

Отчет к лабораторной работе №2

Common information

discipline: Операционные системы

group: НПМбд-01-21

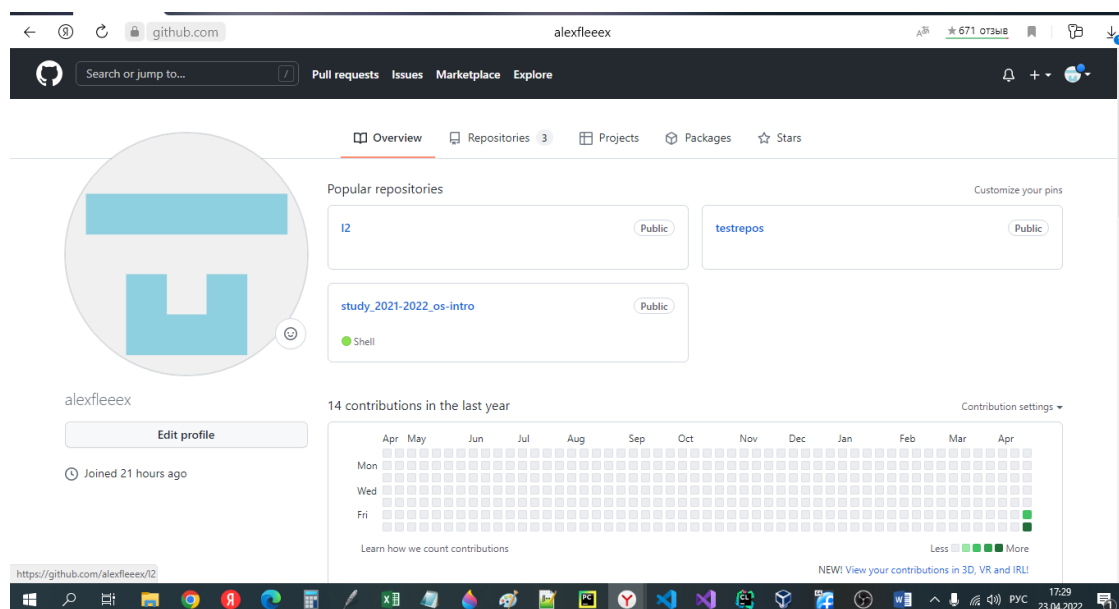
author: Ермолаев А.М.

Цель работы

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

Выполнение работы

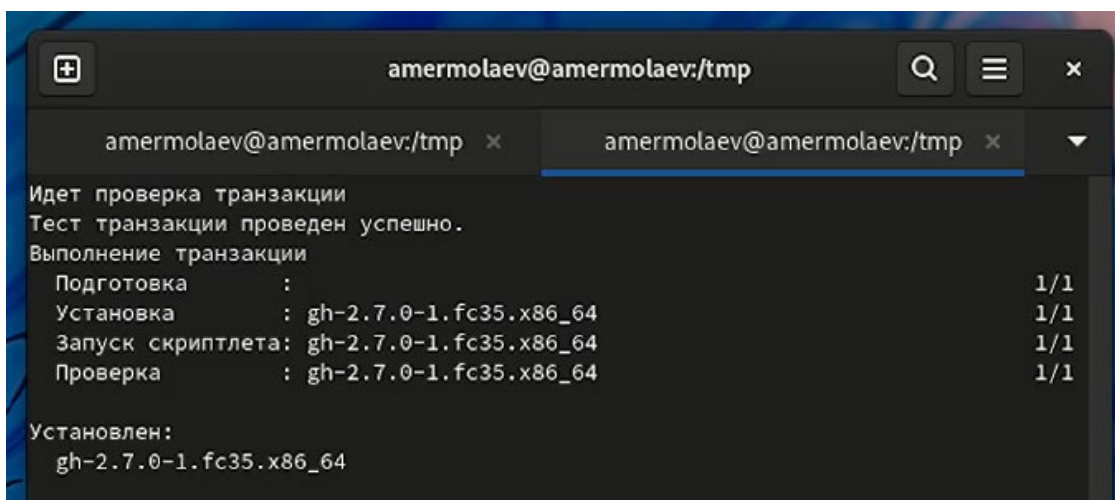
Создадим аккаунт и заполним основные данные на сайте <https://github.com>.



