

Отчет по лабораторной работе №1

Дисциплина: Информационная безопасность

Боровикова Карина Владимировна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.1	Системы контроля версий	7
3.2	Markdown	8
4	Выполнение лабораторной работы	9
5	Выводы	18
	Список литературы	19

Список иллюстраций

4.1	Переходим в infosec	9
4.2	Создаем репозиторий	10
4.3	Рекурсивно клонируем наш репозиторий себе на компьютер . . .	10
4.4	Действия по созданию структуры курса	11
4.5	Пробуем сделать коммит	11
4.6	Конфигурируем почту и имя	11
4.7	Делаем коммит, он прошел успешно	12
4.8	Пушим изменения на git	12
4.9	Конфигурируем Git	12
4.10	Генерация ключа rsa	13
4.11	Генерация ключа ed25519	13
4.12	Генерация ключа gpg	14
4.13	Генерация ключа gpg	14
4.14	Вывод списка ключей	15
4.15	Ключ добавлен на GitHub	15
4.16	Конфигурируем gh	15
4.17	Действия из первого файла, последовательность монтирования также были показана, на записи есть данный момент	16
4.18	Отчет на Markdown	17

Список таблиц

1 Цель работы

Целью данной лабораторной работы является подготовка рабочего пространства для следующих лабораторных работ

2 Задание

- Создать базовую конфигурацию для работы с git.
- Создать ключ SSH.
- Создать ключ PGP.
- Настроить подписи git.
- Зарегистрироваться на Github.
- Создать локальный каталог для выполнения заданий по предмету

3 Теоретическое введение

3.1 Системы контроля версий

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером.

Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий поддерживают возможность отслеживания и разре-

шения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

3.2 Markdown

Markdown (произносится маркдаун) — облегчённый язык разметки, созданный с целью обозначения форматирования в простом тексте, с максимальным сохранением его читаемости человеком, и пригодный для машинного преобразования в язык для продвинутых публикаций (HTML, Rich Text и других).

Более подробно о Git и Markdown см. в [1,2].

4 Выполнение лабораторной работы

1. Работать будем на компьютере в дисплейном классе, поэтому в настройке виртуального окружения нет необходимости, поэтому сразу создаем рабочее пространство на компьютере в соответствии с указаниями к лабораторной работе и переходим в папку infosec: (рис. 4.1).

```
kvborovikova@dk6n54 ~ $ ls
'2023-09-04 13-04-34.mkv'  public      work      Загрузки  Общедоступные
'2023-09-04 13-06-21.mkv'  public_html Видео     Изображения 'Рабочий стол'
ex03.png                 'VirtualBox VMs' Документы Музыка     Шаблоны

kvborovikova@dk6n54 ~ $ cd work
kvborovikova@dk6n54 ~/work $ ls
Julia study
kvborovikova@dk6n54 ~/work $ mkdir 2023-2024
kvborovikova@dk6n54 ~/work $ cd
kvborovikova@dk6n54 ~ $ mkdir Information Security
kvborovikova@dk6n54 ~ $ ls
'2023-09-04 13-04-34.mkv'  public      'VirtualBox VMs'  Документы  Музыка     Шаблоны
'2023-09-04 13-06-21.mkv'  public_html work             Загрузки   Общедоступные
ex03.png                 Security     Видео         Изображения 'Рабочий стол'

kvborovikova@dk6n54 ~ $ cd work
kvborovikova@dk6n54 ~/work $ ls
2023-2024 Julia study
kvborovikova@dk6n54 ~/work $ cd 2023-2024/
kvborovikova@dk6n54 ~/work/2023-2024 $ ls
kvborovikova@dk6n54 ~/work/2023-2024 $ mkdir Information\ Security
kvborovikova@dk6n54 ~/work/2023-2024 $ ls
'Information Security'
kvborovikova@dk6n54 ~/work/2023-2024 $ cd Information\ Security/
kvborovikova@dk6n54 ~/work/2023-2024/Information Security $ mkdir infosec
kvborovikova@dk6n54 ~/work/2023-2024/Information Security $ ls
infosec
kvborovikova@dk6n54 ~/work/2023-2024/Information Security $ cd infosec/
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec $
```

Рис. 4.1: Переходим в infosec

2. Создаем репозиторий по данному в задании шаблону. (рис. 4.2).

