

Отчёт по лабораторной работе №2

Операционные системы

Алина Андреевна Кайнова

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	8
4.1	Создание базовой конфигурации для работы с git	8
4.2	Создание ключей SSH	8
4.3	Создание ключа PGP	9
4.4	Настройка подписи git	10
4.5	Регистрация на Github	10
4.6	Создание локального каталога для выполнения заданий по предмету	11
5	Выводы	12
	Список литературы	13

Список иллюстраций

4.1	Базовая настройка git	8
4.2	Ключ SSH по алгоритму rsa	8
4.3	Ключ SSH по алгоритму ed25519	9
4.4	Ключ PGP	9
4.5	Настройка автоматических подписей коммитов git	10
4.6	Аккаунт git	10
4.7	Создание репозитория курса на основе шаблона	11
4.8	Настройка каталога курса	11
4.9	Отправка файлов	11

1 Цель работы

Изучить идеологию и применение средств контроля версий, приобрести навыки работы с git

2 Задание

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

3 Теоретическое введение

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. 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 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 — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных. Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет

другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом. Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить. В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным. Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд. Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями. Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

4 Выполнение лабораторной работы

4.1 Создание базовой конфигурации для работы с git

Установим git, зададим имя и почту владельца, настроим utf-8, зададим имя начальной ветки, autocrlf, safecrlf

```
[aakayjnova@fedora ~]$ git config --global user.name "Alina Kainova"
warning: user.name имеет несколько значений
error: нельзя перезаписать несколько значений одним
        Используйте регулярные выражения, параметры --add или --replace-all, чтобы изменить user.name
.
[aakayjnova@fedora ~]$ git config --global user.email "1132236004@pfur.ru"
[aakayjnova@fedora ~]$ git config --global core.quotepath false
[aakayjnova@fedora ~]$ git config --global init.defaultBranch master
[aakayjnova@fedora ~]$ git config --global core.autocrlf input
[aakayjnova@fedora ~]$ git config --global core.autocrlf input
[aakayjnova@fedora ~]$ git config --global core.safecrlf warn
```

Рис. 4.1: Базовая настройка git

4.2 Создание ключей SSH

Создадим ключ по алгоритму rsa размером 4096 бит

```
[aakayjnova@fedora ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aakayjnova/.ssh/id_rsa):
/home/aakayjnova/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aakayjnova/.ssh/id_rsa
Your public key has been saved in /home/aakayjnova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:qDsDLR4VNqq/dvpJFwndZPwXKxxLpuAlyTnCsZUFuLY aakayjnova@fedora
The key's randomart image is:
```

Рис. 4.2: Ключ SSH по алгоритму rsa

Создадим ключ по алгоритму ed25519

```
[aakayjnova@fedora ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/aakayjnova/.ssh/id_ed25519):
/home/aakayjnova/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aakayjnova/.ssh/id_ed25519
Your public key has been saved in /home/aakayjnova/.ssh/id_ed25519.pub
The key fingerprint is:
```

Рис. 4.3: Ключ SSH по алгоритму ed25519

4.3 Создание ключа PGP

Генерируем ключ и выбираем заданные опции

```
[aakayjnova@fedora ~]$ gpg --full-generate-key
gpg (GnuPG) 2.4.0; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Выберите тип ключа:
(1) RSA and RSA
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
(9) ECC (sign and encrypt) *default*
(10) ECC (только для подписи)
(14) Existing key from card
Ваш выбор? 1
длина ключей RSA может быть от 1024 до 4096.
Какой размер ключа Вам необходим? (3072) 4096
Запрошенный размер ключа - 4096 бит
Выберите срок действия ключа.
  0 = не ограничен
  <n> = срок действия ключа - n дней
  <n>w = срок действия ключа - n недель
  <n>m = срок действия ключа - n месяцев
  <n>y = срок действия ключа - n лет
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: Alina
Адрес электронной почты: 1132236004@pfur.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
"Alina <1132236004@pfur.ru>"
```

Рис. 4.4: Ключ PGP

4.4 Настройка подписи git

Указываем Git применять введённый email при подписи коммитов

```
[aakayjnova@fedora ~]$ git config --global user.signingkey F5C20A2354A20F0D  
[aakayjnova@fedora ~]$ git config --global commit.gpgsign true  
[aakayjnova@fedora ~]$ git config --global gpg.program $(which gpg2)
```