регистрация на github

Установим git-flow в Fedora Linux при помощи терминала при помощи данных команд:

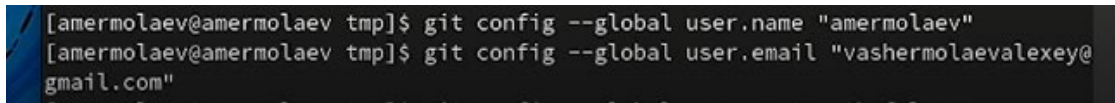
```
cd /tmp
wget --no-check-certificate -q
https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-
installer.sh
chmod +x gitflow-installer.sh
sudo ./gitflow-installer.sh install stable
```



```
amermolaev@amermolaev:/tmp
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка      : 1/1
Установка       : gh-2.7.0-1.fc35.x86_64 1/1
Запуск скриптлета: gh-2.7.0-1.fc35.x86_64 1/1
Проверка        : gh-2.7.0-1.fc35.x86_64 1/1
Установлен:
gh-2.7.0-1.fc35.x86_64
```

результат установки

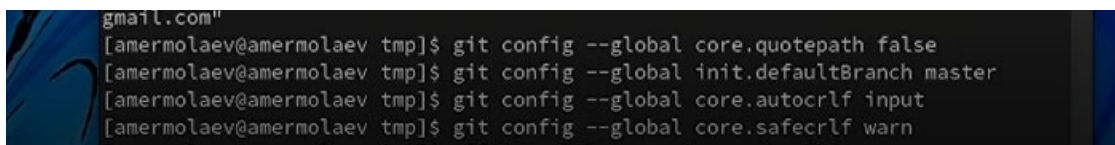
Зададим имя и email владельца репозитория:



```
[amermolaev@amermolaev tmp]$ git config --global user.name "amermolaev"
[amermolaev@amermolaev tmp]$ git config --global user.email "vashermolaevalexey@gmail.com"
```

конфигурация

Произведем другие базовые настройки.



```
gmail.com"
[amermolaev@amermolaev tmp]$ git config --global core.quotePath false
[amermolaev@amermolaev tmp]$ git config --global init.defaultBranch master
[amermolaev@amermolaev tmp]$ git config --global core.autocrlf input
[amermolaev@amermolaev tmp]$ git config --global core.safecrlf warn
```

конфигурация

Создадим ключ ssh. После этого скопируем ключ в буфер обмена и добавим его в поле «SSH Keys» в разделе Settings-SSH and GPG Keys.

```


[amermolaev@amermolaev tmp]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/amermolaev/.ssh/id_rsa):
/home/amermolaev/.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/amermolaev/.ssh/id_rsa
Your public key has been saved in /home/amermolaev/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:QTlV1IDbrJWk0bgVQ00FBz8wEhFOEzSHR1GdKUgVZs0 amermolaev@amermolaev
The key's randomart image is:
+---[RSA 4096]-----+
|      .%%&#%0 o |
|    ++*B0* E |
|   oo.@oo |
|  .+ =. |
|  S o |
|      . |
+---[SHA256]-----+
[amermolaev@amermolaev tmp]$

```

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

 SSH	<p>amermolaev@amermolaev</p> <p>SHA256:QTlV1IDbrJWk0bgVQ00FBz8wEhFOEzSHR1GdKUgVZs0</p> <p>Added on 22 Apr 2022</p> <p>Never used — Read/write</p>	<p>Delete</p>
--	--	---------------

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

Создадим ключ gpg.

```

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? 0
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: ключ 36CBC8EB060715E3 помечен как абсолютно доверенный
gpg: сертификат отзыва записан в '/home/amercolaev/.gnupg/openpgp-revocs.d/9AA4D
B2A74D664624743720236CBC8EB060715E3.rev'.
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2022-04-22 [SC]
      9AA4DB2A74D664624743720236CBC8EB060715E3
uid           amercolaev <vashercolaevalexey@gmail.com>
sub   rsa4096 2022-04-22 [E]

