

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

Дисциплина: Архитектура компьютера

Студент: Четвергова Мария Викторовна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Создание SSH ключа	10
4.2	Создание рабочего пространства и репозитория курса на основе шаблона	10
4.3	Создание репозитория курса на основе шаблона	11
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Создание SSH ключа	10
4.2	каталог для предмета «Архитектура компьютера»	11
4.3	Создание репозитория курса	11
4.4	страница репозитория с шаблоном курса	11
4.5	Настройка каталога курса	12
4.6	Создание отчетов по выполнению лабораторной работы	12
4.7	отчеты по выполнению предыдущих лабораторных работ	13

Список таблиц

1 Цель работы

Целью работы является изучение идеологии и применение средств контроля версий. Важной составляющей лабораторной работы № 2 является приобретение практических навыков по работе с системой git.

2 Задание

Приобрести теоретические навыки работы с системой git. Практически закрепить знания работы и применить на практике полученные знания об идеологии и применении средств контроля версий. Усвоить знания, приобретенные в ходе выполнения лабораторной работы № 2. И получить необходимые навыки для работы с системой git.

3 Теоретическое введение

Системы контроля версий. Общие понятия Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удаленном репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведенные 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 определенных 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 измененных файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объем 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

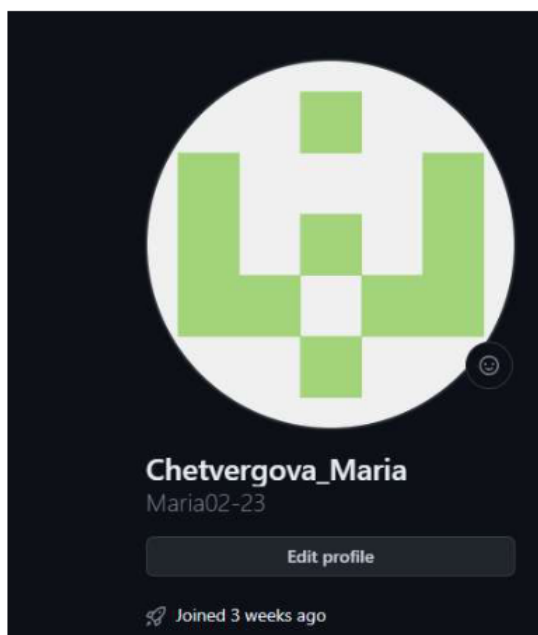
заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с 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 7 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 распределенных системах контроля версий центральный репозиторий не является обязательным. Среди классических VCS наиболее известны CVS, Subversion, а среди распределенных — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

2. Система контроля версий Git 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 распределенной системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

3. Техническое обеспечение Лабораторная работа подразумевает выполнение настройки и работы с системой контроля версий Git (<https://git-scm.com/>). Выполнение работы возможно как в дисплейном классе факультета физико-математических и естественных наук РУДН, так и дома. Описание выполнения работы приведено для дисплейного класса со следующими характеристиками техники: – Intel Core i3-550 3.2 GHz, 4 GB оперативной памяти, 8 GB свободного места на жестком диске; – ОС Linux Gentoo (<http://www.gentoo.ru/>)

4 Выполнение лабораторной работы

##Настройка github Существует несколько доступных серверов репозиториев с возможностью бесплатного размещения данных. Например, <http://bitbucket.org/>, <https://github.com/> и <https://gitflic.ru>. Для выполнения лабораторных работ предлагается использовать Github. Создадим учётную запись на сайте <https://github.com/> и заполним основные данные: ## Базовая настройка git (рис.



??).

рис.1 Учётная запись git.hub

Сначала сделаем предварительную конфигурацию git. Откроем терминал и введём следующие команды, указав имя и email владельца репоз

4.1 Создание SSH ключа

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый). Далее необходимо загрузить сгенерированный открытый ключ. Для этого зайти на сайт github под своей учетной записью и перейти в меню Setting. После этого выбрать в боковом меню SSH and GPG keys и нажать кнопку New SSH key. Скопировав из локальной консоли ключ в буфер, вставляем ключ в появившееся на сайте поле и указываем для ключа имя (Title) (рис. 4.1).

```
nvchetvergova@dk8n59 ~$ git config --global user.name "Murilo2-23"
nvchetvergova@dk8n59 ~$ git config --global user.email "1132732666@p.ur.ru"
nvchetvergova@dk8n59 ~$ git config --global core.quotepath false
nvchetvergova@dk8n59 ~$ git config --global init.defaultBranch master

nvchetvergova@dk8n59 ~$ git config --global core.quotepath false
nvchetvergova@dk8n59 ~$ git config --global init.defaultBranch master

nvchetvergova@dk8n59 ~$ 
nvchetvergova@dk8n59 ~$ git config -- global core.autocrlf input
fatal: not in a git directory
nvchetvergova@dk8n59 ~$ 
nvchetvergova@dk8n59 ~$ git config --global core.autocrlf input
nvchetvergova@dk8n59 ~$ git config --global core.safecrlf warn
```

