

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:

Воропаева Алёна Дмитриевна

Студ. Билет: 1032216529

Группа: НПИбд-01-21

МОСКВА

2022 г.

Цель работы:

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

Ход работы:

На Githabe предварительно была создана учетная запись и заполнены все данные.

1. Создать базовую конфигурацию для работы с git.

Запускаем виртуальную машину и устанавливаем на нее git-flow, gh.

```
адвог
[advoropaeval@fedora ~]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[advoropaeval@fedora ~]$ chmod +x gitflow-installer.sh
[advoropaeval@fedora ~]$ sudo ./gitflow-installer.sh install stable

Мы полагаем, что ваш системный администратор изложил вам основы
безопасности. Как правило, всё сводится к трём следующим правилам:

    №1) Уважайте частную жизнь других.
    №2) Думайте, прежде что-то вводить.
    №3) С большой властью приходит большая ответственность.

[sudo] пароль для advoropaeval:
### git-flow no-make installer ###
```

Рис. 1.1. Установка git-flow в Fedora Linux

```
[advoropaeval@fedora ~]$ sudo dnf install gh
Fedora 35 - x86_64 - Updates          17 kB/s | 20 kB      00:01
Fedora 35 - x86_64 - Updates          1.1 MB/s | 2.6 MB    00:02
Fedora Modular 35 - x86_64 - Updates  37 kB/s | 19 kB     00:00
Последняя проверка окончания срока действия метаданных: 0:00:01 назад, Сб 23 апр 2022 17:57:23.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
gh          x86_64       2.7.0-1.fc35 updates      6.8 М
```

Рис. 1.2. Установка gh в Fedora Linux

Производим базовую настройку в git.

- Зададим имя и email владельца репозитория
- Настроим utf-8 в выводе сообщений git
- Настройте верификацию и подписание коммитов git. – Зададим имя начальной ветки (будем называть её master)
- Параметр autocrlf
- Параметр safecrlf

```
[advoropaeval@fedora ~]$ git config --global user.name "Voropaeva Alyona"
[advoropaeval@fedora ~]$ git config --global user.email "26alyona@gmail.com"
[advoropaeval@fedora ~]$ git config --global core.quotepath false
[advoropaeval@fedora ~]$ git config --global init.defaultBranch master
[advoropaeval@fedora ~]$ git config --global core.autocrlf input
[advoropaeval@fedora ~]$ git config --global core.safecrlf warn
```

Рис. 1.3. Базовая настройка git

2. Создать ключ SSH.

Создаем ключ SSH по алгоритму rsa с ключём размером 4096 бит

```
[advoropaeval@fedora ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/advoropaeval/.ssh/id_rsa): rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in rsa
Your public key has been saved in rsa.pub
The key fingerprint is:
SHA256:wQYK02QjG1IeUvY6JdqyM1lOZYP0sogLUfo1Yr/74s4 advoropaeval@fedora
The key's randomart image is:
+---[RSA 4096]-----+
| =o%      .          |
| .& 0 . o          |
| + X @    +          |
| .B & o . .          |
| * 0 .   S          |
| .0 . .             |
| * . .              |
| o ...              |
|   oEo.             |
+---[SHA256]-----+
```

Рис. 2.1. Создание ключа SSH

3. Создать ключ PGP

Создаем ключ pgr.

– Генерируем ключ

– Из предложенных опций выбираем: – тип RSA and RSA;

– размер 4096;

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

– Вводим личную информацию, которая сохранится в ключе:

- Имя
- Адрес электронной почты, соответствующий адресу, используемому на GitHub.
- Выводим список ключей и копируем отпечаток приватного ключа
- Переходим в настройки GitHub, нажимаем на кнопку New GPG key и вставляем полученный ключ в поле ввода.