```

gpg ключ

После этого воспользуемся командой `gpg2 --list-secret-keys --keyid-format=long`, чтобы перечислить длинную форму ключей GPG, для которых у вас есть открытый и закрытый ключ.

```

[amercolaev@amercolaev ~]$ gpg2 --list-keys --keyid-format LONG
/home/amercolaev/.gnupg/pubring.kbx
-----
pub   rsa4096/8A2BD7EA6958ECF6 2022-04-22 [SC]
      A75AF2D120C88B791405FA938A2BD7EA6958ECF6
uid           [ абсолютно ] amercolaev <vashercolaevalexey@gmail.com>
sub   rsa4096/E898C0C32E4FAAC37 2022-04-22 [E]

pub   rsa4096/36CBC8EB060715E3 2022-04-22 [SC]
      9AA4DB2A74D664624743720236CBC8EB060715E3
uid           [ абсолютно ] amercolaev <vashercolaevalexey@gmail.com>
sub   rsa4096/5879F87D92E69A6E 2022-04-22 [E]

[amercolaev@amercolaev ~]$ gpg2 --armor --export 36CBC8EB060715E3
-----BEGIN PGP PUBLIC KEY BLOCK-----

```

gpg ключ

Затем воспользуемся командой `gpg --armor --export <идентификатор ключа>`, чтобы вывести ключ в терминал.

```
0LpgQw+NVy4I4Ep0STfaXAAQ50r3QSPNhwBJc2tQ8lunzNo6hqrU7gaTYMa1MY1d
zhLTxwV0Gc42rfLaCVjRNNzcGpXhVWeAsKVLMxg2c5XYKPpanzAXwpY6gbxQ8xn0
H80VCW/BU5Q9u4Sb7ZGM6zoFJhuh6gBYcTDhPmupkfLx8ngiUvsGmNd08BvuNMi2
R2Djfx0WXMIXEjeB8npfqmHB0M96d11/2lxJejmZaSxQ1lSDNLXmdbBEQKNJ3RDR
dJcKxHmaPzMr7f0m6aIgtI2I4r6SkdJheRhVfGy9MJnv3W0lV+PGyS+IdYv6PJ70
FMw9TC1krDLE4D6DwHAN6LkCDQRiYv5qARAA9JoRNL6LAM/S9x1W3sGo4jqr3hP
SgA4PPb5IpSlTIZG4bu60MFUIFM0LoLYMQn8aB3QhotIU3fSSUugngIhwXPSxV1x
B18yeNUIBdhL/GNy2hKmwfziUW5lGULbhHfJHFW05L22pgsL0jmVs0ini9FqArsj
ZmgqkMg8owZW/1lwddvJ7TVuBv1LtZyE2Lg5j1kRcpbfV0+xDI3JvcFNrgJ+4Nim
8VGHn+LxbUwap/0zAo4zrNe0b4xbqZqvFWSgJzXH4wam2Zp4WoQtpcB8QmUI64xe
+rHgeA2xQMfat1jWjDkv00+shoHGdyegLw8Im5W5PfNCRHh0AIur4yf20TKPvGPN
UtIAQ2WUp2zBK3r30qLHHPyWumnaA3JH9994rzL++Du+L0dgk20Q9J1hu+ArnR4
khAyrBKp4ZtbwEanRjT1w8fXKjMdFWDKGdDab440JeeLkDMX0N1cjgxbdIaYRpLr
fEHbbd/5mis+uNXN0QiMBivZ4EhfQdMHPWxp999s1AZjOzN6ACC9r0Ni5LjVqnyz
fJJLUt1qmoLnJS07+g7zUvy/shSecfk4dH/w6stkZBhfd/n00QyMH+kd+qDHALvE
pcS6AnlpHNDyrz/frfDSix4xVkkJI566EX00bY54MoCP3P5G7lVL0Netxumm8fFf
W1ruPVp0MfpacWkAEQEAAyKCNgQYAQgAIBYhBJqk2yp0ImRiR0NyAjbLy0sGBxXj
BQJiYv5qAhsMAAoJEDbLy0sGBxXjELYP/1hXdiFmgaLjX3FNobyVWYgj/Wy05zZ8
zevrBvRmBQkjLXREr01fhtRR+X/DLXqXjDbBcQjFotgr+LBnQHcssrLLnoRekNkS
+0rNAbL3sboNnnz1LQrXSq4/5FMMk0kSyV8NL4b9rEkDk+PW/Hn0IJEHPYlIyk1l
a7uzpMZuRpFXZzSf+npbvfp7MIIt5+wd/2hd7ZtpdT6j2V/5+Mb+rLh75JTqmMsyq
neraRmGGTtE0jSRgbWmggKBatqHip7BEC+BvLFEmpcFkZmKGIGmK8CwC9xw38kF
e8EkeoXHqMKRr5CiLEoz6UqFEAZ1fdDurIbwY8jlykd36+baRB4UhK8lJ9QGXSfk
dpQ/3/s/aVCs1ebsiY3MPYsdKbP3B0uSxLDHiXOFNRbyzo5BiNU8T2eF+cm8AV6V
0VTIMkmL7L7XQRwXkKy6lx52Yo9JzGRlg1lgZhMibG3H3jN05fIZIdfPz2zk6JwC
Xv8qdv/RLBLX
=SSE9
```