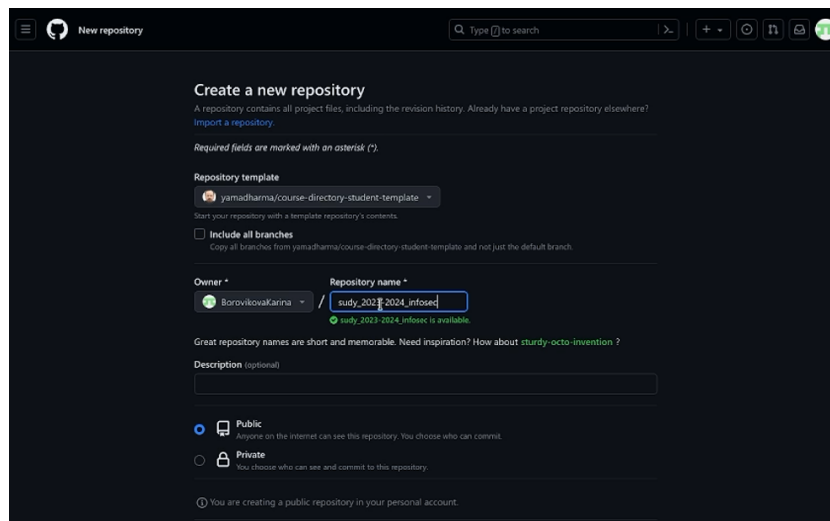


Рис. 4.2: Создаем репозиторий

3. Клонировем созданный нами репозиторий рекурсивно на наш компьютер. (рис. 4.3)

```
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec $ ls
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec $ git clone --recursive git@github.com:BorovikovaKarina/sudy_2023-2024_infosec.git
Клонирование в «sudy_2023-2024_infosec»...
```

Рис. 4.3: Рекурсивно клонируем наш репозиторий себе на компьютер

4. Далее переходим в клонированный репозиторий и выполняем действия указанные в лабораторной работе. (рис. 4.4)
 - удаляем package.json;
 - в файл COURSE записываем название курса “infosec”
 - запускаем make и создаем структуру курса

```

Получение объектов: 100% (101/101), 327.25 Киб | 1.23 Миб/с, готово.
Определение изменений: 100% (40/40), готово.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5809264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3ae11a33b1e3b2'
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec $ ls
sudy_2023-2024_infosec
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec $ cd sudy_2023-2024_infosec/
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ ls
CHANGELOG.md  COURSE  Makefile  README.en.md  README.md
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ rm package.json
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ echo "infosec" > CO
URSE
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ make
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ ls
CHANGELOG.md  COURSE  LICENSE  prepare  README.en.md  README.md
config  labs  Makefile  presentation  README.git-flow.md  template

```

Рис. 4.4: Действия по созданию структуры курса

5. Пробуем добавить изменения в индекс и закоммитить изменения, при попытке сделать коммит видим сообщение о том, что мы - неизвестный автор и о том, что коммит не выполнен. (рис. 4.5)

```

kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ git add .
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ git commit -am "Feat(main): make course structure"
Author identity unknown

*** Пожалуйста, скажите мне кто вы есть.

Запустите

  git config --global user.email "you@example.com"
  git config --global user.name "Ваше Имя"

для указания идентификационных данных аккаунта по умолчанию.
Пропустите параметр --global для указания данных только для этого репозитория.

fatal: не удалось выполнить автоопределение адреса электронной почты (получено «kvborovikova@dk6n54.(none)»)
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $

```

Рис. 4.5: Пробуем сделать коммит

6. Конфигурируем почту и имя для git. (рис. 4.6)

```

kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ git config --glob
al user.email "kvbk2002@gmail.com"
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ git config --glob
al user.name "Карина Боровикова"
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $

