

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

Дисциплина: Архитектура компьютеров и операционные системы

Раздел: Архитектура компьютеров

Студент: Кадров В. М.

Группа: НКАбд-02-22

Студ. билет: № 1132226454

МОСКВА

2022 г.

Содержание

Цель	1
Теоретическое введение	1
Задачи	2
Описание результатов выполнения задания	3
Отчёт о выполнении заданий для самостоятельной работы	9
Выводы	9

Цель

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.

Теоретическое введение

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 разрешения 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

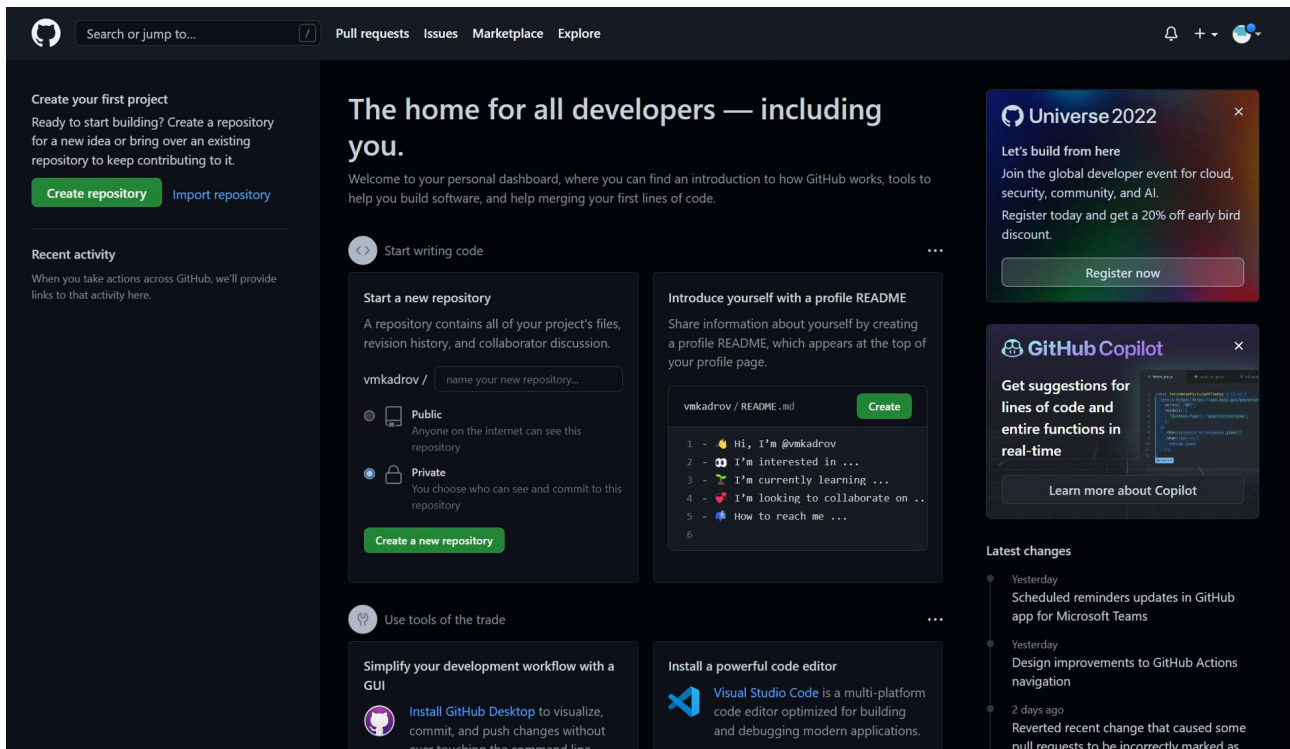
Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями. Благодаря тому, что *Git* является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

Задачи

1. Настройка github
2. Базовая настройка git
3. Создание SSH-ключа
4. Создание репозитория курса на основе шаблона
5. Настройка каталога курса