gpg ключ

Скопируем свой ключ GPG, начиная с `—BEGIN PGP PUBLIC KEY BLOCK—` и заканчивая `—END PGP PUBLIC KEY BLOCK—`, и добавим его в поле «GPG Keys» в разделе Settings-SSH and GPG Keys.

GPG keys

New GPG key

This is a list of GPG keys associated with your account. Remove any keys that you do not recognize.



Email address: vashermolaevaalexey@gmail.com
Key ID: 36CBC8E060715E3
Subkeys: 5879F87D92E69A6E
Added on 22 Apr 2022

Delete

Learn how to [generate a GPG key and add it to your account](#).

gpg ключ

Настроим автоматические подписи коммитов git.


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

Авторизуемся и настроим gh, параллельно получив специальный токен.

```
[amermolaev@amermolaev ~]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'workflow'.
? Paste your authentication token: *****
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as alexfleex
```

Авторизация и настройка gh

Создадим шаблон рабочего пространства.

```
[amermolaev@amermolaev ~]$ mkdir -p ~/work/study/2021-2022/"Операционные системы"
[amermolaev@amermolaev ~]$ cd -p ~/work/study/2021-2022/"Операционные системы"
bash: cd: -p: недопустимый параметр
cd: использование: cd [-L|[-P [-e]] [-@]] [каталог]
[amermolaev@amermolaev ~]$ cd ~/work/study/2021-2022/"Операционные системы"
[amermolaev@amermolaev Операционные системы]$
```

создание шаблона

После этого скопируем шаблон в собственный удаленный репозиторий и загрузим файлы в собственный каталог.

```
[amermolaev@amermolaev Операционные системы]$ gh repo create study_2021-2022_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository alexfleex/study_2021-2022_os-intro on GitHub

[amermolaev@amermolaev Операционные системы]$ git clone https://github.com/alexfleex/study_2021-2022_os-intro
Клонирование в «study_2021-2022_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 | 1.04 МБ/с, готово.
Определение изменений: 100% (2/2), готово.
[amermolaev@amermolaev Операционные системы]$
```

Удалим файлы и создадим новый каталог os-intro. Проверим статус контроля версий при помощи команды git status.

```
[amermolaev@amermolaev study_2021-2022_os-intro]$ ls
config labs LICENSE Makefile README.en.md README.git-flow.md README.md template
[amermolaev@amermolaev study_2021-2022_os-intro]$ mkdir os-intro
[amermolaev@amermolaev study_2021-2022_os-intro]$ ls
config labs LICENSE Makefile os-intro README.en.md README.git-flow.md README.md template
[amermolaev@amermolaev study_2021-2022_os-intro]$ git status
На ветке master
Ваша ветка обновлена в соответствии с «origin/master».

Изменения, которые не в индексе для коммита:
(используйте «git add/rm <файл>...», чтобы добавить или удалить файл из индекса)
(используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
удалено: package.json

нет изменений добавленных для коммита
(используйте «git add» и/или «git commit -a»)
[amermolaev@amermolaev study_2021-2022_os-intro]$
```