```

Рис. 4.6: Конфигурируем почту и имя

7. Делаем коммит. (рис. 4.7)

```

kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ git commit -am "feat
t(main): make course structure"
[master 6cc8ec6] feat(main): make course structure
151 files changed, 41045 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab1/presentation/Makefile
create mode 100644 labs/lab1/presentation/image/kulyabov.jpg

```

Рис. 4.7: Делаем коммит, он прошел успешно

8. Пушим изменения на git. (рис. 4.8)

```

kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.15 КиБ | 11.40 МБ/с, готово.
Всего 35 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:BorovikovaKarina/sudy_2023-2024_infosec.git
0780f32..6cc8ec6 master -> master
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $

```

Рис. 4.8: Пушим изменения на git

9. Продолжаем конфигурировать git. (рис. 4.9)

```

kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ git config --global c
ore.quotepath false
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ git config --global c
ore.autocrlf input
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ git config --global c
ore.safecrlf warn

```

Рис. 4.9: Конфигурируем Git

10. Генерируем ключ rsa, видим, что он уже существует, перезапишем и добавим этот ключ на GitHub. (рис. 4.10)

```

kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ ssh-keygen -t rsa -b
4096
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/k/v/kvborovikova/.ssh/id_rsa):
/afs/.dk.sci.pfu.edu.ru/home/k/v/kvborovikova/.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 /afs/.dk.sci.pfu.edu.ru/home/k/v/kvborovikova/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/k/v/kvborovikova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:HFgGhTutia4ogENwLfudBqYJ74dsjZNkNwCXK2AMxxU kvborovikova@dk6n54
The key's randomart image is:
+---[RSA 4096]-----+
|+..+E..++          |
|+++ . . +          |
|++ + .o..          |
|o.+ o o.o..        |
|. = * o o.S.        |
|o B + = .          |
|* B o . .          |
|O + . .            |
|. o .              |
+---[SHA256]-----+

```

Рис. 4.10: Генерация ключа rsa

11. Генерируем ключ ed25519, добавляем этот ключ на GitHub. (рис. 4.11)

```

kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/k/v/kvborovikova/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/k/v/kvborovikova/.ssh/id_ed25519
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/k/v/kvborovikova/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:3+2MSGu2cxPlyfy0dGHxw3lZpZl3St/safscZjRIrbk kvborovikova@dk6n54
The key's randomart image is:
+---[ED25519 256]---+
|          o|
|      o.o.|
|    o.+.*|
|   .+o.o|
|  .+o.o|
| S .+oB=|
|  o.oE+=+|
| .+o+o+o+|
| o++..+o*|
+---[SHA256]-----+

```

Рис. 4.11: Генерация ключа ed25519

12. Начинаем генерацию ключа gpg. (рис. 4.12)

```

kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/audy_2023-2024_infosec $ gpg --full-generate-key
gpg (GnuPG) 2.2.40; Copyright (C) 2022 g10 Code GmbH
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) Имеющийся на карте ключ
Ваш выбор?
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
<п> = срок действия ключа - в днях
<п>w = срок действия ключа - в недель
<п>m = срок действия ключа - в месяцев
<п>y = срок действия ключа - в 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: Karina Borovikova
Адрес электронной почты: kvbk2002@gmail.com
Примечание: meow

```

Рис. 4.12: Генерация ключа gpg

13. Генерация завершена. (рис. 4.13)

```

<п>y = срок действия ключа - в 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: Karina Borovikova
Адрес электронной почты: kvbk2002@gmail.com
Примечание: meow
Вы выбрали следующий идентификатор пользователя:
"Karina Borovikova (meow) <kvbk2002@gmail.com>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? C
Примечание: -
Вы выбрали следующий идентификатор пользователя:
"Karina Borovikova (-) <kvbk2002@gmail.com>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? O
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печатать
на клавиатуре, движения мыши, обращения к диску); 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 (печатать
на клавиатуре, движения мыши, обращения к диску); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
gpg: создан каталог "/afs/.dk.sci.pfu.edu.ru/home/k/v/kvborovikova/.gnupg/openpgp-revocs.d"
gpg: сертификат отзыва записан в "/afs/.dk.sci.pfu.edu.ru/home/k/v/kvborovikova/.gnupg/openpgp-revocs.d/3ACBC759
FFBACC1F423B9B9554CC398D83257897.rev".
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2023-09-04 [SC]
      3ACBC759FFBACC1F423B9B9554CC398D83257897
uid    Karina Borovikova (-) <kvbk2002@gmail.com>
sub    rsa4096 2023-09-04 [E]

