

Отчет по лабораторной работе №2  
Дисциплина: Операционные системы

Студент: Чепелев Алексей Николаевич

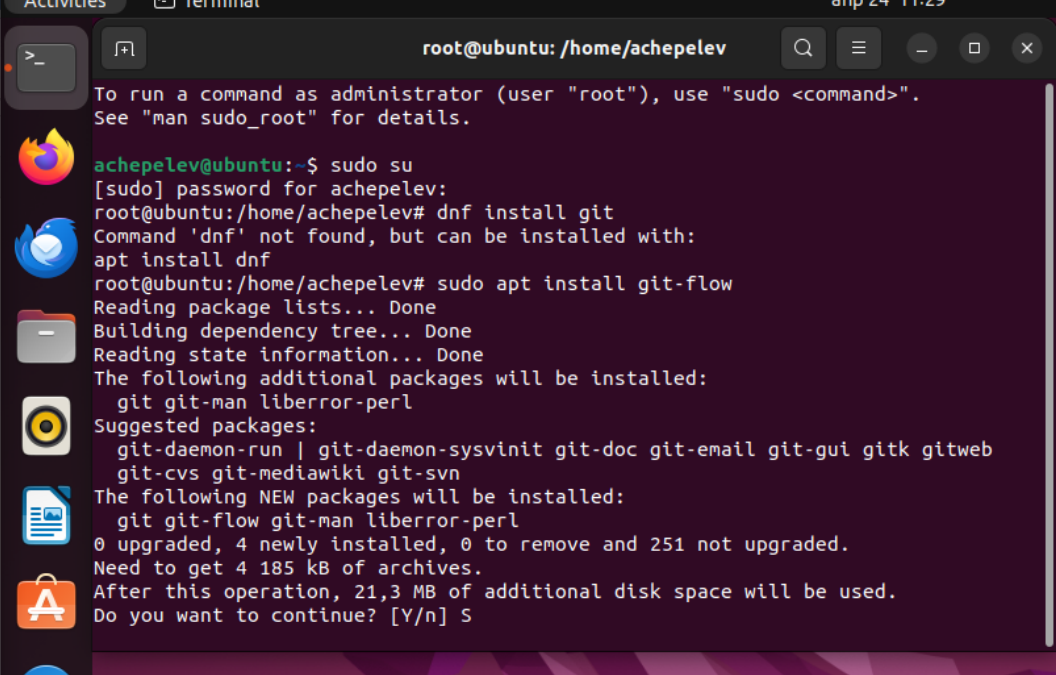
Группа: НПМбв-02-20

Москва 2024 г.

## Цель работы

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

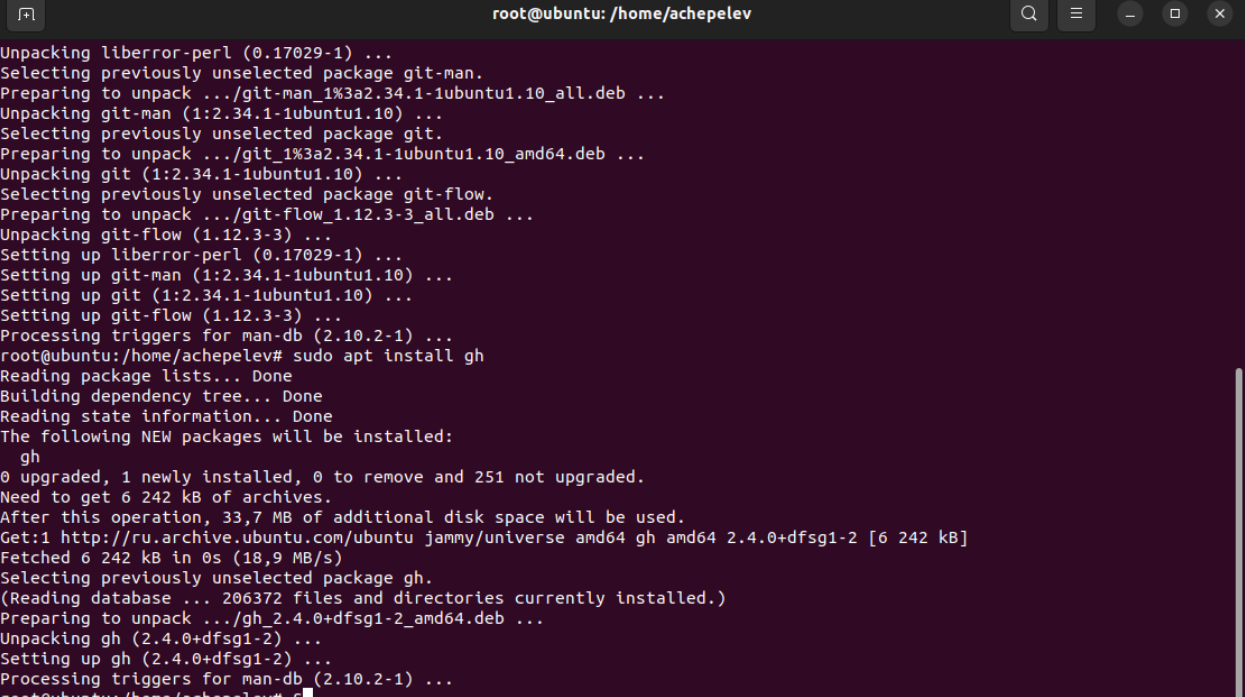
## Установка git



```
root@ubuntu: /home/achepelev
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

achepelev@ubuntu:~$ sudo su
[sudo] password for achepelev:
root@ubuntu:/home/achepelev# dnf install git
Command 'dnf' not found, but can be installed with:
apt install dnf
root@ubuntu:/home/achepelev# sudo apt install git-flow
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  git git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb
  git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-flow git-man liberror-perl
0 upgraded, 4 newly installed, 0 to remove and 251 not upgraded.
Need to get 4 185 kB of archives.
After this operation, 21,3 MB of additional disk space will be used.
Do you want to continue? [Y/n] S
```