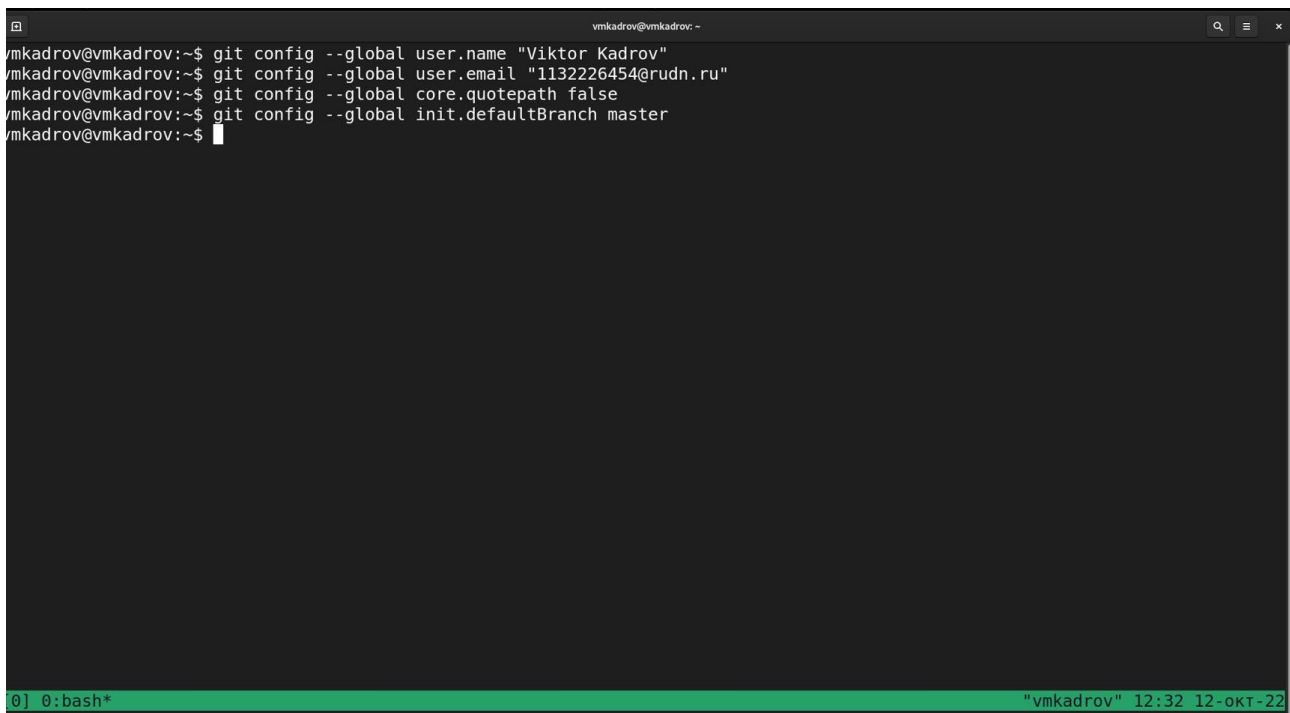
Описание результатов выполнения задания

1. Создаем учетную запись на github.



1.1 Интерфейс после создания учетной записи

2. Выполняем базовую настройку git в linux. Устанавливаем username, email и название начальной ветки, а так же настраиваем отображение и перевод строк.



2.1 Настройка username, email пр.

```
vmkadrov@vmkadrov:~$ git config --global core.autocrlf input
vmkadrov@vmkadrov:~$ git config --global core.safecrlf warn
vmkadrov@vmkadrov:~$

[0] 0: bash* "vmkadrov" 12:34 12-окт-22
```

2.2 Задание параметров переноса строк

3. Генерируем SSH-ключ для доступа к репозиторию, а после цепляем открытую часть к учетной записи Github,

```
vmkadrov@vmkadrov:~$ ssh-keygen -C "Viktor Kadrov 1132226454@rudn.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vmkadrov/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vmkadrov/.ssh/id_rsa
Your public key has been saved in /home/vmkadrov/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:XlpkFLPTPmHc0U+s0hHL5BKxWalzbCYlfG8VN+qcMvc Viktor Kadrov 1132226454@rudn.ru
The key's randomart image is:
+---[RSA 3072]---+
|          +.+++.==+
|          .*=.XoB|
|          *= %.=+
|          +. @ X +
|          S o+ & o
|          . + * .
|          o   . E
|          +-----+
+-----[SHA256]-----+
vmkadrov@vmkadrov:~$ █

[0] 0: bash* "mc [vmkadrov@vmkadrov" 13:02 12-окт-22
```