Рис. 4.5: Настройка автоматических подписей коммитов git

4.5 Регистрация на Github

У меня уже есть аккаунт на Github

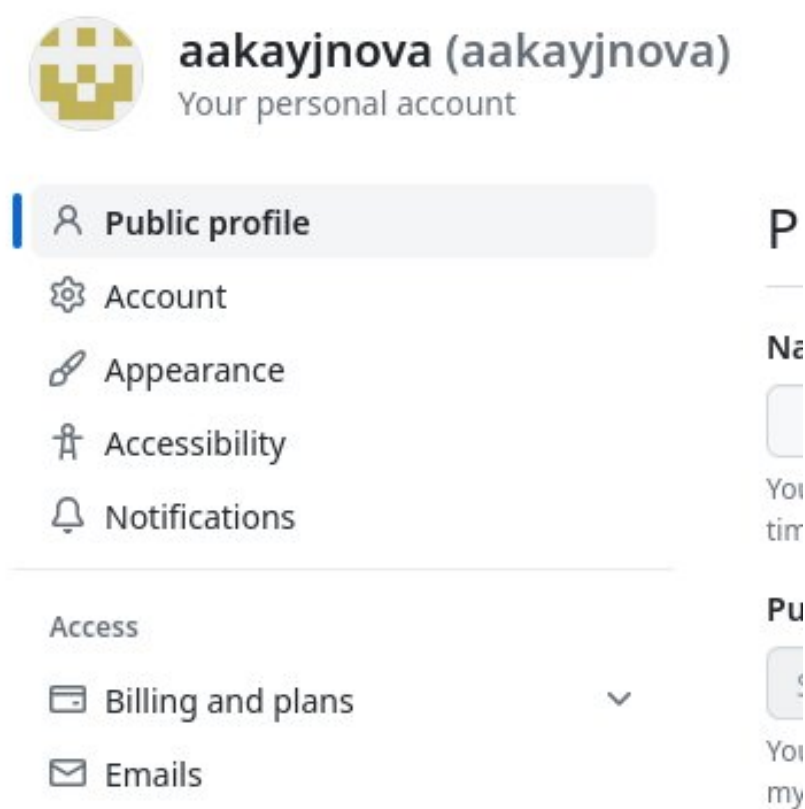


Рис. 4.6: Аккаунт git

4.6 Создание локального каталога для выполнения заданий по предмету

Создадим шаблон рабочего пространства

```
[aakayjnova@fedora ~]$ mkdir -p ~/work/study/2023-2024/"Операционные системы"
[aakayjnova@fedora ~]$ cd ~/work/study/2023-2024/"Операционные системы"
[aakayjnova@fedora Операционные системы]$ gh repo create study_2023-2024_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository aakayjnova/study_2023-2024_os-intro on GitHub
[aakayjnova@fedora Операционные системы]$ git clone --recursive git@github.com:aakayjnova/study_2023-2024_os-intro.git os-intro
Клонирование в «os-intro»...
```

Рис. 4.7: Создание репозитория курса на основе шаблона

Настроим каталог курса

```
[aakayjnova@fedora Операционные системы]$ cd ~/work/study/2023-2024/"Операционные системы"/os-intro
[aakayjnova@fedora os-intro]$ rm package.json
[aakayjnova@fedora os-intro]$ echo os-intro > COURSE
[aakayjnova@fedora os-intro]$ make
```

Рис. 4.8: Настройка каталога курса

Отправим файлы на сервер

```
[aakayjnova@fedora os-intro]$ git add .
[aakayjnova@fedora os-intro]$ git commit -am 'feat(main): make course structure'
[master 52eb8e3] feat(main): make course structure
2 files changed, 1 insertion(+), 14 deletions(-)
delete mode 100644 package.json
[aakayjnova@fedora os-intro]$ git push
```

Рис. 4.9: Отправка файлов

5 Выводы

В ходе данной лабораторной работы мы изучили идеологию и применение средств контроля версий, также приобрели навыки по работе с git.

Список литературы

1. <https://esystem.rudn.ru/mod/page/view.php?id=1098790#orgf425532>

∴ ∴