## Установка gh



```
root@ubuntu: /home/achepelev
Unpacking liberror-perl (0.17029-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.34.1-1ubuntu1.10_all.deb ...
Unpacking git-man (1:2.34.1-1ubuntu1.10) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.34.1-1ubuntu1.10_amd64.deb ...
Unpacking git (1:2.34.1-1ubuntu1.10) ...
Selecting previously unselected package git-flow.
Preparing to unpack .../git-flow_1.12.3-3_all.deb ...
Unpacking git-flow (1.12.3-3) ...
Setting up liberror-perl (0.17029-1) ...
Setting up git-man (1:2.34.1-1ubuntu1.10) ...
Setting up git (1:2.34.1-1ubuntu1.10) ...
Setting up git-flow (1.12.3-3) ...
Processing triggers for man-db (2.10.2-1) ...
root@ubuntu:/home/achepelev# sudo apt install gh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  gh
0 upgraded, 1 newly installed, 0 to remove and 251 not upgraded.
Need to get 6 242 kB of archives.
After this operation, 33,7 MB of additional disk space will be used.
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy/universe amd64 gh amd64 2.4.0+dfsg1-2 [6 242 kB]
Fetched 6 242 kB in 0s (18,9 MB/s)
Selecting previously unselected package gh.
(Reading database ... 206372 files and directories currently installed.)
Preparing to unpack .../gh_2.4.0+dfsg1-2_amd64.deb ...
Unpacking gh (2.4.0+dfsg1-2) ...
Setting up gh (2.4.0+dfsg1-2) ...
Processing triggers for man-db (2.10.2-1) ...
root@ubuntu:/home/achepelev# S
```

## Базовая настройка git

Зададим имя и email владельца репозитория:

```
root@ubuntu: /home/achepelev
Preparing to unpack .../git-man_1%3a2.34.1-1ubuntu1.10_all.deb ...
Unpacking git-man (1:2.34.1-1ubuntu1.10) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.34.1-1ubuntu1.10_amd64.deb ...
Unpacking git (1:2.34.1-1ubuntu1.10) ...
Selecting previously unselected package git-flow.
Preparing to unpack .../git-flow_1.12.3-3_all.deb ...
Unpacking git-flow (1.12.3-3) ...
Setting up liberror-perl (0.17029-1) ...
Setting up git-man (1:2.34.1-1ubuntu1.10) ...
Setting up git (1:2.34.1-1ubuntu1.10) ...
Setting up git-flow (1.12.3-3) ...
Processing triggers for man-db (2.10.2-1) ...
root@ubuntu:/home/achepelev# sudo apt install gh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  gh
0 upgraded, 1 newly installed, 0 to remove and 251 not upgraded.
Need to get 6 242 kB of archives.
After this operation, 33,7 MB of additional disk space will be used.
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy/universe amd64 gh amd64 2.4.0+dfsg1-2 [6 242 kB]
Fetched 6 242 kB in 0s (18,9 MB/s)
Selecting previously unselected package gh.
(Reading database ... 206372 files and directories currently installed.)
Preparing to unpack .../gh_2.4.0+dfsg1-2_amd64.deb ...
Unpacking gh (2.4.0+dfsg1-2) ...
Setting up gh (2.4.0+dfsg1-2) ...
Processing triggers for man-db (2.10.2-1) ...
root@ubuntu:/home/achepelev# git config --global user.name "VortexHum"
root@ubuntu:/home/achepelev# git config --global user.email "alexcom86@yandex.ru"
root@ubuntu:/home/achepelev#
```

