ЗАДАНИЕ 1. РАБОТА В КОМАНДНОЙ СТРОКЕ

1. Выведите сообщение «Имя Фамилия», указав Ваше имя и фамилию, в виде

нескольких строк с помощью команд echo и banner. При необходимости

установите требуемое приложение.



2. Прочитайте статью справочной системы man о пользовании справочной

системой.

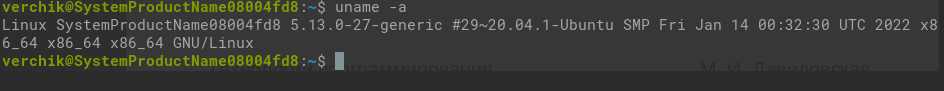
3. Прочитайте статью справочной системы о команде uname. Из какого раздела

справочника Вы прочитали статью?( 2-го)

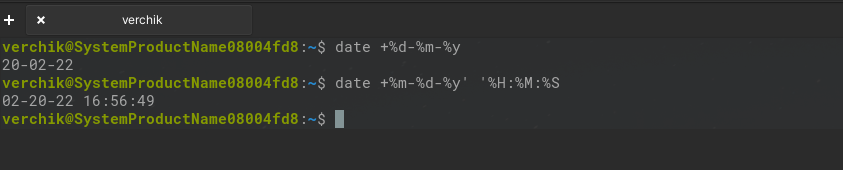


4. Определите имя машины, название и версию операционной системы, с

которой Вы работаете. Каков аппаратный тип системы?

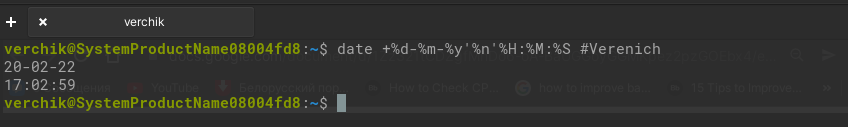


5. Выведите дату в форматах dd-mm-yy, mm-dd-yy hh:mm:ss.



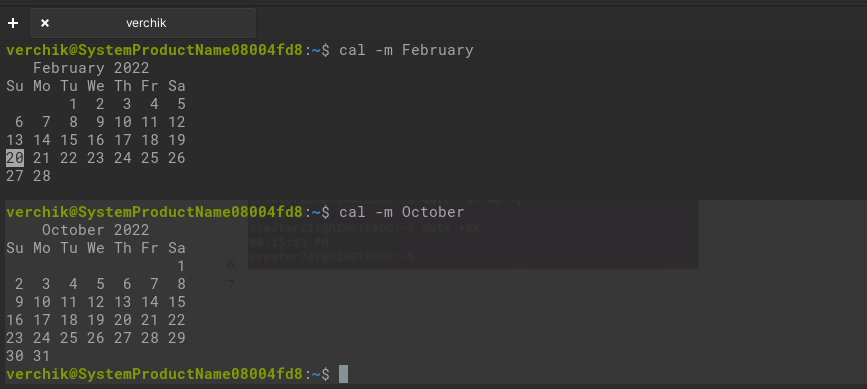
6. Выведите дату в две строки: на первой – день, месяц, год, на второй –

текущее время, снабдив вывод комментарием с Вашим именем и фамилией.



7. Выведите календарь на текущий месяц, на месяц Вашего рождения текущего

года.

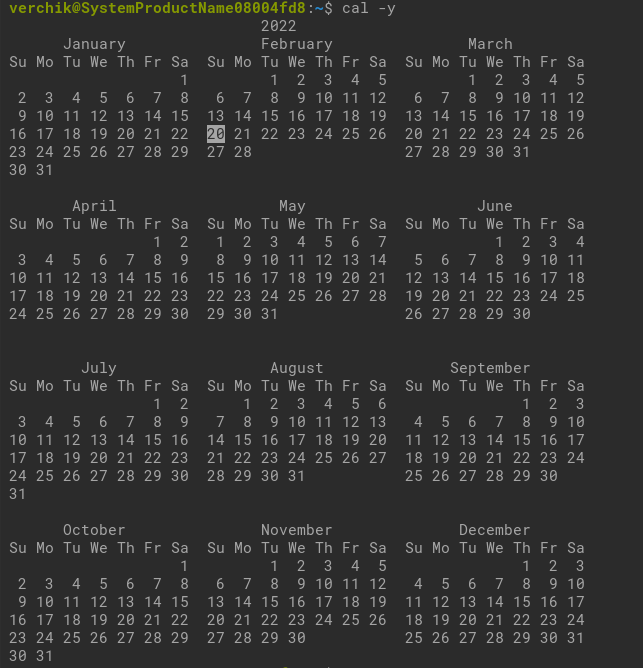


8. Выведите календарь на будущий год. В каком столбце отображаются

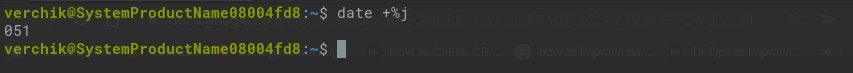
воскресенья? Какую команду использовать, чтобы неделя начиналась с

понедельника? С воскресенья?

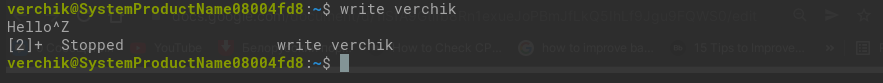
воскресенья отображаются в первосм столбце, чтобы неделя начиналась с понедельника, можем изменить локальные настройки



9. Определите порядковый номер текущего дня с начала года.



11.Используя команду write, пошлите сообщение на консоль.



12.Используя команду mesg, определите, разрешены ли сообщения на Ваш

терминал. Запретите сообщения. Какова будет реакция системы, если кто-

нибудь попытается передать Вам сообщение?

Сообщение придёт



Для запрета отправки сообщения необходимо ввести mesg n. После этого вводя mesg результатом будет is n

13.Прочитайте свою почту в консоли.

mail -u [username]

14.Привести несколько примеров команды конвертации файла \*.doc в формат

\*.html.

На Linux Elementary можно использовать утилиту Pandoc:

pandoc -t doc file.html -o file.doc

15.Привести несколько примеров изменения кодировки файла, например из

cp1251 в utf-8.

iconv -f WINDOWS-1251 -t UTF-8 -o output\_file.txt original\_file.txt

ЗАДАНИЕ 2. РАБОТА С ФАЙЛАМИ

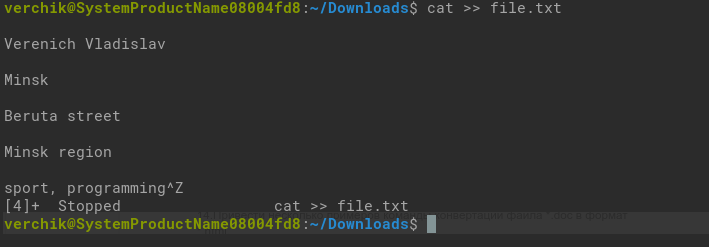
cat

1. Создавать файл dirlist.txt, состоящий из 13 строк. Нечётные строки —

пустые, чётные — содержат 2) Ваше ФИО, 4) место (город/посёлок)

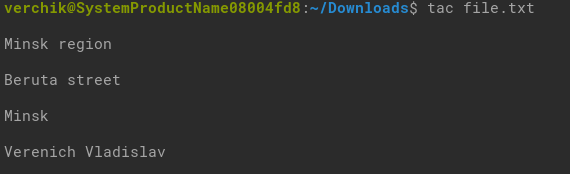
проживания, 6) улица проживания, 8) область проживания, 10) список Ваших

хобби и интересов.

