

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**  
**Факультет физико-математических и естественных наук**  
**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**  
**К ЛАБОРАТОРНОЙ РАБОТЕ № 2**  
**Дисциплина: Операционные системы**

**Студент: Ермолаев А. М.**  
**Группа: НПМбд-01-21**

**МОСКВА**

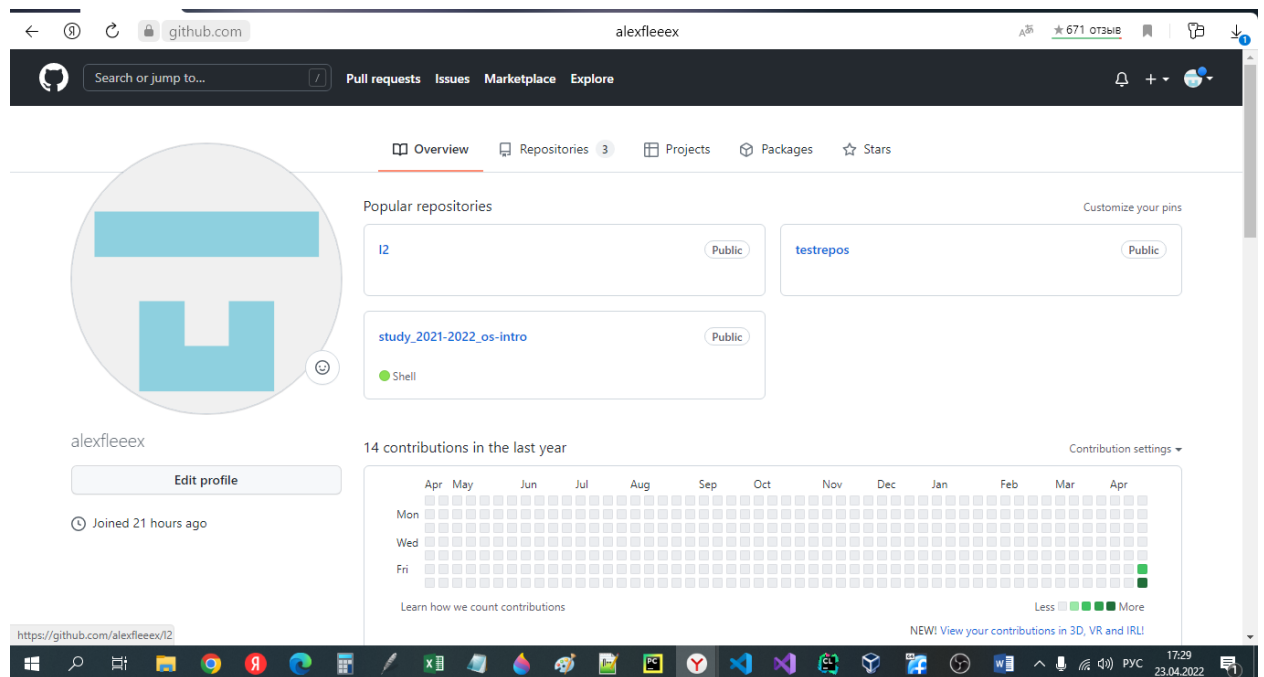
**2022 г.**

## Цель работы

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

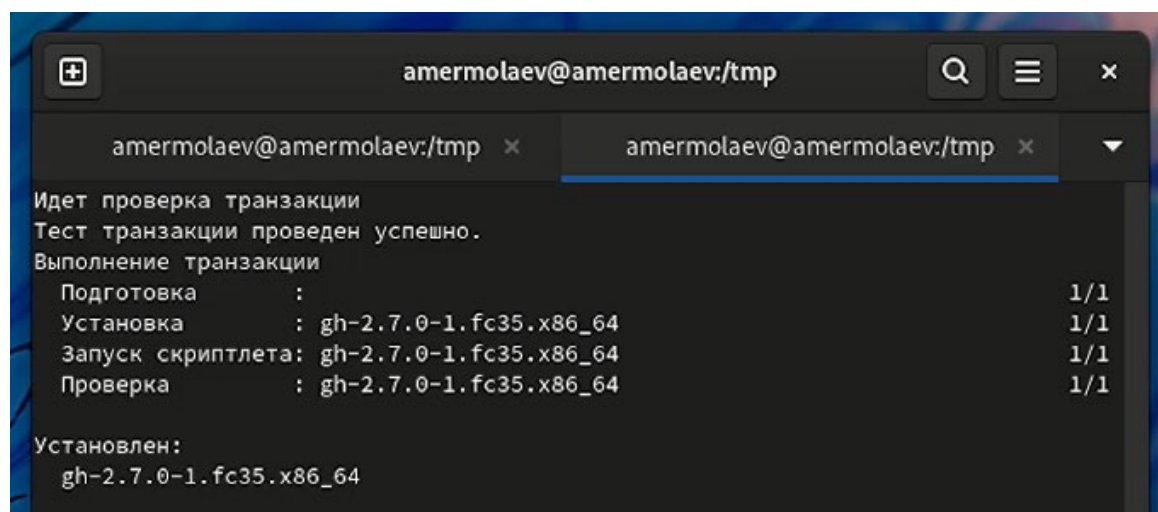
## Выполнение работы

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



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
```



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"
```

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
```

Создадим ключ ssh.

```
[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]-----+
|  .%&#%O o |
| ++*BO* E |
| oo.@oo |
| .+ =. |
| S o |
| . |
+---[SHA256]-----+
[amermolaev@amermolaev tmp]$
```

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

## SSH keys

[New SSH key](#)

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



amermolaev@amermolaev  
SHA256:QTLV1IDbrJWk0bgVQ00FBz8wEhFOEzSHR1GdKUgVZs0  
Added on 22 Apr 2022  
Never used — Read/write

[Delete](#)

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

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/amermlaev/.gnupg/openpgp-revocs.d/9AA4D  
B2A74D664624743720236CBC8EB060715E3.rev'.  
открытый и секретный ключи созданы и подписаны.

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

```
[amermolaev@amermolaev ~]$ gpg2 --list-keys --keyid-format LONG
/home/amermolaev/.gnupg/pubring.kbx
-----
pub   rsa4096/8A2BD7EA6958ECF6 2022-04-22 [SC]
      A75AF2D120C88B791405FA938A2BD7EA6958ECF6
uid           [ а6солютно ] amermolaev <vashermolaevalexey@gmail.com>
sub   rsa4096/E898C0C32E4FAAC3 2022-04-22 [E]

pub   rsa4096/36CBC8EB060715E3 2022-04-22 [SC]
      9AA4DB2A74D664624743720236CBC8EB060715E3
uid           [ а6солютно ] amermolaev <vashermolaevalexey@gmail.com>
sub   rsa4096/5879F87D92E69A6E 2022-04-22 [E]

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


[illegible]

Скопируйте 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: vashermolaevalexey@gmail.com

Key ID: 36CBC8EB060715E3

Subkeys: 5879F87D92E69A6E

Added on 22 Apr 2022

Delete

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

Настроим автоматические подписи коммитов git.

```
[amermolaev@amermolaev ~]$ git config --global commit.gpgsign true
[amermolaev@amermolaev ~]$ git config --global gpg.program $(which gpg2)
[amermolaev@amermolaev ~]$
```

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
```

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]$
```

После этого скопируем шаблон в собственный репозиторий.

```
[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/alexfleeex/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]$
```

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]$
```

## 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 объём 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=yamadharma/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.