```

Рис. 4.13: Генерация ключа gpg

14. Выводим список ключей и копируем отпечаток приватного ключа. Далее копируем ключ в буфер обмена и добавляем на GitHub (рис. 4.14)

```

kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ gpg --list-secret-key
s --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
/afs/.dk.sci.pfu.edu.ru/home/k/v/kvborovikova/.gnupg/pubring.kbx

```

Рис. 4.14: Вывод списка ключей

15. Видим, что ключ добавлен на GitHub (рис. 4.15)

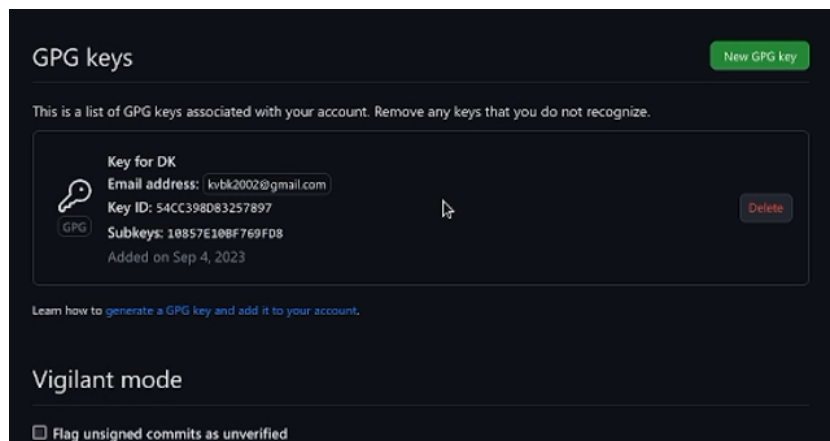


Рис. 4.15: Ключ добавлен на GitHub

16. Заканчиваем конфигурацию git с помощью gh. (рис. 4.16)

```

kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ git config --global user.signingKey 54CC398D83257897
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ git config --global commit.gpgSign true
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ git config --global gpg.program $(which gpg2)
kvborovikova@dk6n54 ~/work/2023-2024/Information Security/infosec/sudy_2023-2024_infosec $ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? SSH
? Upload your SSH public key to your GitHub account? /afs/.dk.sci.pfu.edu.ru/home/k/v/kvborovikova/.ssh/id_rsa.pub
? Title for your SSH key: Key for Infosec
? How would you like to authenticate GitHub CLI? Login with a web browser
! First copy your one-time code: 3DBE-9DC0
Press Enter to open github.com in your browser...
✓ Authentication complete.
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Uploaded the SSH key to your GitHub account: /afs/.dk.sci.pfu.edu.ru/home/k/v/kvborovikova/.ssh/id_rsa.pub
✓ Logged in as BorovikovaKarina