3.1 Генерация ключа

```
vmkadrov@vmkadrov:~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
vmkadrov@vmkadrov:~$
```

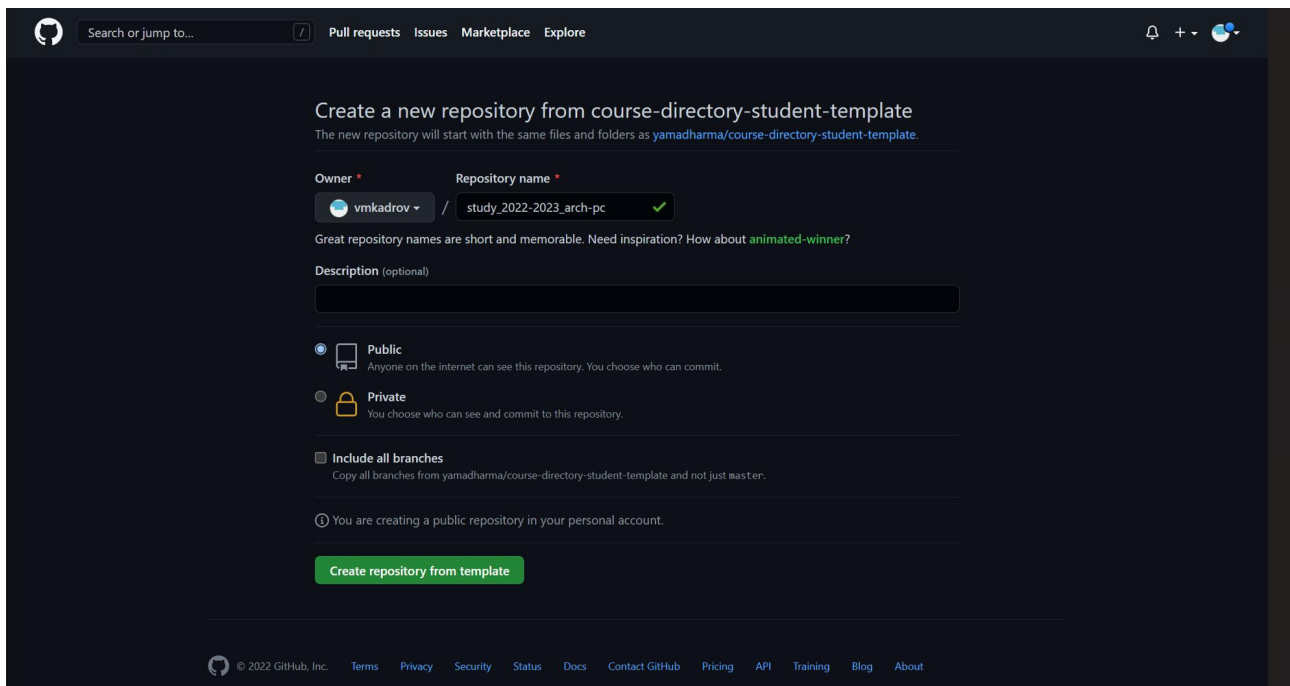
[0] 0: bash* "mc [vmkadrov@vmkadrov" 13:06 12-окт-22