Настроим utf-8 в выводе сообщений git:

```
root@ubuntu: /home/achepelev
Unpacking git-man (1:2.34.1-1ubuntu1.10) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.34.1-1ubuntu1.10_amd64.deb ...
Unpacking git (1:2.34.1-1ubuntu1.10) ...
Selecting previously unselected package git-flow.
Preparing to unpack .../git-flow_1.12.3-3_all.deb ...
Unpacking git-flow (1.12.3-3) ...
Setting up liberror-perl (0.17029-1) ...
Setting up git-man (1:2.34.1-1ubuntu1.10) ...
Setting up git (1:2.34.1-1ubuntu1.10) ...
Setting up git-flow (1.12.3-3) ...
Processing triggers for man-db (2.10.2-1) ...
root@ubuntu:/home/achepelev# sudo apt install gh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  gh
0 upgraded, 1 newly installed, 0 to remove and 251 not upgraded.
Need to get 6 242 kB of archives.
After this operation, 33,7 MB of additional disk space will be used.
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy/universe amd64 gh amd64 2.4.0+dfsg1-2 [6 242 kB]
Fetched 6 242 kB in 0s (18,9 MB/s)
Selecting previously unselected package gh.
(Reading database ... 206372 files and directories currently installed.)
Preparing to unpack .../gh_2.4.0+dfsg1-2_amd64.deb ...
Unpacking gh (2.4.0+dfsg1-2) ...
Setting up gh (2.4.0+dfsg1-2) ...
Processing triggers for man-db (2.10.2-1) ...
root@ubuntu:/home/achepelev# git config --global user.name "VortexHum"
root@ubuntu:/home/achepelev# git config --global user.email "alexcom86@yandex.ru"
root@ubuntu:/home/achepelev# git config --global core.quotePath false
root@ubuntu:/home/achepelev#
```

Зададим имя начальной ветки:

```
root@ubuntu: /home/achepelev
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.34.1-1ubuntu1.10_amd64.deb ...
Unpacking git (1:2.34.1-1ubuntu1.10) ...
Selecting previously unselected package git-flow.
Preparing to unpack .../git-flow_1.12.3-3_all.deb ...
Unpacking git-flow (1.12.3-3) ...
Setting up liberror-perl (0.17029-1) ...
Setting up git-man (1:2.34.1-1ubuntu1.10) ...
Setting up git (1:2.34.1-1ubuntu1.10) ...
Setting up git-flow (1.12.3-3) ...
Processing triggers for man-db (2.10.2-1) ...
root@ubuntu:/home/achepelev# sudo apt install gh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  gh
0 upgraded, 1 newly installed, 0 to remove and 251 not upgraded.
Need to get 6 242 kB of archives.
After this operation, 33,7 MB of additional disk space will be used.
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy/universe amd64 gh amd64 2.4.0+dfsg1-2 [6 242 kB]
Fetched 6 242 kB in 0s (18,9 MB/s)
Selecting previously unselected package gh.
(Reading database ... 206372 files and directories currently installed.)
Preparing to unpack .../gh_2.4.0+dfsg1-2_amd64.deb ...
Unpacking gh (2.4.0+dfsg1-2) ...
Setting up gh (2.4.0+dfsg1-2) ...
Processing triggers for man-db (2.10.2-1) ...
root@ubuntu:/home/achepelev# git config --global user.name "VortexHum"
root@ubuntu:/home/achepelev# git config --global user.email "alexcom86@yandex.ru"
root@ubuntu:/home/achepelev# git config --global core.quotepath false
root@ubuntu:/home/achepelev# git config --global init.defaultBranch master
root@ubuntu:/home/achepelev#
```