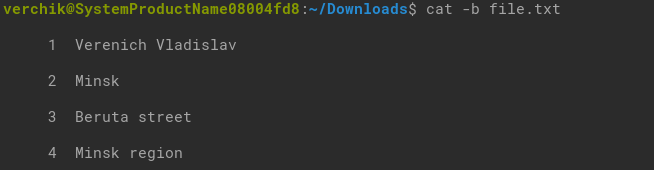


2. вывести содержимое файла dirlist.txt:

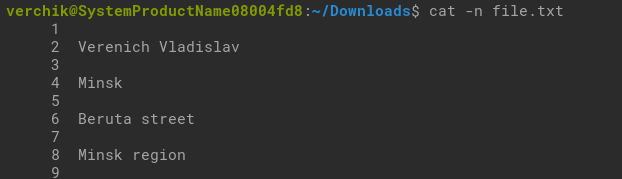
a в обратном порядке



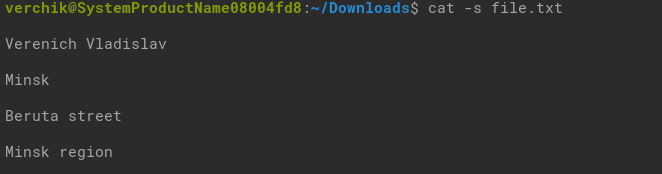
b с нумерацией непустых строк



c с нумерацией всех строк

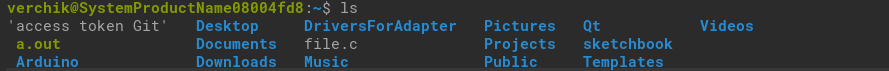


d схлопывая подряд идущие пустые строки в одну



Работа с файлами

3. Просмотрите содержимое Вашего домашнего каталога.



4. Определите имя своего домашнего каталога и его родительского каталога.

5. Определите полный путь до Вашего домашнего каталога.

6. Создайте в своём домашнем каталоге подкаталоги вида и выведите их

структуру в виде дерева:



7. Находясь в домашнем каталоге, с помощью команды touch создайте файл

macintosh в имеющемся каталоге apple и несколько файлов в каталогах

dark и green.



8. Войдите в каталог flowers. Находясь в каталоге flowers, скопируйте все

подкаталоги fruits вместе с находящимися в них файлами в специально

созданный каталог basket.



9. Находясь в каталоге flowers, удалите каталог fruits.

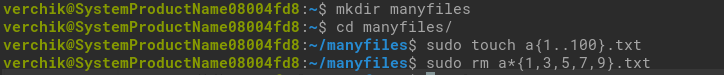


10. Создайте в домашнем каталоге директорию manyfiles. Создайте в ней с

помощью команды и регулярных выражений 100 файлов с именами a1,

a2, a3, .... a100. Цикл не используем!

Удалите только файлы с нечётными номерами.



11. Находясь в своём домашнем каталоге, создайте следующие подкаталоги,

используя регулярные выражения с помощью ОДНОЙ команды:

◦ A/B/C;

◦ A/B;

◦ A/B/C/D;

◦ A/E.

Находясь в своем домашнем каталоге, удалите все подкаталоги каталога A.

grep



12. Обновите в редакторе nano текстовый файл dirlist.txt так, чтобы он

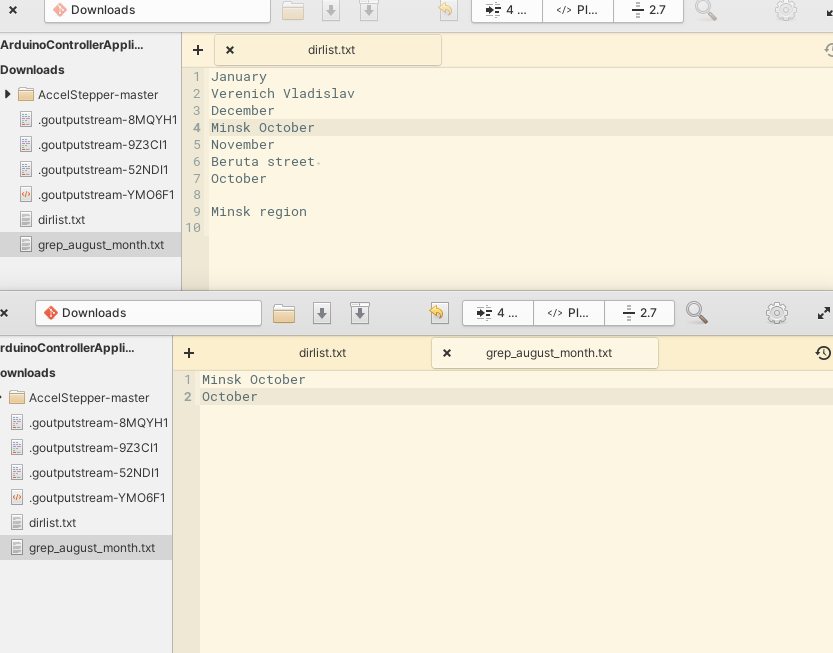
содержал в каждой строке как минимум одно название месяца. Вывести

строки файла dirlist.txt, содержащие строки с определенным месяцем

(месяц определяем по дате рождения) и записать их в файл

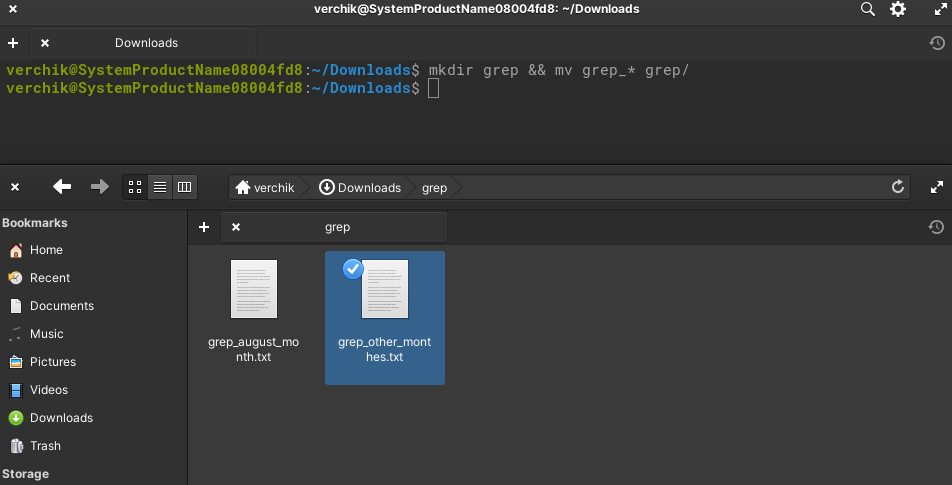
grep\_month\_name.txt





13. Записать строки, не содержащие этот месяц, в файл grep\_other\_monthes.txt.



14. Создать папку grep, переместить в нее файлы созданные в пунктах 12 и 13.