3.2 Копируем ключ в буфер обмена

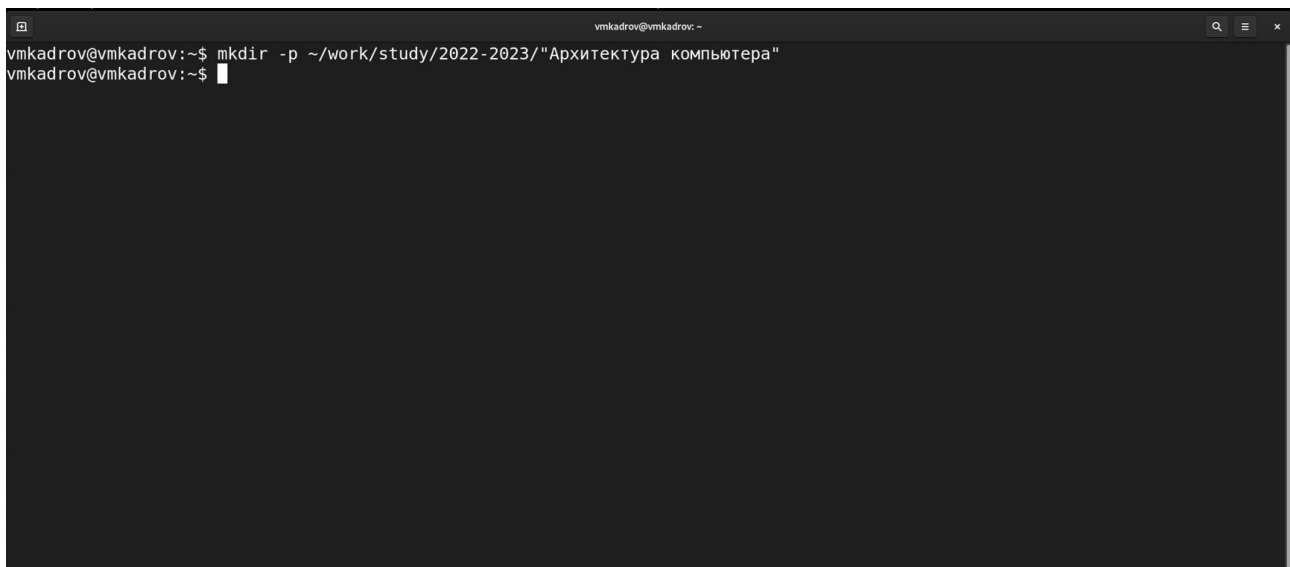
The screenshot shows the GitHub user profile page for 'vmkadrov'. The left sidebar contains navigation links: Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, SSH and GPG keys (highlighted), Organizations, Moderation, Code, planning, and automation, Repositories, Packages, GitHub Copilot, Pages, and Saved replies. The main content area is titled 'SSH keys' and includes a 'New SSH key' button. Below this, it states 'This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.' Under the 'Authentication Keys' section, there is one key listed: 'linux labs' with a SHA256 fingerprint, added on 14 Oct 2022, and marked as 'Never used' with 'Read/write' permissions. A 'Delete' button is next to it. Below the key list, there is a link to a guide on generating SSH keys. The 'GPG keys' section has a 'New GPG key' button and states 'There are no GPG keys associated with your account.' The 'Vigilant mode' section has a checkbox for 'Flag unsigned commits as unverified' and a link to learn about vigilant mode.

3.3 Добавляем новый ключ на github

4. Создаем новый репозиторий на основе шаблона. Затем создаем в домашнем каталоге путь `~/work/study/2022-2023/"Архитектура компьютера"`, куда скачиваем содержимое репозитория.