Каждый шаг подробно проиллюстрирован скриншотами.

```
[advoropaeva1@fedora ~]$ gpg --full-generate-key
gpg (GnuPG) 2.3.2; 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.

gpg: создан каталог '/home/advoropaeva1/.gnupg'
gpg: создан щит с ключами '/home/advoropaeva1/.gnupg/pubring.kbx'

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: Флёна
Адрес электронной почты: 26alyona@gmail.com
Примечание:
Используется таблица символов 'utf-8'.
Вы выбрали следующий идентификатор пользователя:
  "Флёна <26alyona@gmail.com>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? n
Ваше полное имя: Alyona Voropaeva
Вы выбрали следующий идентификатор пользователя:
  "Alyona Voropaeva <26alyona@gmail.com>"
```

Рис. 3.1. Генерация ключа

```
[advoropaeva1@fedora ~]$ 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
/home/advoropaeva1/.gnupg/pubring.kbx
-----
sec   rsa4096/26B1038C0DBFEA09 2022-04-23 [SC]
      1224B9E0F1C6872AF590124326B1038C0DBFEA09
uid   [ абсолютно ] Alyona Voropaeva <26alyona@gmail.com>
ssb   rsa4096/37191BF4DB9CDAC7 2022-04-23 [E]
```

Рис. 3.3. вывод списка ключей

```
[advoropaeva1@fedora ~]$ gpg --armor --export 26B1038C0DBFEA09 | xclip -sel clip
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGJkFtwBEADebkqLFzWR5tGPPqVus8Nt28qeFXpTJbIfRuKN9ZjeV7qEcbL9
oXBjeQbmmJwpI7JnDVyH2nRFa+OjpRtouQgjfaRXL6X2ZFLidK2/QaGNeTYWx2V+
FcL1Qy4tuoirrC63KxoeIoIfzhVWDftgTd1KoxBc72PlEQ8UFpGRf79hUqVSYHbp
RwK2eZachrabzvstC7Xxcyt91nLxTtipsYtg542EQJKilMEKfJzpDPAAnYibr1q4
gX+SeLD08qKcBMLJM/5M99o6XB4tAB0ky0ZsrX5LlqEDx49N6qb0R56MWjUWDaIF
+UpFs4EZjTOXHzkk0PGG2JIX8f1fGnQjiEVhArovHaEkax821gkrIk5sAcFGJzRV
PSQZA000cfDajspRosQgjXPn5UA3nSAHHZoeHxSk7dTqR5khal5Jyr7jKUoHwIU
zT8Z0LQJKd7sBB0FYLFaVcT0GAVKxL3swX4k84L6MEIacFGJdKCjhrchfqjvMSdy
CLn92AEWQWXZSFR1E7DsFseES7fEr02NuCkak0WNFLl1fKoppf+o9gwKJWJ+oQi
Zv+soeyp4ml6+uzqEGZ+kkezC+4AvNcWhiNbn9rQardsxxyP9lkTrj6GytTLymDp
H7wz0o2PdggcOUi16xQsrJma9+IPpYR5dAX30mD0X9AasoKW9sELnt9g2QARAQAB
tCVBbHlVbmEgVm9yb3BhZXZhIDwyNmFseW9uYUBnbWVpbC5jb20+iQJSBBMBCAA8
FiEEiS54PHGhyr1kBJDJrEDJA2/6gkFAMJkFtwCGwMFCwkIBwIDIGIBBhUKCQGL
AgQWAGMAh4HAheAAAJECaxA4wNv+oJICQQAALDZcsnPlC4dooX7s2gVheRPN5ws
tlvuiSaFvTx/cQ1X6SkS70BYUEyxrSck0ueIdkkisbVLsg6i/ImpK61VPA7AgCE/
07NYV1eWe2VEF6Wt1RDGzUqq87RCo+wiuTKs3e/Q4zD2ARXmiGDj0hnp5gI2Dz17
SybIuLpJv5J39VqQSYzaqbkQTR0BcIPJHHgN4SgB78o036tuyd7l9zrjRXMTWU2B
02MwoP25IisrXKmkAVpQM682v8ujm0cQ5GaMLPsORLXTcfDnSwRoea5SQYdaD+0
OHd0o6WJBr28TbvnL+IugeFM41YHW/prWoyqpELxa+8HVY6+XinkwvhjCmOFLytC
7fpQpZLPSrLHf0xb0I5wFKbzjzP/h+olkxWB0sIEZ/fFokwSDpd8CRoJuo/hLHGU
rG6h0xFPs+GMP/pIKU0eB1TlatU40jd30zac1TeaLnfnZnwZ8rZsHMNjxeijil8sC
qcNBZjq1m+LJGjoM2Y5o17zjDLNxVCI3BpjhWNnwG04vBpXl3abySGfLJNT1TtsM
EeYHbo30ukGCP5oawFk64n5cT86y+S7VbofLMxQdcAqwfLZnnVa12oQxcZSzi1fQ
oN0i5m2ekW0adQf20espFrswFG5N4nS0w93UbpIgI59QW4dQEvdB/fEGs/AqQ04D
m5qjmIu7NHqYq5EpuQINBGJkFtwBEAC7a6hcgW5L3CbazlNnw386nLCP17gNzi+d
8ZdR2FrJ5TIPodu4NoRrjS65npVlyqY5SCxLubHCfg0ntsJBUhoQ2Zy1xj0lX5ca
6JoJ7zqq6eaw3vWTVhJviPesB0rn6owbK6TAFd7Nb7rszMXSp0XXkXC7vVbSUEgQ
KjYVP+VzBLTb2zxww+Fw5e55C6oBZdB2st72RuNZnshwiQ0RgF5rnWLJkmSD6Yjf
Jk0x6uuQyUbhRW68JnjzaUuIqUzBRVgYIfCRI5SnEEQzLiH9f5TkbmUefBXPuKg
TQybufzuWUXN0pxK7re8VfGW5vDbU0LbbsbdNp3+6lUNLYSvMhcRgo1YAWbXm9lh
/5smWb3zwTuoK14G0P+ov7p/0/4vcF+n7wMa3LwTaWst/FQXrH6Fb93u/Vye5Cch
/ep+HIDKCCR3TqSJ4V/MhBBH+dM94k5Io5/5uz2rljgbw9UW9vKfpuXqLI0wocCx
k1JGtRyEqspnJuk2ohJP/TTDc+jz9oCt5wxLAoZ6oqqvqYBEpx40BQ82arsZetOI
mSfjqVQ0Bh+0zDFqplsNDsxaiTRSK35D+ssUg5Lq9KId823hP+aYKGdJtqx7TpU1
xzwt0PqTxEO5Fuk5RSnSrGCDNOYAyqij1B5tmID3mEv27FNKE7wCRwPyVr0IcM5U
```