Параметр **autocrlf**; Параметр **safecrlf**;

```
root@ubuntu: /home/achepelev
Unpacking git (1:2.34.1-1ubuntu1.10) ...
Selecting previously unselected package git-flow.
Preparing to unpack .../git-flow_1.12.3-3_all.deb ...
Unpacking git-flow (1.12.3-3) ...
Setting up liberror-perl (0.17029-1) ...
Setting up git-man (1:2.34.1-1ubuntu1.10) ...
Setting up git (1:2.34.1-1ubuntu1.10) ...
Setting up git-flow (1.12.3-3) ...
Processing triggers for man-db (2.10.2-1) ...
root@ubuntu:/home/achepelev# sudo apt install gh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  gh
0 upgraded, 1 newly installed, 0 to remove and 251 not upgraded.
Need to get 6 242 kB of archives.
After this operation, 33,7 MB of additional disk space will be used.
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy/universe amd64 gh amd64 2.4.0+dfsg1-2 [6 242 kB]
Fetched 6 242 kB in 0s (18,9 MB/s)
Selecting previously unselected package gh.
(Reading database ... 206372 files and directories currently installed.)
Preparing to unpack .../gh_2.4.0+dfsg1-2_amd64.deb ...
Unpacking gh (2.4.0+dfsg1-2) ...
Setting up gh (2.4.0+dfsg1-2) ...
Processing triggers for man-db (2.10.2-1) ...
root@ubuntu:/home/achepelev# git config --global user.name "VortexHum"
root@ubuntu:/home/achepelev# git config --global user.email "alexcom86@yandex.ru"
root@ubuntu:/home/achepelev# git config --global core.quotepath false
root@ubuntu:/home/achepelev# git config --global init.defaultBranch master
root@ubuntu:/home/achepelev# git config --global core.autocrlf input
root@ubuntu:/home/achepelev# git config --global core.safecrlf warn
root@ubuntu:/home/achepelev# S
```

## Создание ключа *ssh*

по алгоритму *rsa* с ключём размером 4096 бит:

```
root@ubuntu:/home/achepelev# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:fjiquEWEng70t/bzWhbtJHX7vzpErDNgcS0UoPKgyw4 root@ubuntu
The key's randomart image is:
+---[RSA 4096]-----+
|          .          |
|       . . . .       |
|    oo . + . +...    |
| +.. o =. +.o.o.     |
| ... o .S...o.       |
| .. o .. .o+ ..      |
| E.o  o+o..+ .       |
| oo.  .+o= . .       |
| +o.....+ .o.o]     |
+---[SHA256]-----+
root@ubuntu:/home/achepelev#
```

по алгоритму *ed25519*:

```
root@ubuntu:/home/achepelev# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:eTaDyQUGZreiKxNMjy1xEV1mCExZ/NdNr0v1+RIVQBw root@ubuntu
The key's randomart image is:
+--[ED25519 256]--+
| o00=o+ oEo |
| oo++= . .. |
| o o ... . oo|
| . B o =. . =|
| o + o S =. +o|
| . . o o o +..|
| o . . o .|
| o . o .|
| . |
+---[SHA256]-----+
root@ubuntu:/home/achepelev#
```

## Создание ключа *gpg*

Генерируем ключ

```
root@ubuntu:/home/achepelev# gpg --full-generate-key
gpg (GnuPG) 2.2.27; 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: directory '/root/.gnupg' created
gpg: keybox '/root/.gnupg/pubring.kbx' created
Please select what kind of key you want:
  (1) RSA and RSA (default)
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
  0 = key does not expire
  <n> = key expires in n days
  <n>w = key expires in n weeks
  <n>m = key expires in n months
  <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.
```

```

Real name: Alexey
Email address: alexcom86@yandex.ru
Comment:
You selected this USER-ID:
"Alexey <alexcom86@yandex.ru>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key F103D9E6BEFF7BDA marked as ultimately trusted
gpg: directory '/root/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/root/.gnupg/openpgp-revocs.d/BFB022B7F4C53EF7C81AD148F103D9E6BEFF7BDA.rev'
sh) lic and secret key created and signed.

pub   rsa4096 2024-04-24 [SC]
       BFB022B7F4C53EF7C81AD148F103D9E6BEFF7BDA
uid           Alexey <alexcom86@yandex.ru>
sub   rsa4096 2024-04-24 [E]

root@ubuntu:/home/achepelev#

