

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

дисциплина: Операционные системы

Студент: Давитян Эдуард

Группа: НПИ-01-21

Цели работы

Изучить идеологию и применение средств контроля версий, а также освоить умения по работе с git.

Выполнение работы :

1. Установка git-flow

```
[eduard.davitiani@localhost ~]$ cd /tmp
[eduard.davitiani@localhost tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
wget: нераспознанный параметр «--no-check-certificate»
Использование: wget [ПАРАМЕТР]... [URL]...

Дополнительные параметры выводятся по команде «wget --help».
[eduard.davitiani@localhost tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[eduard.davitiani@localhost tmp]$ chmod +x gitflow-installer.sh
chmod: невозможно получить доступ к 'gitflow-installer.sh': Нет такого файла или каталога
[eduard.davitiani@localhost tmp]$ sudo ./gitflow-installer.sh install stable
[sudo] пароль для eduard.davitiani:
sudo: ./gitflow-installer.sh: command not found
[eduard.davitiani@localhost tmp]$ sudo dnf install gh
Ошибка: Не удается найти правильный baseurl для репозитория: appstream
[eduard.davitiani@localhost tmp]$ gpg --full-generate-key
gpg (GnuPG) 2.2.20; Copyright (C) 2020 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.
```

2. Установка gh

```
[eduard.davitiani@localhost tmp]$ sudo dnf install gh
[sudo] пароль для eduard.davitiani:
```

3. Базовая настройка git

```
git config --global user.name "Name Surname"
```

```
git config --global user.email "work@mail"
```

```
git config --global core.quotepath false
```

```
git config --global init.defaultBranch master
```

```
git config --global core.autocrlf input
```

```
git config --global core.safecrlf warn
```

С помощью данных команд зададим имя и email, настроим utf8, зададим имя начальной ветки, а также параметры autocrlf и safecrlf.

4. Создаем ключи pgr. Создаем ключи ssh с помощью команд `ssh-keygen -t rsa -b 4096` и `ssh-keygen -t ed25519`

```
[eduard.davitiani@localhost tmp]$ gpg --full-generate-key
gpg (GnuPG) 2.2.20; Copyright (C) 2020 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 и RSA (по умолчанию)
  (2) DSA и Elgamal
  (3) DSA (только для подписи)
  (4) RSA (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? (2048) 2048
Запрошенный размер ключа - 2048 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) 10
Ключ действителен до Вс 19 июн 2022 18:52:30 EDT
Резервное копирование? (y/N) y
```

Добавление PGP ключа в GitHub

```
[eduard.davitiani@localhost ~]$ 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
gpg: срок следующей проверки таблицы доверия 2022-05-08
/home/eduard.davitiani/.gnupg/pubring.kbx
-----
sec   rsa2048/3CCB39348BD88A39 2022-04-28 [SC] [   годен до: 2022-05-08]
      FC7CDBB476CD3A0EFD8E21253CCB39348BD88A39
uid           [   абсолютно ] eduard.davitiani (edo1223) <eduard.davitiani@gmail.com>
ssb   rsa2048/61F07666D49EBD3E 2022-04-28 [E] [   годен до: 2022-05-08]

[eduard.davitiani@localhost ~]$
```

```
eduard.davitiani@localhost:~
Файл  Правка  Вид  Поиск  Терминал  Справка
[eduard.davitiani@localhost ~]$ ssh-keygen -t rsa -b 4096 -C "eduard.davitiani@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/eduard.davitiani/.ssh/id_rsa):
Created directory '/home/eduard.davitiani/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/eduard.davitiani/.ssh/id_rsa.
Your public key has been saved in /home/eduard.davitiani/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:NOHzSCwPnMQulhtIKfFDIDqjpVkpPluTx0CIH7kICyc eduard.davitiani@gmail.com
The key's randomart image is:
+---[RSA 4096]----+
|+ .. .
|o .o.+ .
|.= + += B
|E & o +* =
|+& 0 o .S .
|+.B = +
|o+ o + .
|o.. .
|
+----[SHA256]-----+
[eduard.davitiani@localhost ~]$
```

Выводы:

В ходе выполнения данной лабораторной работы изучили идеологию и применение средств контроля версий, а также освоили умения по работе с git.

Контрольные вопросы

1. VCS – это программное обеспечение для облегчения работы с изменяющейся информацией, применяется при работе группы людей над одним проектом.
2. Хранилище – это место хранения всех версий и служебной информации

Commit – это процесс создания новой версии

История – это информация по изменениям, внесенным в проект

Рабочая копия – это текущее состояние файлов, основанное на версии, загруженной из хранилища.

3. Централизованные были созданы для бэкапирования, отслеживания и синхронизации файлов, а также все изменения в ней проходят через центральный сервер. Примеры: CVS, Subversion

Децентрализованные были созданы для обмена изменениями, и у каждого есть свой полноценный репозиторий. И у децентрализованных нет такой жестко заданной структуры репозитория с центральным сервером. Примеры: Git, Bazaar, Mercurial.

6. Есть 2 основные задачи: хранение информации о всех изменениях в коде и обеспечение удобства командной работы

7.

- создание основного дерева репозитория:

```
1 git init
```

- получение обновлений (изменений) текущего дерева из центрального репозитория:

```
1 git pull
```

- отправка всех произведённых изменений локального дерева в центральный репозиторий:

```
1 git push
```

- просмотр списка изменённых файлов в текущей директории:

```
1 git status
```

- просмотр текущих изменений:

```
1 git diff
```

- сохранение текущих изменений:

- добавить все изменённые и/или созданные файлы и/или каталоги:

```
1 git add .
```

- добавить конкретные изменённые и/или созданные файлы и/или каталоги:

```
1 git add имена_файлов
```

- удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории):

```
1 git rm имена_файлов
```

- сохранение добавленных изменений:

- сохранить все добавленные изменения и все изменённые файлы:

```
1 git commit -am 'Описание коммита'
```

8. Если вы захотите продолжить работу над проектом на другом устройстве то вам понадобится удаленный репозиторий.

9. При работе нескольких человек над одним проектом, ветви будут нужны чтобы они могли спокойно работать, не мешая при этом друг другу.

10. Игнорируемые файлы – это обычно артефакты сборки или файлы, генерируемые машиной из исходных файлов в репозитории или же файлы которые не должны попасть в коммиты.