Рис. 3.4. вывод ключа для добавления в Github

4. Настроить подписи git.

Используя введённый email, указываем Git применять его при подписи КОММИТОВ

```
[advoropaeval@fedora ~]$ git config --global user.signingkey 26B1038C0DBFEA09  
[advoropaeval@fedora ~]$ git config --global commit.gpgsign true  
[advoropaeval@fedora ~]$ git config --global gpg.program $(which gpg2)  
[advoropaeval@fedora ~]$ gh auth login
```

Рис. 4.1. Настройка автоматических подписей коммитов git

5. Зарегистрироваться на Github.

Учетная запись на Github была создана заранее, поэтому просто входим в аккаунт.

```
[advoropaeval@fedora ~]$ gh auth login  
? What account do you want to log into? GitHub.com  
? What is your preferred protocol for Git operations? SSH  
? Generate a new SSH key to add to your GitHub account? Yes  
? Enter a passphrase for your new SSH key (Optional) ****  
? How would you like to authenticate GitHub CLI? Login with a web browser  
  
! First copy your one-time code: 442F-DF23  
  
Press Enter to open github.com in your browser...  
  
✓ Authentication complete.  
- gh config set -h github.com git_protocol ssh
```

Рис. 5.1. Авторизация

6. Создать локальный каталог для выполнения заданий по предмету Создание репозитория курса.

```
[advoropaeva@fedora Операционные системы]$ git clone --recursive git@github.com:alyona-V/study_2021-2022_os-intro.git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 20 (delta 2), reused 15 (delta 2), pack-reused 0
Получение объектов: 100% (20/20), 12.49 КиБ | 121.00 КиБ/с, готово.
Определение изменений: 100% (2/2), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/advoropaeva/work/study/2021-2022/Операционные системы/os-intro/template/presentation»...
```

Рис 6.1. Создание репозитория курса на основе шаблона

- Переходим в каталог курса
- Удаляем лишние файлы
- Создаем необходимые каталоги
- Отправляем файлы на сервер

```
[advoropaeva@fedora Операционные системы]$ cd ~/work/study/2021-2022/"Операционные системы"/os-intro
[advoropaeva@fedora os-intro]$ rm package.json
[advoropaeva@fedora os-intro]$ make COURSE=os-intro
[advoropaeva@fedora os-intro]$ git add .
[advoropaeva@fedora os-intro]$ git commit -am 'feat(main): make course structure'
[master 726edf5] feat(main): make course structure
149 files changed, 16590 insertions(+), 14 deletions(-)
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
```

```
[advoropaeva@fedora os-intro]$ git push
Перечисление объектов: 20, готово.
Подсчет объектов: 100% (20/20), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (16/16), готово.
Запись объектов: 100% (19/19), 266.53 КиБ | 1.90 МиБ/с, готово.
```

Рис. 6.2; 6.3. Настройка каталога курса.