```

## Добавление GPG ключа в GitHub

```


root@ubuntu:/home/achepelev# gpg --list-secret-keys --keyid-format LONG
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
/root/.gnupg/pubring.kbx
-----
sec   rsa4096/F103D9E6BEFF7BDA 2024-04-24 [SC]
       BFB022B7F4C53EF7C81AD148F103D9E6BEFF7BDA
uid           [ultimate] Alexey <alexcom86@yandex.ru>
ssb   rsa4096/E5E349CC679F8D53 2024-04-24 [E]

root@ubuntu:/home/achepelev#

```

## Скопирую мой сгенерированный GPG ключ в буфер обмена:

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



**GPG\_KEY1**

**Email address:** alexcom86@yandex.ru

**Key ID:** F103D9E6BEFF7BDA

**Subkeys:** E5E349CC679F8D53

Added on Apr 24, 2024

Delete

## Настройка автоматических подписей коммитов git

```

-----END PGP PUBLIC KEY BLOCK-----
root@ubuntu:/home/achepelev# git config --global user.signingkey BFB022B7F4C53EF7C81AD148F103D9E6BEFF7BDA
root@ubuntu:/home/achepelev# git config --global commit.gpgsign true
root@ubuntu:/home/achepelev# git config --global gpg.program $(which gpg2)
root@ubuntu:/home/achepelev#

```



## Настройка gh (Логин в github)

```
root@ubuntu:/home/achepelev# 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 VortexHum
root@ubuntu:/home/achepelev#
```

## Создание репозитория курса на основе шаблона

```
achepelev@ubuntu:~/work/study/2023-2024$ git clone --recursive git@github.com:VortexHum/study_2023-2024_os-intro.git os-intro
Cloning into 'os-intro'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 32 (delta 1), reused 18 (delta 0), pack-reused 0
Receiving objects: 100% (32/32), 18.58 KiB | 6.19 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharm/academic-presentation-markdown-template.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharm/academic-laboratory-report-template.git) registered for path 'template/report'
Cloning into '/home/achepelev/work/study/2023-2024/os-intro/template/presentation'...
remote: Enumerating objects: 95, done.
remote: Counting objects: 100% (95/95), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 95 (delta 34), reused 87 (delta 26), pack-reused 0
Receiving objects: 100% (95/95), 96.99 KiB | 1.11 MiB/s, done.
Resolving deltas: 100% (34/34), done.
Cloning into '/home/achepelev/work/study/2023-2024/os-intro/template/report'...
remote: Enumerating objects: 126, done.
remote: Counting objects: 100% (126/126), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 126 (delta 52), reused 108 (delta 34), pack-reused 0
Receiving objects: 100% (126/126), 335.80 KiB | 1.94 MiB/s, done.
Resolving deltas: 100% (52/52), done.
Submodule path 'template/presentation': checked out '40a1761813e197d00e8443ff1ca72c60a304f24c'
Submodule path 'template/report': checked out '7c31ab8e5dfa8cdb2d67caeb8a19ef8028ced88e'
achepelev@ubuntu:~/work/study/2023-2024$ rm
```

```
achepelev@ubuntu:~/work/study/2023-2024$ rm package.json
rm: cannot remove 'package.json': No such file or directory
achepelev@ubuntu:~/work/study/2023-2024$ cd /home/achepelev/work/study/2023-2024/os-intro
achepelev@ubuntu:~/work/study/2023-2024/os-intro$ rm package.json
achepelev@ubuntu:~/work/study/2023-2024/os-intro$ echo os-intro > COURSE
achepelev@ubuntu:~/work/study/2023-2024/os-intro$ make
Usage:
  make <target>

Targets:
  list           List of courses
  prepare        Generate directories structure
  submodule       Update submodules

achepelev@ubuntu:~/work/study/2023-2024/os-intro$ git add .
achepelev@ubuntu:~/work/study/2023-2024/os-intro$ git commit -am 'feat(main): make course structure'
[master 402b021] feat(main): make course structure
  Committer: achepelev <achepelev@ubuntu.myguest.virtualbox.org>
  Your name and email address were configured automatically based
  on your username and hostname. Please check that they are accurate.
  You can suppress this message by setting them explicitly. Run the
  following command and follow the instructions in your editor to edit
  your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 2 files changed, 1 insertion(+), 14 deletions(-)
  delete mode 100644 package.json
achepelev@ubuntu:~/work/study/2023-2024/os-intro$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 300 bytes | 300.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:VortexHum/study_2023-2024_os-intro.git
   005b262..402b021  master -> master
achepelev@ubuntu:~/work/study/2023-2024/os-intro$
```

## **Ответы на вопросы:**

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

Система контроля версий (VCS) - это программное обеспечение, которое помогает отслеживать изменения в файлах и коде проекта, управлять различными версиями файлов, контролировать доступ к изменениям, а также облегчает совместную работу разработчиков над проектом.