```

Рис. 4.16: Конфигурируем gh

17. Также сделаем некоторые действия из первого файла с настройкой виртуального окружения. (рис. 4.17)


```

kvborovikova@dk8n52 ~ $ dmesg | grep "Linux"
[ 0.153917] ACPI: Added _OSI(Linux-Dell-Video)
[ 0.154020] ACPI: Added _OSI(Linux-Lenovo-NV-HDMI-Audio)
[ 0.154126] ACPI: Added _OSI(Linux-HPI-Hybrid-Graphics)
[ 0.553513] mc: Linux media interface: v0.10
[ 0.553623] videodev: Linux video capture interface: v2.00
[ 0.553745] pps_core: LinuxPPS API ver. 1 registered
[ 0.717815] Linux agpgart interface v0.103
[ 4.379547] usb usb1: Manufacturer: Linux 5.15.91-gentoo xhci-hcd
[ 4.381812] usb usb2: Manufacturer: Linux 5.15.91-gentoo xhci-hcd
[ 198.903496] ntfs3: Enabled Linux POSIX ACLs support
kvborovikova@dk8n52 ~ $ dmesg | grep "MHz"
[ 1.681818] tsc: Refined TSC clocksource calibration: 1704.000 MHz
kvborovikova@dk8n52 ~ $ dmesg | grep "mhz"
kvborovikova@dk8n52 ~ $ dmesg | grep "Hz"
kvborovikova@dk8n52 ~ $ dmesg | grep "MHZ"
kvborovikova@dk8n52 ~ $ dmesg | grep "MHz"
kvborovikova@dk8n52 ~ $ dmesg | grep "CPU0"
[ 0.115318] CPU0: Thermal monitoring enabled (TM1)
[ 0.115484] smpboot: CPU0: Intel(R) Core(TM) i5-8400T CPU @ 1.70GHz (family: 0x6, model: 0x9e, stepping: 0xa)
kvborovikova@dk8n52 ~ $ free -h
total        used        free      shared  buff/cache   available
Mem:          7.6Gi       2.6Gi       765Mi       1.1Gi       4.3Gi       3.7Gi
Swap:         39Gi          0B        39Gi
kvborovikova@dk8n52 ~ $ dmesg | grep "Hypervisor"
kvborovikova@dk8n52 ~ $ dmesg | grep Hypervisor
kvborovikova@dk8n52 ~ $ df -T /root
Файловая система Тип      1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
/dev/sda8      ext4  484939832    73663004    386569772         17% /
kvborovikova@dk8n52 ~ $ mount

```

Рис. 4.17: Действия из первого файла, последовательность монтирования также были показана, на записи есть данный момент

18. Пишем отчет к лабораторной работе на облегченном языке разметки Markdown. (рис. 4.18)

ных – Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

Markdown

Markdown (произносится маркдаун) – облегчённый язык разметки, созданный с целью обозначения форматирования в простом тексте, с максимальным сохранением его читаемости человеком, и пригодный для машинного преобразования в языки для продвинутых публикаций (HTML, Rich Text и других).

Более подробно о Git и Markdown см. в [\[@git-versions;@md\]](#).

Выполнение лабораторной работы

1. Работать будем на компьютере в дисплейном классе, поэтому в настройке виртуального окружения нет необходимости, поэтому сразу создаем рабочее пространство на компьютере в соответствии с указаниями к лабораторной работе и переходим в папку infosec: (рис. @fig:001).

![Переходим в infosec](image/1.png){#fig:001 width=70%}

2. Создаем репозиторий по данному в задании шаблону. (рис. @fig:002).

![Создаем репозиторий](image/2.png){#fig:002 width=70%}

3. Клонировем созданный нами репозиторий рекурсивно на наш компьютер. (рис. @fig:003)

![Рекурсивно клонируем наш репозиторий себе на компьютер](image/3.png){#fig:003 width=70%}

4. Далее переходим в склонированный репозиторий и выполняем действия указанные в лабораторной работе. (рис. @fig:004)
 - удаляем package.json;
 - в файл COURSE записываем название курса "infosec"
 - запускаем make и создаем структуру курса

![Действия по созданию структуры курса](image/4.png){#fig:004 width=70%}

5. Пробуем добавить изменения в индекс и закоммитить изменения, при попытке сделать коммит видим сообщение о том, что мы – неизвестный автор и о том, что коммит не выполнен. (рис. @fig:005)

Рис. 4.18: Отчет на Markdown

5 Выводы

В ходе лабораторной работы нам удалось подготовить пространство для следующих лабораторных работ.

Список литературы

1. Управление версиями. Git. Введение в систему контроля версий git. [Электронный ресурс]. 2016. URL: <https://asrumiantceva.github.io/post/writing-technical-content/>.
2. Markdown [Электронный ресурс]. 2016. URL: <https://ru.wikipedia.org/wiki/Markdown>.