобства командной работы над кодом.

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`

`git push -all` отправляем из локального репозитория все сохраненные изменения в центральный репозиторий, предварительно создав локальный репозиторий и сделав предварительную конфигурацию.

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.

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

4 Выводы

В ходе этой работы я изучила идеологию и применение средств контроля версий. Освоила умения по работе с git.

Список литературы

Лабораторная работа № 2 [Электронный ресурс] URL: <https://esystem.rudn.ru/mod/page/view>.