4.1 Создание репозитория



4.2 Создание иерархии каталогов


```

vmkadrov@vmkadrov: ~/work/study/2022-2023/Архитектура компьютера
vmkadrov@vmkadrov:~/work/study/2022-2023/Архитектура компьютера$ git clone --recursive git@github.com:vmkadrov/study_2022-2023_
arch-pc.git arch-pc
Клонирование в «arch-pc»...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ECDSA key fingerprint is SHA256:p2QAMXNIC1TJYWeI0ttrVc98/R1BUFWu3/LiyKgUfQM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com,140.82.121.4' (ECDSA) to the list of known hosts.
remote: Enumerating objects: 26, done.
remote: Counting objects: 100% (26/26), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 26 (delta 0), reused 17 (delta 0), pack-reused 0
Получение объектов: 100% (26/26), 16.39 KiB | 202.00 KiB/s, готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован п
о пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «te
mplate/report»
Клонирование в «/home/vmkadrov/work/study/2022-2023/Архитектура компьютера/arch-pc/template/presentation»...
remote: Enumerating objects: 71, done.
remote: Counting objects: 100% (71/71), done.
remote: Compressing objects: 100% (49/49), done.
remote: Total 71 (delta 23), reused 68 (delta 20), pack-reused 0
Получение объектов: 100% (71/71), 88.89 KiB | 323.00 KiB/s, готово.
Определение изменений: 100% (23/23), готово.
Клонирование в «/home/vmkadrov/work/study/2022-2023/Архитектура компьютера/arch-pc/template/report»...
remote: Enumerating objects: 78, done.
remote: Counting objects: 100% (78/78), done.
remote: Compressing objects: 100% (52/52), done.
remote: Total 78 (delta 31), reused 69 (delta 22), pack-reused 0
Получение объектов: 100% (78/78), 292.27 KiB | 449.00 KiB/s, готово.
Определение изменений: 100% (31/31), готово.
Подмодуль по пути «template/presentation»: забрано состояние «2703b47423792d472694aaf7555a5626dce51a25»
Подмодуль по пути «template/report»: забрано состояние «df7b2ef80f8def3b9a496f8695277469a1a7842a»
vmkadrov@vmkadrov:~/work/study/2022-2023/Архитектура компьютера$

```

4.3 Клонирование репозитория в созданную папку

5. Переходим в каталог курса, после чего удаляем лишние файлы и создаем необходимые каталоги. Сохраняем изменения и отправляем их в центральный репозиторий.

```

vmkadrov@vmkadrov: ~/work/study/2022-2023/Архитектура компьютера/arch-pc
vmkadrov@vmkadrov:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ rm package.json
vmkadrov@vmkadrov:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
vmkadrov@vmkadrov:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ make
vmkadrov@vmkadrov:~/work/study/2022-2023/Архитектура компьютера/arch-pc$

```

5.1 Удаление файла и создание каталогов

```
vmkadrov@vmkadrov: ~/work/study/2022-2023/Архитектура компьютера/arch-pc
create mode 100644 labs/lab09/report/Makefile
create mode 100644 labs/lab09/report/bib/cite.bib
create mode 100644 labs/lab09/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab09/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab09/report/report.md
create mode 100644 labs/lab10/presentation/Makefile
create mode 100644 labs/lab10/presentation/image/kulyabov.jpg
create mode 100644 labs/lab10/presentation/presentation.md
create mode 100644 labs/lab10/report/Makefile
create mode 100644 labs/lab10/report/bib/cite.bib
create mode 100644 labs/lab10/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab10/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab10/report/report.md
create mode 100644 labs/lab11/presentation/Makefile
create mode 100644 labs/lab11/presentation/image/kulyabov.jpg
create mode 100644 labs/lab11/presentation/presentation.md
create mode 100644 labs/lab11/report/Makefile
create mode 100644 labs/lab11/report/bib/cite.bib
create mode 100644 labs/lab11/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab11/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab11/report/report.md
delete mode 100644 package.json
create mode 100644 prepare
vmkadrov@vmkadrov:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 22, готово.
Подсчет объектов: 100% (22/22), готово.
Сжатие объектов: 100% (16/16), готово.
Запись объектов: 100% (20/20), 310.95 KiB | 1.50 MiB/s, готово.
Total 20 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:vmkadrov/study_2022-2023_arch-pc.git
  97bfe7d..dae2989 master -> master
vmkadrov@vmkadrov:~/work/study/2022-2023/Архитектура компьютера/arch-pc$
```

5.2 Отправка в репозиторий

Отчёт о выполнении заданий для самостоятельной работы

6. Копируем отчеты в созданные каталоги и отправляем в репозиторий.

Выводы

В результате выполнения лабораторной работы были приобретены навыки работы с операционной системой на уровне командной строки (организация файловой системы, навигация по файловой системе, создание и удаление файлов и директорий)