15. Создать папку mac\_os\_lab и вложенные папки (не менее 3-х уровней

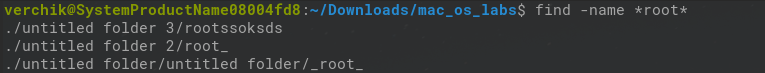
вложения и 5 вложенных папок). Создать текстовые файлы в каждой из

директорий. Ряд файлов должны содержать слово «root». Находясь в

папке mac\_os\_lab найти все файлы в этой директории и ее

поддиректориях, в которых встречается подстрока root, вывести строки с

указанием их номеров.



16. Найти все файлы в системе, содержащие в имени «sh» (с помощью команд

find и locate). Какая из команд работает быстрее, почему? Доп:

Проверить совпадают ли результаты выполнения обеих команд, чем они

отличаются, почему?

Locate работает быстрее (т.к. результат её использования зависит от базы данных файлов в файловой системе). Также она ищет не только файлы и названия директорий. С другой стороны - find имеет больше возможностей.

17. Найти файлы, изменённые за последний час.



18. Удалить весь каталог manyfiles со всеми файлами.



Математические операции в консоли

19. Используя редактор nano, создайте текстовый файл следующего

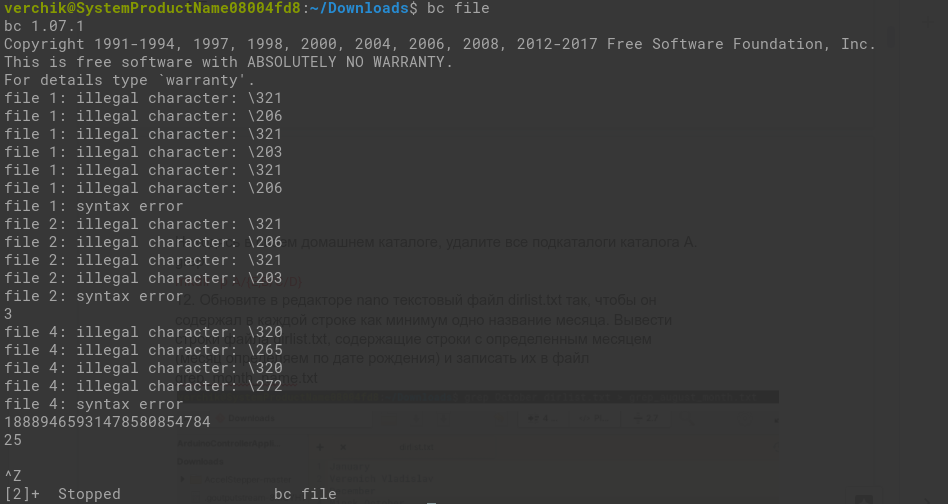
содержания:

Посчитайте все примеры из файла с помощью одной команды. (Вариантов команды

существует несколько, засчитывается каждый из них). Использовать встроенную

команду!

Команды для работы с текстом



20. Используя редактор Vi/Vim, создайте текстовый документ, содержащий

список студентов группы. Структура каждой строки: символ табуляции,

фамилия, пробел, инициалы.



21. Выведите первые 10 строк файла, последние 15 строк файла.



22. Используя команду expand, замените символы табуляции в начале строки

на соответствующее количество пробелов и сохраните с новым именем.

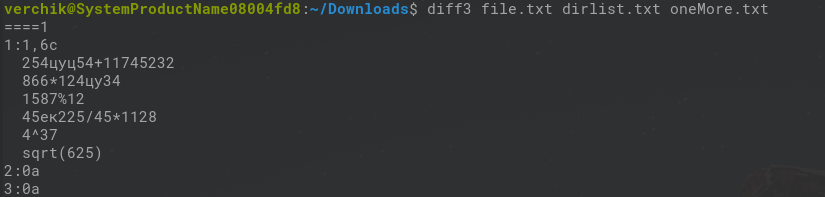


23. Используя команды diff, diff3 сравните файлы. В каких случаях

используется каждая из команд.



diff для сравнения двух файлов, а diff3 для сравнения 3



24. Для каких задач используется команды cmp, comm

Команда comm сравнивает построчно два файла на совпадение строк и на уникальные строки

Команда cmp сравнивает два файла байт за байтом

25. Используя команду split разбейте файл dirlist.txt из 10 строк на два

файла по 5 строк.



ЗАДАНИЕ 3. Изучение GitHub

1. Для работы с Git-репозиторием необходимо установить утилиту git, если

она не установлена.

2. Задать данные пользователя (имя, адрес электронной почты), в консоли на

глобальном уровне.



3. Изучить основные команды работы с git в консоли (см. СТРУКТУРА GIT).

4. Создать учётную запись на проекте github.

5. Откройте в браузере сайт https://lab.github.com

6. Активируйте курс Introduction to GitHub . При активации курса

используйте вариант курса для командной строки (см. скриншот

ниже:

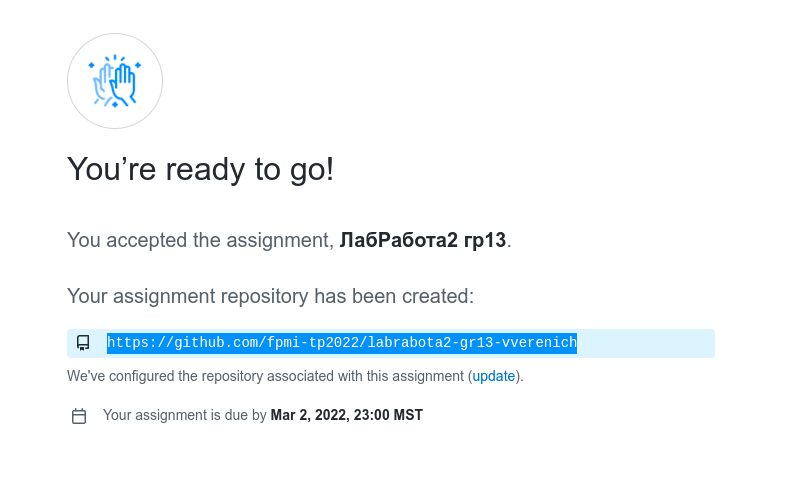
7. Выполните все задания курса.

8. Репозиторий после выполнения не закрывайте и не удаляйте.

ЗАДАНИЕ 4. РАБОТА С GIT-РЕПОЗИТОРИЕМ

1. Перейти по ссылке-приглашению, опубликованному для Вашей подгруппы в

теме 2 курса на сайте <https://edufpmi.bsu.by/>.



2. Клонировать репозиторий на свою машину в консоли. При клонировании

использовать

ссылку

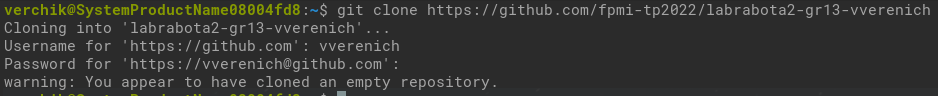
для

протокола

ssh,

например

[git@github.com](mailto:git@github.com):maryiad/github-slideshow.git.