### **2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.**

**Хранилище:** Хранилище в системе контроля версий (VCS) - это централизованное место, где хранятся все файлы проекта, его история изменений, метаданные и другая информация, управляемая VCS.

**Commit:** Commit в системе контроля версий (VCS) - это действие, при котором разработчик сохраняет текущие изменения в файле или наборе файлов в репозиторий. При коммите разработчик оставляет комментарий, описывающий сделанные изменения.

**История:** История в системе контроля версий - это список всех коммитов и изменений, которые были сделаны в проекте. В истории можно отследить, кто и когда внес какие изменения.

**Рабочая копия:** Рабочая копия в системе контроля версий представляет собой локальную копию файлов проекта на компьютере разработчика. Это набор файлов, с которым мы работаем в настоящий момент. Рабочая копия может быть изменена, но изменения не будут отправлены в репозиторий до коммита.

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

**Централизованные VCS:** В централизованных VCS существует единый центральный сервер, на котором хранится вся история проекта и куда отправляются все изменения от разработчиков. (CVS (Concurrent Versions System) и SVN (Subversion))

**Децентрализованные VCS:** В децентрализованных VCS каждый участник проекта имеет собственную копию репозитория, что позволяет им работать независимо друг от друга, выполнять коммиты локально и синхронизироваться с другими участниками по необходимости. (Git и Mercurial)

### **4. Опишите действия с VCS при единоличной работе с хранилищем:**

Создание нового репозитория > Добавление файлов > Коммит изменений > Просмотр истории изменений

### **5. Опишите порядок работы с общим хранилищем VCS:**

Клонирование репозитория > Добавление и коммит изменений > Получение обновлений > Разрешение конфликтов > Отправка изменений > Code Review



## **6. Каковы основные задачи, решаемые инструментальным средством git?**

Управление версиями; Работа с ветками; Совместная работа; Отслеживание изменений; Работа в оффлайне; Резервное копирование и восстановление; Code Review.

## **7. Назовите и дайте краткую характеристику командам git:**

**git init:** Создает новый репозиторий Git.

**git clone:** Клонирование существующий репозиторий в локальный каталог.

**git add:** Добавляет изменения в рабочий каталог в индекс (подготовленные изменения).

**git commit:** Фиксирует индексированные изменения в истории репозитория.

**git push:** Отправляет локальные изменения в удаленный репозиторий.

**git pull:** Получает изменения из удаленного репозитория и объединяет их с локальным репозиторием.

**git branch:** Показывает, создает, удаляет или переключает ветки.

**git merge:** Объединяет изменения из одной ветки в другую.

**git checkout:** Переключается между ветками или восстанавливает файлы из индекса.

**git status:** Показывает текущее состояние рабочего каталога и индекса.

**git log:** Показывает историю коммитов.

**git reset:** Позволяет отменять коммиты или сбрасывать изменения.

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

### **Работа с локальным репозиторием:**

Инициализация репозитория: `git init`

Добавление файлов в индекс: `git add file.txt`

Фиксация изменений в коммите: `git commit -m "Добавлен новый файл file.txt"`

Создание новой ветки: `git branch new-feature`

Переключение на другую ветку: `git checkout new-feature`

### **Работа с удаленным репозиторием:**

Клонирование удаленного репозитория: `git clone https://github.com/username/repository.git`

Отправка изменений в удаленный репозиторий: `git push origin main`

Получение изменений с удаленного репозитория: `git pull origin main`

Просмотр удаленных репозитория: `git remote -v`

Удаление ветки в удаленном репозитории: `git push origin --delete branch-name`

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

Изоляция изменений; Параллельная разработка; Стабильность основной ветви; История изменений.

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

Игнорирование определенных файлов при коммите в Git осуществляется с помощью файла `.gitignore`. Этот файл содержит шаблоны для файлов или папок, которые мы хотим игнорировать при фиксации изменений в репозитории.