Рис.2 предварительная конфигурация git

Рис. 4.1: Создание SSH ключа

4.2 Создание рабочего пространства и репозитория курса на основе шаблона

При выполнении лабораторных работ следует придерживаться структуры рабочего пространства. Название проекта на хостинге git имеет вид: study__ Например, для 2023–2024 учебного года и предмета «Архитектура компьютера» (код предмета arch-рс) название проекта примет следующий вид: study_2023–2024_arch-рс. Откроем терминал и создадим каталог для предмета «Архитектура компьютера» (рис. 4.2):

```

march-pc ~$ ssh-keygen -C "Maria02-23 1102202666@pc.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/data/.ssh/ssh-pc.ru/home/m/marchetvergova/.ssh/id_rsa):
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your new public key has been saved in /data/.ssh/ssh-pc.ru/home/m/marchetvergova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:4p9w+KvYUj49M/3M6H/gcXUjz7A Nuc aUM2-23 1102202666@pc.ru
The key's randomart image is:
+-----+
|  .    .+  |
| o  .  + + |
| x  . + . + |
| =  + . + . |
| =D . o o . |
|  .  .  .  |
|  .  .  .  |
|  .  .  .  |
|  .  .  .  |
|  .  .  .  |
+-----+
marchetvergova@kde59 ~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip

```

Рис.3 создание SSH ключа

9

Рис. 4.2: каталог для предмета «Архитектура компьютера»

4.3 Сознание репозитория курса на основе шаблона

Репозиторий на основе шаблона можно создать через web-интерфейс github. Перейд^м на страницу репозитория с шаблоном курса. Далее выбираем Use this template:

```

marchetvergova@kde59 ~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
marchetvergova@kde59 ~$ mkdir -p ~/work/study/2023-2024/Архитектура компьютера
marchetvergova@kde59 ~$ cd ~/work/study/2023-2024/ Архитектура компьютера
marchetvergova@kde59 ~/work/study/2023-2024/Архитектура компьютера $ git clone --recursive git@github.com:Maria02-23/study_2023-2024
Клонирование в «arch-pc»...
ssh: Could not resolve hostname github.com: Device or resource busy
fatal: Не удалось проанять из внешнего репозитория.

```

Рис.4 создание каталога для предмета «Архитектура компьютера»:

Рис. 4.3: Сознание репозитория курса

```

marchetvergova@kde59 ~/work/study/2023-2024/Архитектура компьютера $ git clone --recursive git@github.com:Maria02-23/study_2023-2024
Клонирование в «arch-pc»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
SHA256:4p9w+KvYUj49M/3M6H/gcXUjz7A Nuc aUM2-23 1102202666@pc.ru
This key is not known by any other name.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (SHA256:4p9w+KvYUj49M/3M6H/gcXUjz7A Nuc aUM2-23 1102202666@pc.ru) to the list of known hosts.
remote: Enumerating objects: 77, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 77 (delta 1), reused 11 (delta 0), pack-reused 0

```

Рис. 4.4: страница репозитория с шаблоном курса

В открывшемся окне задайте имя репозитория (Repository name) study_2023–

2024_archpc и создайте репозиторий (кнопка Create repository from template). Откроем терминал и перейдем в каталог курса, клонируем созданный репозиторий:

[клонировать созданный репозиторий](/m/v/mvchetvergova/Изображения/Снимки экрана/Снимок экрана от 2023-10-11 14-47-50.png){#fig:fig1 width=70%}

4.6. Настройка каталога курса Перейдите в каталог курса. Удалите лишние файлы, создайте необходимые каталоги и отправим файлы на сервер:

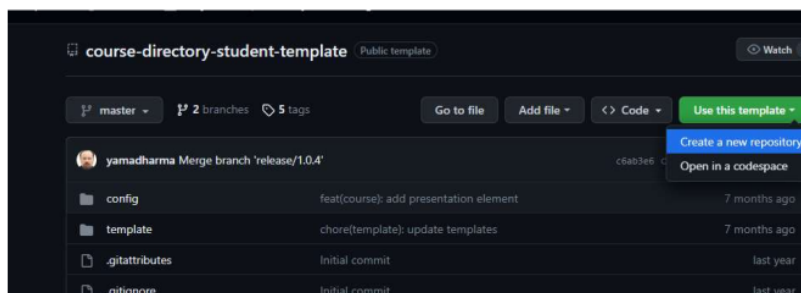


Рис. 4.5: Настройка каталога курса

Проверим правильность создания иерархии рабочего пространства в локальном репозитории и на странице github. Пояснение: командой «`cd ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc`» перейдем в каталог курса и с помощью команды «`rm package.json`» удалим лишние файлы. Затем создадим новые необходимые файлы и добавим файлы на сервер. Затем переместимся на страницу github и проверим правильность создания репозитория. Самостоятельная работа 5.1 Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report):

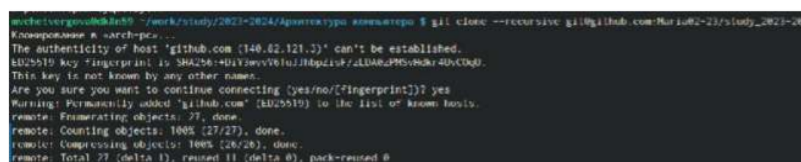


Рис. 4.6: Создание отчетов по выполнению лабораторной работы

Пояснение: В подкаталоге lab02 каталога labs создаем файл report. Данный

5.2 Скопируйте отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.

Рис. 4.7: отчеты по выполнению предыдущих лабораторных работ

5 Выводы

В ходе лабораторной работы №2 мы получили необходимые теоретические знания для работы с системой git, а также закрепили полученные знания на практике. Изучены базовые команды для работы с системой git, изучили идеологию и применение средств контроля версий. Работа выполнена с помощью операционной системы Linux и сайта git hub. Таким образом получены базовые знания, позволяющие работать с системой git и средствами контроля версий.

Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN
6. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
7. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
8. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
9. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
10. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
11. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
12. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
13. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
14. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург — 656 с. — ISBN 978-5-94157-538-1.
15. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-

- е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.
16. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
17. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).