3. Создать файл Readme. Добавить в список отслеживаемых (git add) и

зафиксировать в репозитории (закоммитить) (git commit). Данная

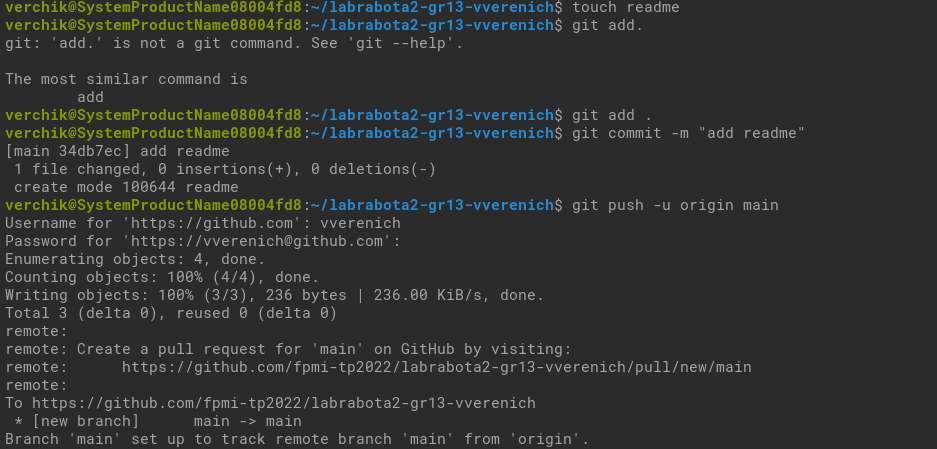
последовательность команд обеспечивает создание ветки Master1. Если не

опубликовать первый коммит и попробовать создать ветку, то ветка

Main не будет создана! Опубликовать изменения из локального репозитория

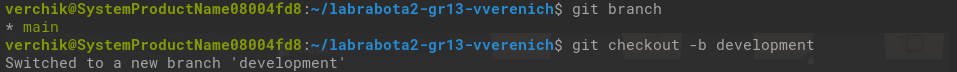
во внешнем репозитории на github, выполнив команду git push, т. е.

запушить.



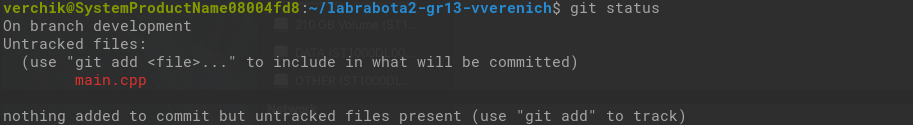
4. Создайте в локальном репозитории ветку Development в консоли и

переключитесь в неё.

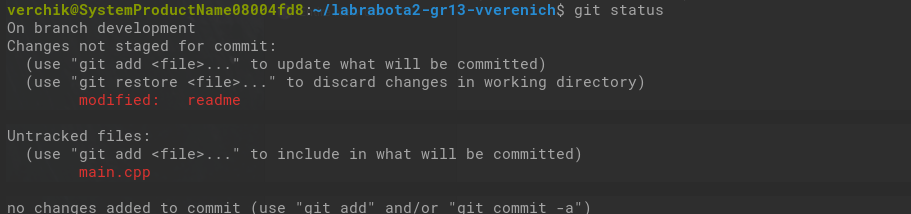


5. Добавьте в локальный репозиторий проект из задания 6 лабораторной работы

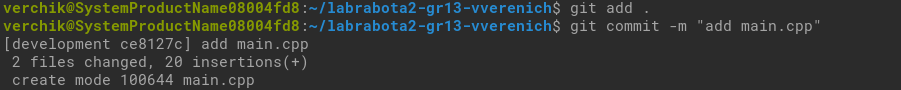
1.



6. Добавьте в файл Readme ccылку на ваш репозиторий из задания 3.

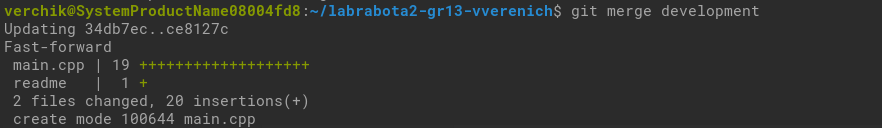


7. Выполните commit.



8. После внесения изменений слейте ветку Development и главную ветку

Master.



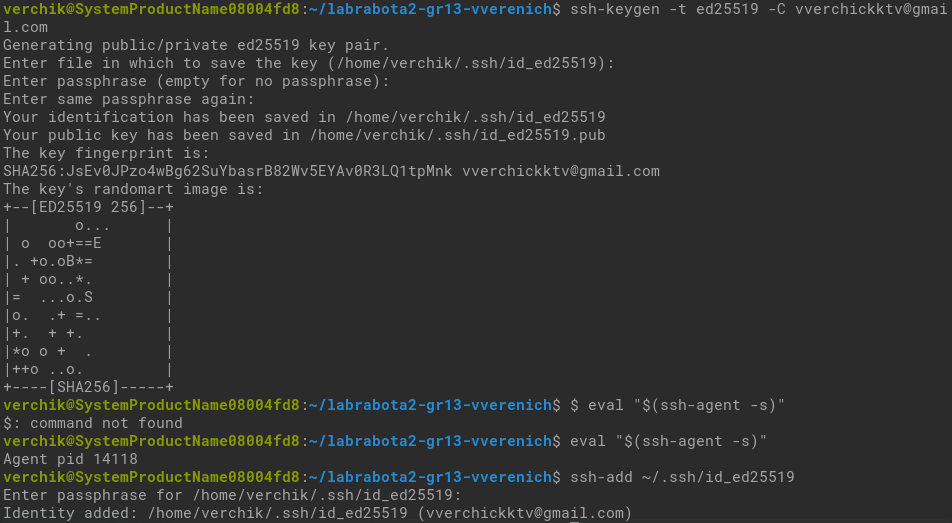
9. Отправьте изменения во внешний репозиторий



10. Изучите документацию Генерация открытого SSH-ключа в macOS и

добавление в Github account. Сгенерируйте ssh-ключ и добавьте публичный

ключ в свою учетную запись на Github.



11. Переключитесь в ветку Development.

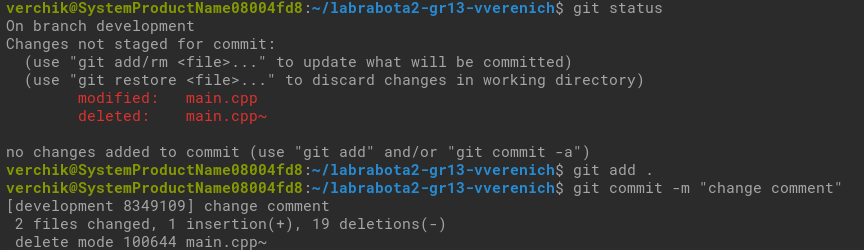
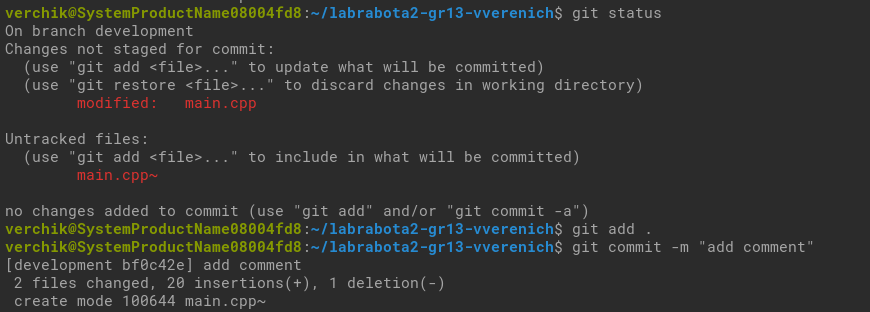


12. Выполните несколько изменений файлов (такими изменениями может

быть улучшение читаемости кода, добавление комментариев и т. д.). После

каждого атомарного изменения выполняйте команду commit в репозиторий в

консоли. К каждому коммиту добавить осмысленный комментарий, чтобы при

необходимости можно было опознать, какие были выполнены изменения.