Контрольные вопросы

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

Системы контроля версий – это набор программного обеспечения, они предназначены для работы нескольких человек над одним проектом

2. Объясните следующие понятия VCS и их отношения: хранилище,

commit, история, рабочая копия.

- Хранилище — сервер, на котором хранится вся история изменений проекта.
- Commit — фиксация "дельта-изменений", т.е. изменений с последнего commit'а с его последующей записью как версии в истории.
- История — список всех изменений проекта с возможностью отката в любую точку истории.
- Рабочая копия — все файлы проекта, с которыми происходит основная работа.

3. В централизованных VCS необходим центральный репозиторий для хранения файлов. Примером таковых могут служить CVS и Subversion. В децентрализованных VCS наличие центрального репозитория не обязательно. Децентрализованными VCS являются Git, Bazaar и Mercurial.

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

В централизованных VCS необходим центральный репозиторий для хранения файлов. Примером таковых могут служить CVS и Subversion. В децентрализованных VCS наличие центрального репозитория не обязательно. Децентрализованными VCS являются Git, Bazaar и Mercurial.

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

Инициализация системы управления версиями git через git init. Работа над проектом используя git-flow для отдельных частей проекта. Git commit для фиксации изменений. При необходимости использование удаленного сервера для хранения с помощью remote и git push. Удаленный сервер также позволяет работать с нескольких устройств с использованием git pull.

5. Опишите порядок работы с общим хранилищем VCS.

При существующей версии проекта в хранилище, скопировать его оттуда

через `git pull`. Использовать `git-flow` для работы над частями проекта. После окончания работы зафиксировать изменения через `git commit` и загрузить в хранилище через `git push`.

6. Каковы основные задачи, решаемые инструментальным средством `git`?

Ведение истории изменений, фиксирование изменений, совмещение версий, веток и тд. Откат к прошлым версиям.

7. Назовите и дайте краткую характеристику командам `git`.

- `git init` — инициализация проекта с системой контроля версий
- `git add` — добавление файла/директории в систему контроля версий как отслеживаемое
- `git commit` — фиксация изменений в отслеживаемых файлах
- `git push` — загрузка локальной версии на сервер
- `git pull` — выгрузка актуальной версии с сервера
- `git fetch` — "часть" команды `git pull`, которая собирает актуальную версию, но не вносит её в работу
- `git merge` — слияние веток

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

Для того, чтобы просмотреть список настроенных удалённых репозиториях, вы можете запустить команду `git remote`. Она выведет названия доступных удалённых репозиториях. Если вы клонировали репозиторий, то увидите, как минимум `origin` — имя по умолчанию, которое Git даёт серверу, с которого производилось клонирование.

Получение изменений из удалённого репозитория — `Fetch` и `Pull`

Для того, чтобы добавить удалённый репозиторий и присвоить ему имя (`shortname`), просто выполните команду `git remote add <shortname> <url>`.

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

Ветка — это последовательность коммитов, в которой ведётся

параллельная разработка какого-либо функционала.

Ветки нужны, чтобы несколько программистов могли вести работу над одним и тем же проектом или даже файлом одновременно, при этом не мешая друг другу. Кроме того, ветки используются для тестирования экспериментальных функций: чтобы не повредить основному проекту, создается новая ветка специально для экспериментов.

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

Во время работы над проектом так или иначе могут создаваться файлы, которые не требуется добавлять в последствии в репозиторий. Например, временные файлы, создаваемые редакторами, или объектные файлы, создаваемые компиляторами. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл .gitignore с помощью сервисов.

Вывод

В результате работы мы изучили идеологию и применение средств контроля версий и освоили умения по работе с git.

Список литературы:

URL: <https://git-scm.com/book/ru/v2/>

URL: <https://smartiqa.ru/courses/git/lesson-3>

URL:

https://esystem.rudn.ru/pluginfile.php/1383438/mod_resource/content/4/002-lab_vcs.pdf