изменения в локальном репозитории

Произведем отправку на сервер при помощи команд add, commit и push.

```
[amermolaev@amermolaev study_2021-2022_os-intro]$ git add .
[amermolaev@amermolaev study_2021-2022_os-intro]$ git status
На ветке master
Ваша ветка обновлена в соответствии с «origin/master».

Изменения, которые будут включены в коммит:
(используйте «git restore --staged <файл>...», чтобы убрать из индекса)
удалено: package.json

[amermolaev@amermolaev study_2021-2022_os-intro]$ git commit m "feat(main): make course structure"
error: pathspec 'm' did not match any file(s) known to git
error: pathspec 'feat(main): make course structure' did not match any file(s) known to git
[amermolaev@amermolaev study_2021-2022_os-intro]$ git commit am "feat(main): make course structure"
error: pathspec 'am' did not match any file(s) known to git
error: pathspec 'feat(main): make course structure' did not match any file(s) known to git
[amermolaev@amermolaev study_2021-2022_os-intro]$ git commit -am "feat(main): make course structure"
[master 6d889f6] feat(main): make course structure
1 file changed, 14 deletions(-)
delete mode 100644 package.json
[amermolaev@amermolaev study_2021-2022_os-intro]$
```

отправка на сервер

Ответы на контрольные вопросы

Вопрос 1

Система контроля версий Git - набор программ командной строки, которые можно использовать посредством ввода команды git с различными опциями. Системы контроля версий (VCS) применяются при работе нескольких человек над одним проектом.

Вопрос 2

Хранилище – сервер, куда пользователь размещает новую версию проекта. При этом предыдущие версии не удаляются из центрального хранилища; к ним можно вернуться в любой момент. Сервер сохраняет только изменения между последовательными версиями, что позволяет уменьшить объем хранимых данных.

Коммит – команда сохранения определенной версии в локальном репозитории в рамках системы управления версиями Git.

История версии содержит информацию об изменениях и служебную информацию.

Рабочая копия - созданная клиентской программой локальная копия части данных из хранилища.

Вопрос 3

Централизованные системы используют архитектуру клиент - сервер, где один или несколько клиентских узлов напрямую подключены к центральному серверу. Например, данная стратегия реализована с Subversion.

В децентрализованных системах каждый узел принимает свое собственное решение, и конечное поведение системы является совокупностью решений отдельных узлов. Bitcoin является отличным примером.

Вопрос 4

Локальный репозиторий создается в два этапа. Сначала устанавливается предварительная конфигурация при помощи команд `git config --global user.name "Имя Фамилия"` и `git config --global user.email "work@mail"`. Для инициализации локального репозитория в каталоге необходимо воспользоваться командой `git init`.

Вопрос 5

Для работы с общим хранилищем VCS необходимо сгенерировать ssh и pgp ключи, разместив их в соответствующих полях системы. Если говорить о Git, то для отправки изменений на сервер надо будет воспользоваться командой `git push`.

Вопрос 6

Git решает две основные задачи: хранение информации о всех изменениях в вашем коде, начиная с самой первой строчки, и обеспечение удобства командной работы над кодом.

Вопрос 7

Наиболее часто используемые команды git:

- `git init` – создание основного дерева репозитория
- `git pull` – получение обновлений (изменений) текущего дерева из центрального репозитория
- `git push` – отправка всех произведённых изменений локального дерева в центральный репозиторий
- `git status` – просмотр списка изменённых файлов в текущей директории
- `git diff` – просмотр текущих изменения
- `git add` – добавление файла в раздел сохранения изменений
- `git commit` - сохранение добавленных изменений
- `git checkout/ branch` – работа с ветками

Вопрос 8

Локальный репозиторий:

```
git config --global user.name "Имя Фамилия"
git config --global user.email "work@mail"
cd
mkdir tutorial
cd tutorial
```



```
git init
echo 'hello world' > hello.txt
git add hello.txt
git commit -am 'Новый файл'
```

Удаленный репозиторий:

```
mkdir -p ~/work/study/2021-2022/"Операционные системы"
cd ~/work/study/2021-2022/"Операционные системы"
gh repo create study_2021-2022_os-intro --template=yamadharm/course-directory-
student-template --public
git clone --recursive git@github.com:<owner>/study_2021-2022_os-intro.git os-
intro
cd ~/work/study/2021-2022/"Операционные системы"/os-intro
rm package.json
make COURSE=os-intro
git add .
git commit -am 'feat(main): make course structure'
git push
```

Вопрос 9

Ветки в Git 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.

Вопрос 10

Во время работы над проектом так или иначе могут создаваться файлы (например, объектные файлы или файлы библиотек), которые не требуется добавлять в последствии в репозиторий. Для таких файлов можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл .gitignore.

Вывод

В рамках выполнения работы я изучил идеологию и применение средств контроля версий и освоил умения по работе с git.