13. После внесения изменений слейте ветку Development и главную ветку

Master.



14. Отправьте изменений во внешний репозиторий. Требуется ли ввод

пароля после добавления ssh-ключа?

На этот раз ввод ключа не потребовался

15.Выберите git-клиент с графическим интерфейсом

SourceTree/SmartGit/GitKraken/GitHub Desktop. И ознакомьтесь с

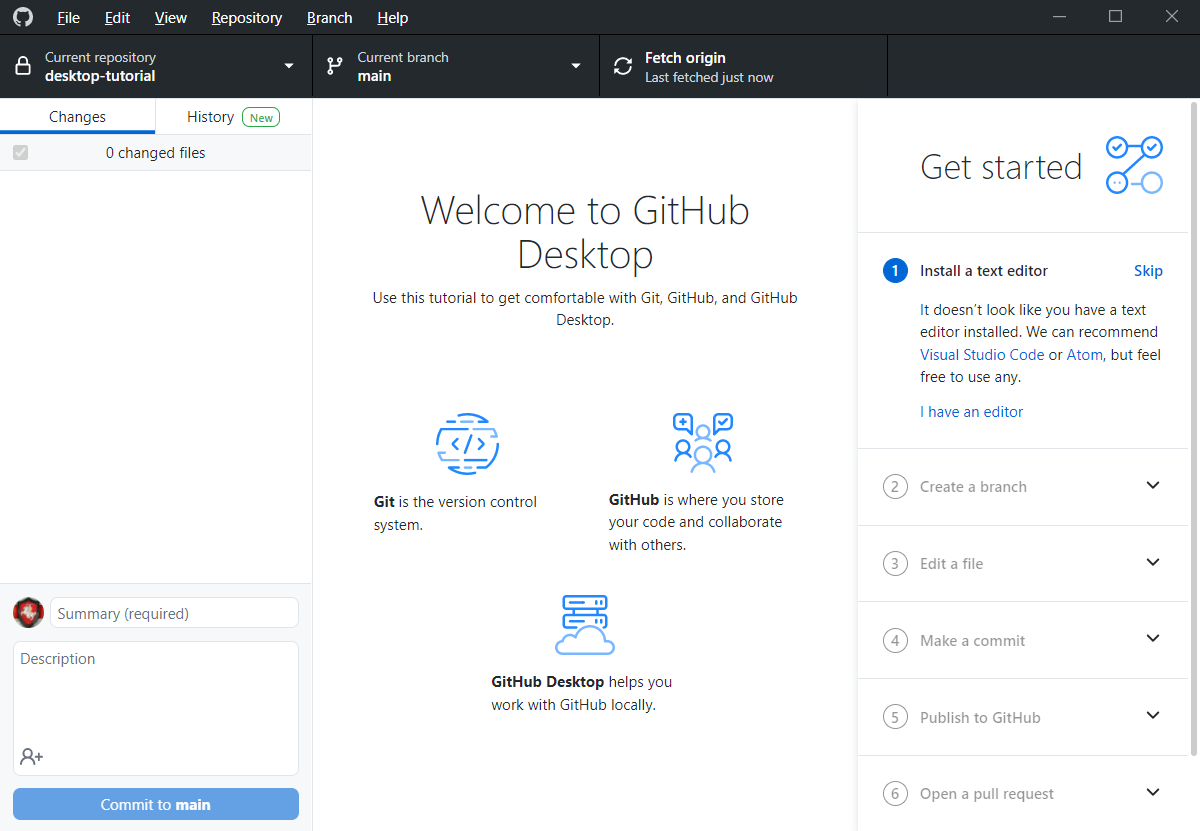
документацией, например GitKraken Git Client Tutorial For Beginners, GitKraken

Git Client – пример, Методическое руководство по работе с Git в консоли и в

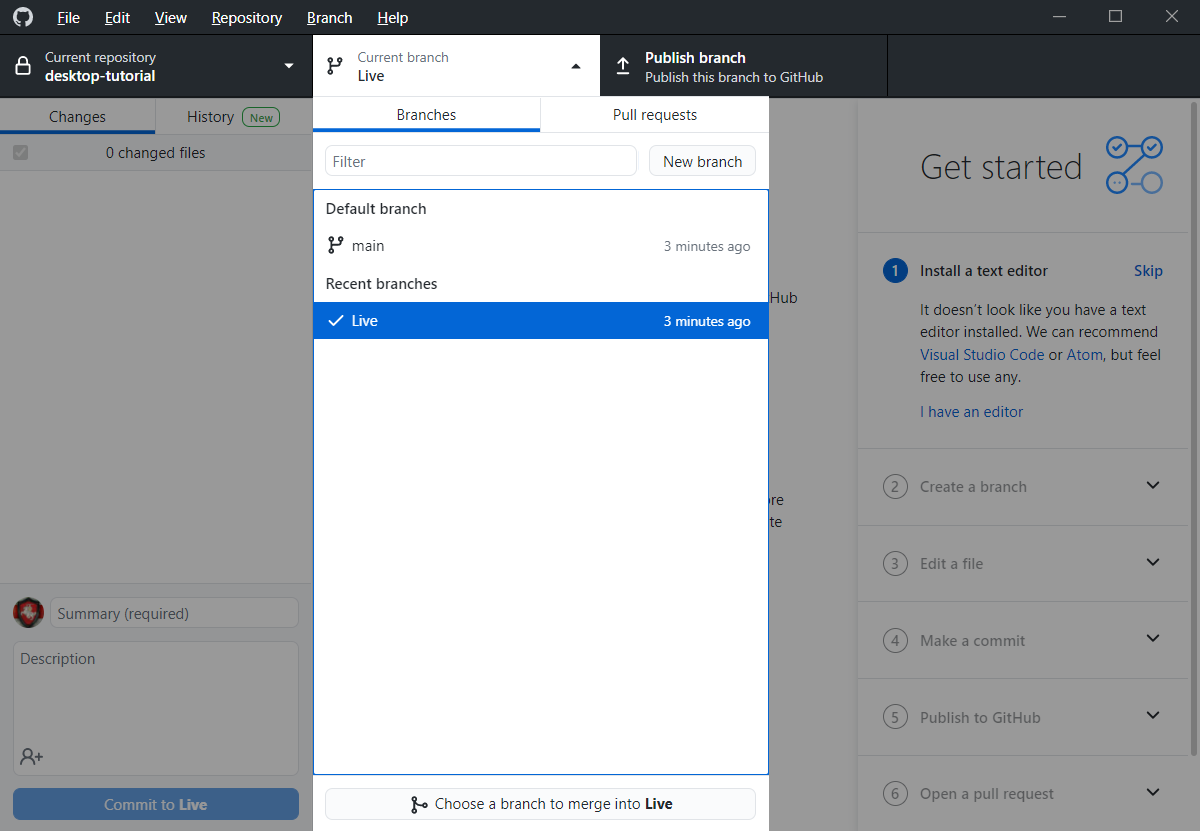
утилите SourceTree.

16. Настройте репозиторий в SourceTree/SmartGit/GitKraken/GitHub

Desktop, указав путь к локальной и внешней версии репозитория.



17.Создайте ветку Live и перейдите в неё.

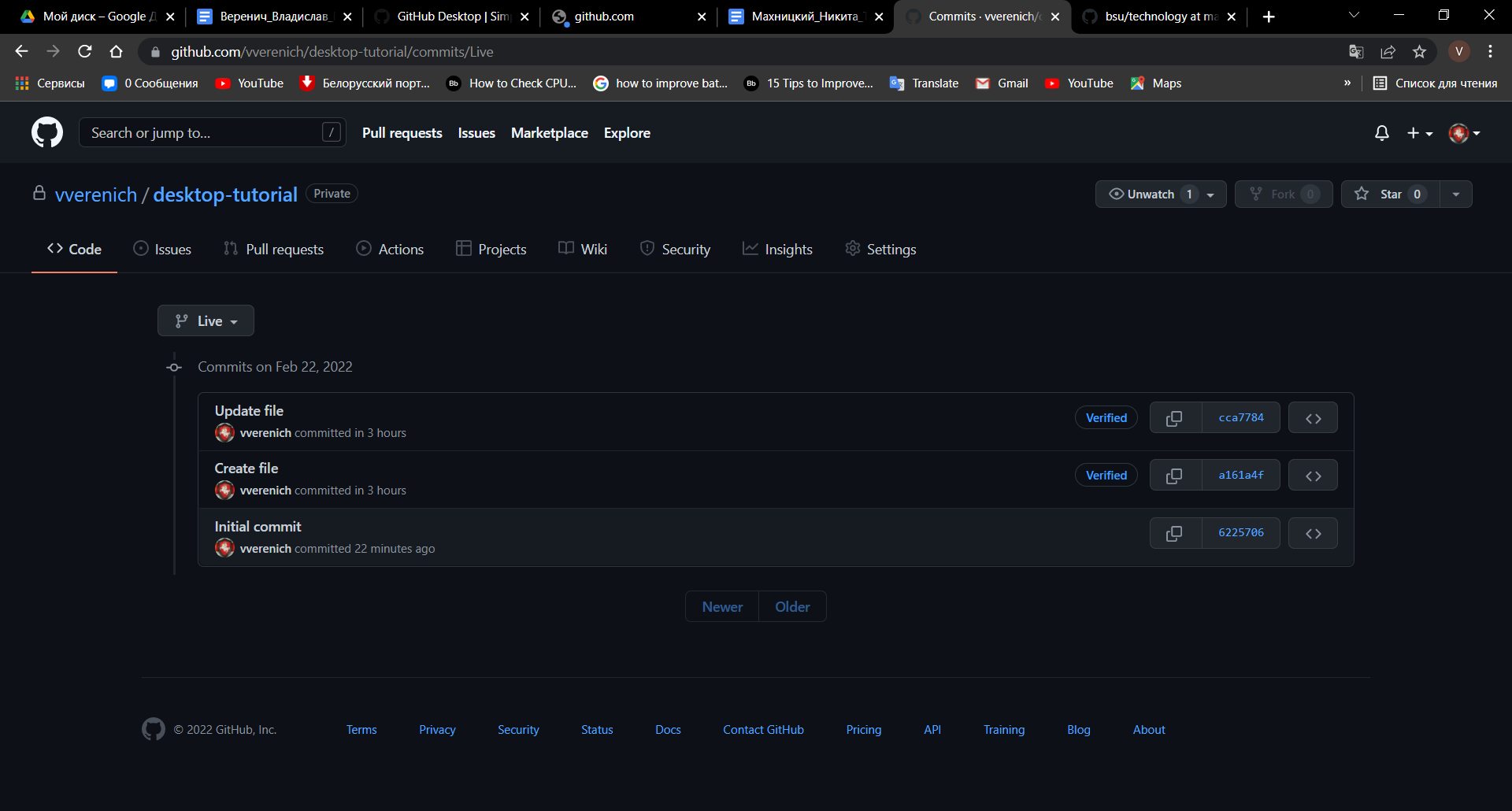
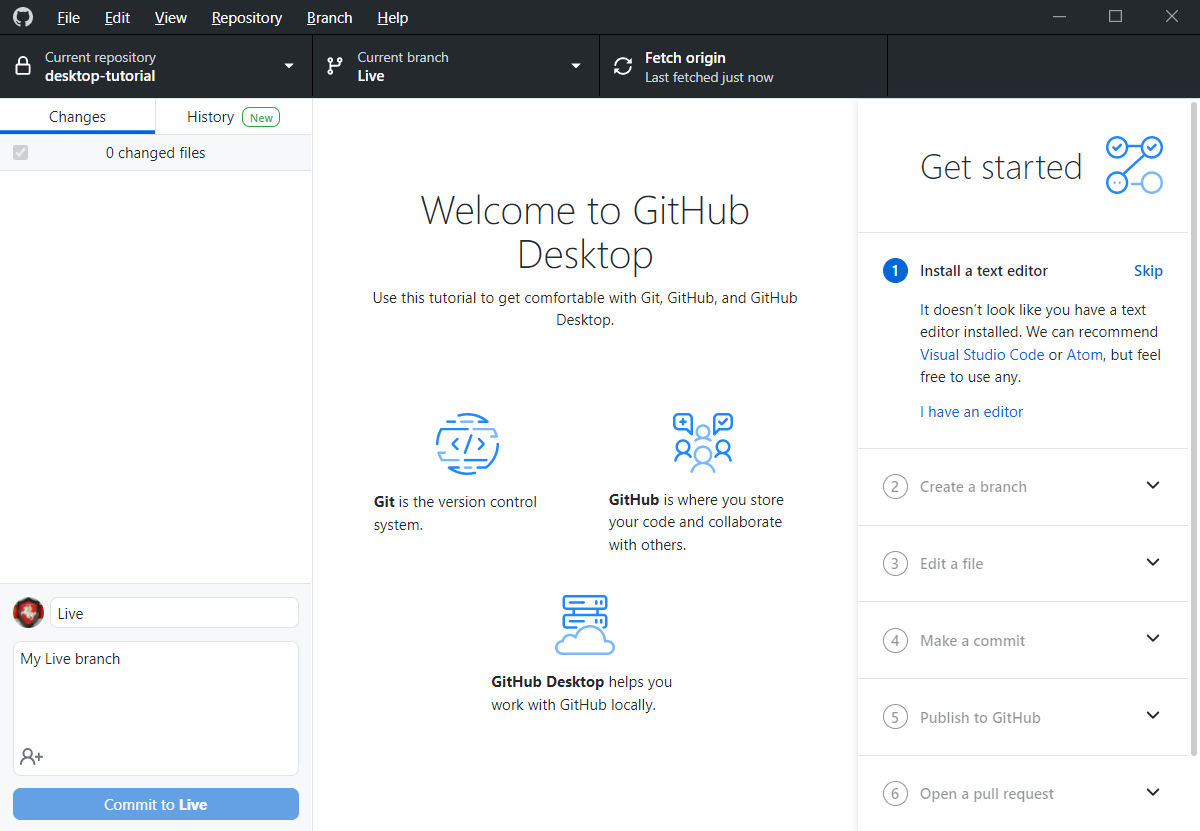


18. Внесите изменения в проект (добавьте новую функциональность) (см. п.

10 текущего задания) и после каждого атомарного изменения передавайте

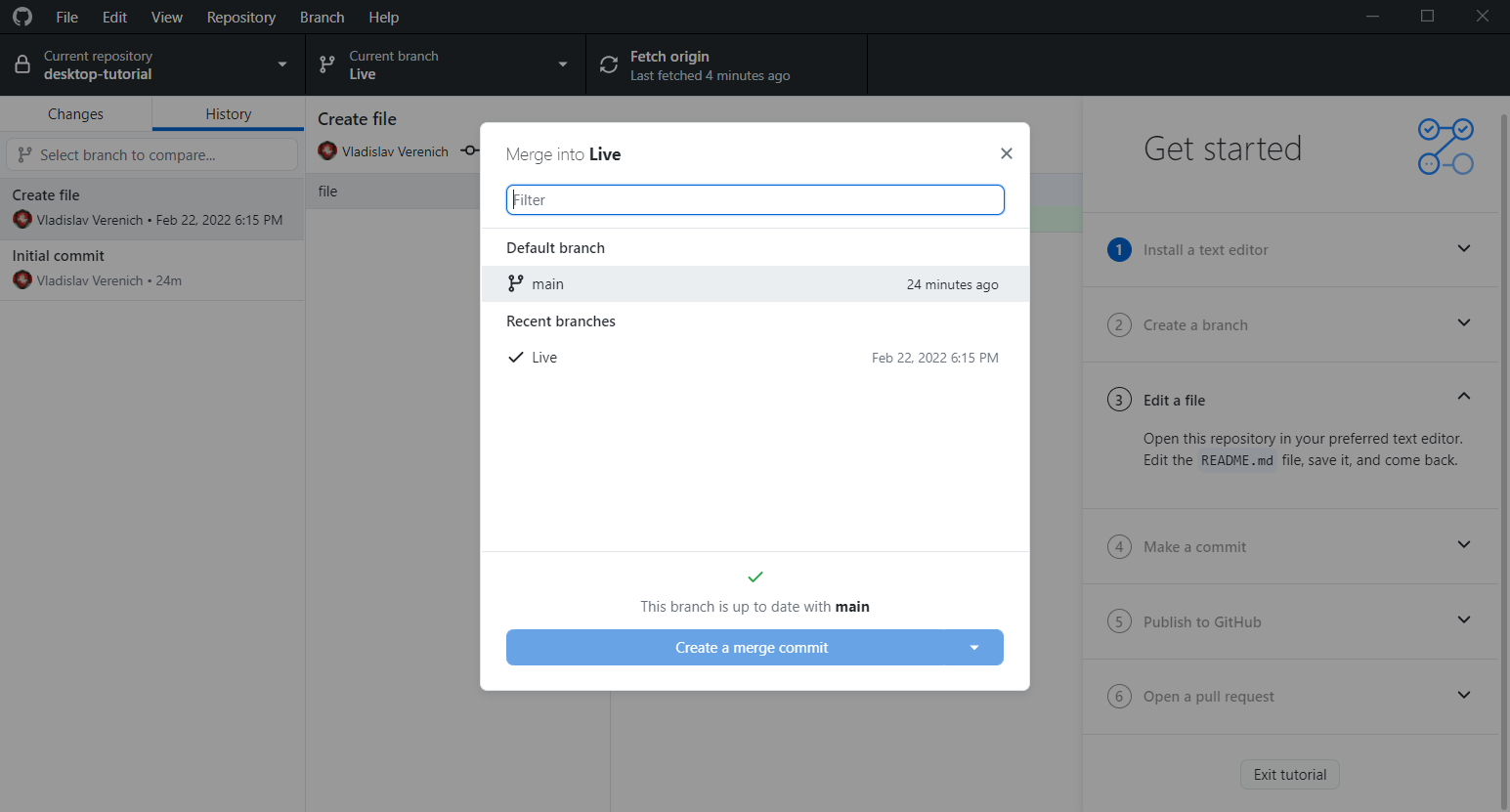
коммит в репозиторий в SourceTree/SmartGit/GitKraken/GitHub

Desktop. К каждому коммиту добавлять комментарий.

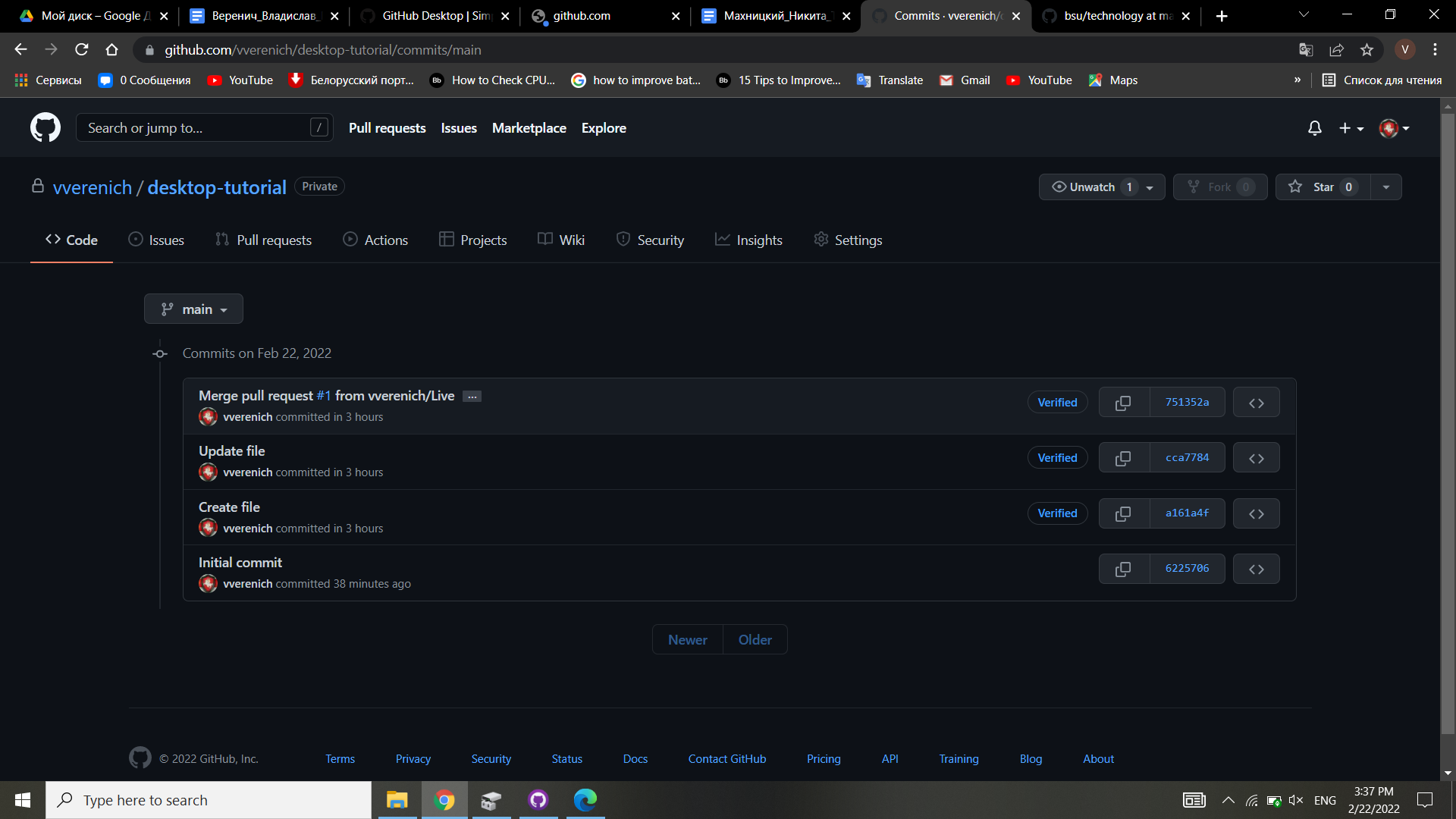


19. После внесения изменений слейте ветку Live и главную ветку master.

Проиллюстрируйте слияние двух веток в отчёте в SourceTree/SmartGit/GitKraken /GitHub Desktop.



20. Отправьте изменения во внешний репозиторий.

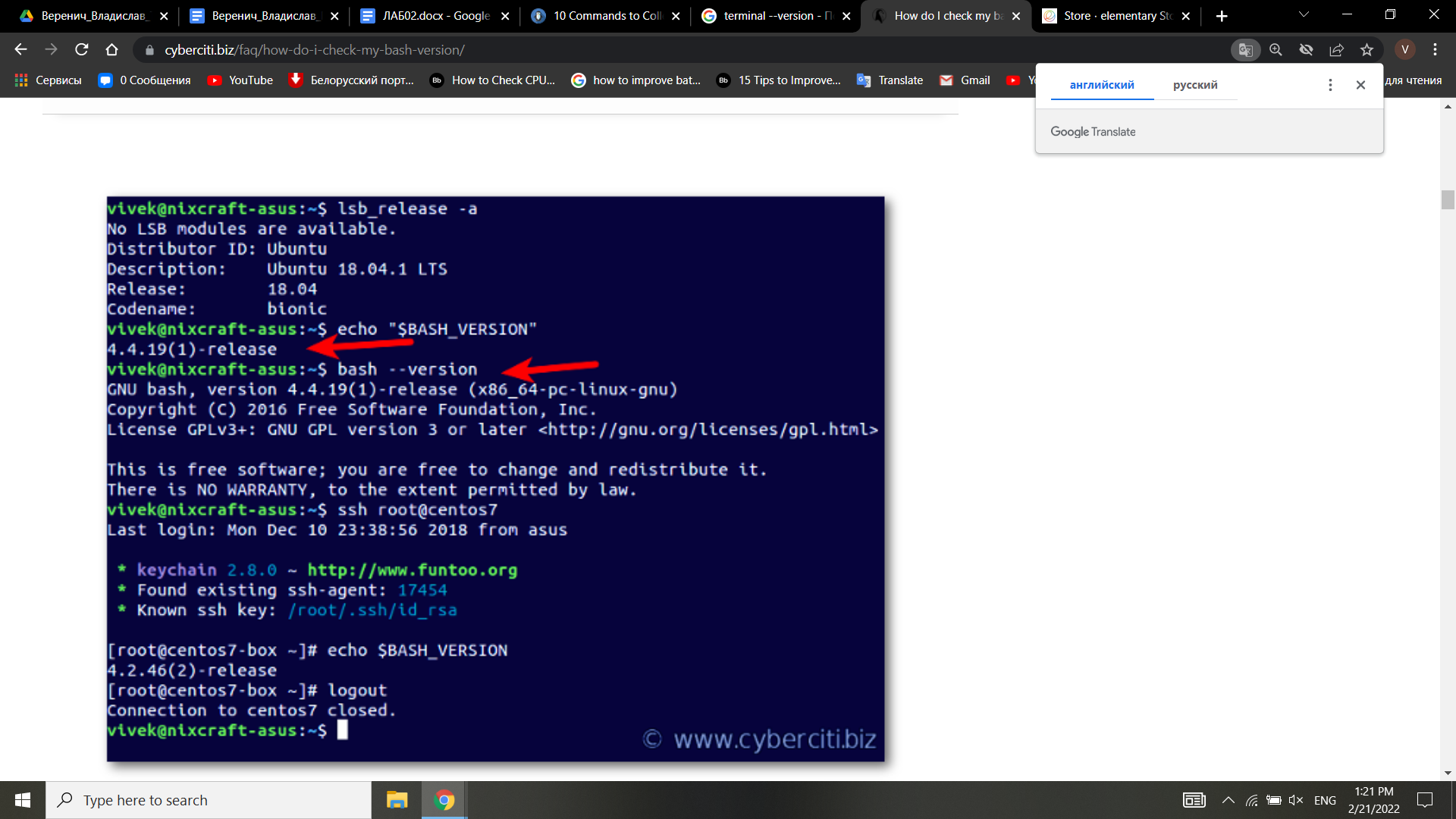
21. В отчёте проиллюстрируйте историю коммитов, работы с ветками.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое «командная оболочка» / «командный интерпретатор»?

Интерпретатор команд, встроенный в Unix системы.

2. Как называется командный интерпретатор и какая его версия в используемой

операционной системе?

3. Для каких задач предназначено приложение Terminal в macOS?

Для доступа к командной оболочке.

4. Приведите полный путь к домашней папке пользователя, под которым вы

авторизованы в системе. Привести пример команды, позволяющей

определить путь к домашнему (текущему) каталогу в консоли.

/home/verchik

5. Какой командой можем вывести содержимое каталога системных конфигурационных файлов?

ls

6. С помощью какой команды, можем изменить командный интерпретатор с zsh

на bash?

chsh -s /bin/bash

7. В каких случаях выполняется инициализация репозитория? Какой командой?

Инициализация выполняется, если не создан локальный репозиторий. Используется команда git init

8. Что такое коммит?

Коммит - это команда, при выполнении которой данные, которые находятся в index'е переносятся в репозиторий и устанавливаются соответствующие ссылки на эти данные

9. Почему перед созданием новой ветки важно создать коммит?

У меня не было возможности создать новую ветку от текущей(к примеру, main) если присутствуют какие-то изменения в файлах в текущей ветке

10. В каких состояниях может находиться файл в репозитории? Как происходит

изменение состояния файла?

Файл может находиться в одном из трёх состояний: modified, staged, commited

11. Что такое ветка?

Ветка - полная копия проекта, изменяя которую вы никак не влияете на сам проект. Веток может быть несколько, их можно сливать

12. Какое условие является обязательным для формирования ветки master

после клонирования репозитория или его инициализации?

git push -u origin master - после инициализации

git clone --branch <branchname> <remote-repo-url> - если ветка уже есть на удаленном репозитории

13. Что такое HEAD?

HEAD - это указатель на текущую ветку

14. Способы создания веток.

git branch [name]  
git checkout -d [name]

15. Как узнать текущую ветку?

Командой git branch

16. Как переключаться между ветками?

Командой git checkout [name]

17. С помощью какой команды можно слить две ветки?

Командой git merge [name]

18. Как подключить к инициализированному (созданному)

репозиторию внешний репозиторий? Какой командой?

git remote add origin git@github.com:username/reponame.git

локальному?

git clone https://github.com/*YOUR-USERNAME*/*YOUR-REPOSITORY*

19. Какая команда позволяет отменить последний коммит?

git reset --soft HEAD~1

20. Для чего предназначена команда rebase?

Rebase служит для перемещения. Можно взять всё, что попало в коммиты на одной ветке, и применить это к другой.

21. В каких случаях требуется создание форка? Приведите способы создания

форка, включая консольную команду.

Создание форка - это клонирование чужого репозитория. Можно осуществить командами:

git clone [github repo]  
cd [github repo]  
git fork