

Отчёт по лабораторной работе №3

Дисциплина: архитектура компьютера

Сибомана Ламек

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Настройка GitHub	7
3.2	Базовая настройка Git	7
3.3	Создание SSH-ключа	8
3.4	Создание рабочего пространства и репозитория курса на основе шаблона	9
3.5	Создание репозитория курса на основе шаблона	10
3.6	Настройка каталога курса	11
3.7	Выполнение заданий для самостоятельной работы	12
4	Выводы	16

Список иллюстраций

3.1	Учётная запись на сайте https://github.com/	7
3.2	Аккаунт GitHub	7
3.3	Предварительная конфигурация git	7
3.4	Настройка кодировки	8
3.5	Создание имени для начальной ветки	8
3.6	Параметр autocrlf	8
3.7	Параметр safecrlf	8
3.8	Генерация SSH-ключа	8
3.9	Установка утилиты xclip	9
3.10	Копирование содержимого файла	9
3.11	Окно SSH and GPG keys	9
3.12	Добавление ключа	9
3.13	Создание рабочего пространства	10
3.14	Страница шаблона для репозитория	10
3.15	Окно создания репозитория	10
3.16	Созданный репозиторий	10
3.17	Перемещение между директориями	10
3.18	Клонирование репозитория	11
3.19	Окно с ссылкой для копирования репозитория	11
3.20	Перемещение между директориями	11
3.21	Удаление файлов	11
3.22	Создание каталогов	11
3.23	Добавление и сохранение изменений на сервере	11
3.24	Выгрузка изменений на сервер	12
3.25	Страница репозитория	12
3.26	Создание файла	12
3.27	Работа с отчетом в текстовом процессоре	12
3.28	Перемещение между директориями	13
3.29	Проверка местонахождения файлов	13
3.30	Копирование файла	13
3.31	Перемещение между директориями	13
3.32	Копирование файла	13
3.33	Добавление файла на сервер	13
3.34	Перемещение между директориями	14
3.35	Добавление файла на сервер	14
3.36	Подкаталоги и файлы в репозитории	14
3.37	Отправка в центральный репозиторий сохраненных изменений	14

3.38	Страница каталога в репозитории	14
3.39	Страница последних изменений в репозитории	15
3.40	Каталог lab01/report	15
3.41	Каталог lab02/report	15
3.42	Каталог lab03/report	15

1 Цель работы

Целью данной работы является изучить идеологию и применение средств контроля версий, а также приобрести практические навыки по работе с системой git.

2 Задание

1. Техническое обеспечение

1. Настройка GitHub.
2. Базовая настройка Git.
3. Создание SSH-ключа.
4. Создание рабочего пространства и репозитория курса на основе шаблона.
5. Создание репозитория курса на основе шаблона.
6. Настройка каталога курса.

2. Выполнение заданий для самостоятельной работы.

3 Выполнение лабораторной работы

3.1 Настройка GitHub

Для начала создадим учётную запись на сайте <https://github.com/> и заполните основные данные (рис. 3.1)

Учётная запись на сайте <https://github.com/>

Рис. 3.1: Учётная запись на сайте <https://github.com/>

Аккаунт создан (рис. 3.2).

Аккаунт GitHub

Рис. 3.2: Аккаунт GitHub

3.2 Базовая настройка Git

Открываю виртуальную машину, затем открываю терминал и делаю предварительную конфигурацию git. Ввожу команду `git config --global user.name ""`, указывая свое имя и команду `git config --global user.email "work@mail"`, указывая в ней электронную почту владельца, то есть мою (рис. 3.3).

Предварительная конфигурация git

Рис. 3.3: Предварительная конфигурация git

Настраиваю utf-8 в выводе сообщений git для корректного отображения символов (рис. 3.4).

Настройка кодировки

Рис. 3.4: Настройка кодировки

Задаю имя «master» для начальной ветки (рис. 3.5).

Создание имени для начальной ветки

Рис. 3.5: Создание имени для начальной ветки

Задаю параметр `autocrlf` со значением `input`, так как я работаю в системе Linux, чтобы конвертировать CRLF в LF только при коммитах (рис. 3.6). CR и LF – это символы, которые можно использовать для обозначения разрыва строки в текстовых файлах.

Параметр `autocrlf`

Рис. 3.6: Параметр `autocrlf`

Задаю параметр `safecrlf` со значением `warn`, так Git будет проверять преобразование на обратимость (рис. 3.7). При значении `warn` Git только выведет предупреждение, но будет принимать необратимые конвертации.

Параметр `safecrlf`

Рис. 3.7: Параметр `safecrlf`

3.3 Создание SSH-ключа

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый). Для этого ввожу команду `ssh-keygen -C "Имя Фамилия, work@email"`, указывая имя владельца и электронную почту владельца (рис. 3.8). Ключ автоматически сохранится в каталоге `~/.ssh/`.

Генерация SSH-ключа

Рис. 3.8: Генерация SSH-ключа

Xclip – утилита, позволяющая скопировать любой текст через терминал. Оказывается, в дистрибутиве Linux Kali ее сначала надо установить. Устанавливаю xclip с помощью команды apt-get install с ключом -у от имени суперпользователя, введя в начале команды sudo (рис. 3.9).

Установка утилиты xclip

Рис. 3.9: Установка утилиты xclip

Копирую открытый ключ из директории, в которой он был сохранен, с помощью утилиты xclip (рис. 3.10).

Копирование содержимого файла

Рис. 3.10: Копирование содержимого файла

Открываю браузер, захожу на сайт GitHub. Открываю свой профиль и выбираю страницу «SSH and GPG keys». Нажимаю кнопку «New SSH key» (рис. 3.11).

Окно SSH and GPG keys

Рис. 3.11: Окно SSH and GPG keys

Вставляю скопированный ключ в поле «Key». В поле Title указываю имя для ключа. Нажимаю «Add SSH-key», чтобы завершить добавление ключа (рис. 3.12).

Добавление ключа

Рис. 3.12: Добавление ключа

3.4 Создание рабочего пространства и репозитория курса на основе шаблона

Закрываю браузер, открываю терминал. Создаю директорию, рабочее пространство, с помощью утилиты mkdir, благодаря ключу -p создаю все директории после домашней ~/work/study/2022-2023/“Архитектура компьютера”

рекурсивно. Далее проверяю с помощью `ls`, действительно ли были созданы необходимые мне каталоги (рис. 3.13).

Создание рабочего пространства

Рис. 3.13: Создание рабочего пространства

3.5 Создание репозитория курса на основе шаблона

В браузере перехожу на страницу репозитория с шаблоном курса по адресу <https://github.com/yamadharma/course-directory-student-template>. Далее выбираю «Use this template», чтобы использовать этот шаблон для своего репозитория (рис. 3.14).

Страница шаблона для репозитория

Рис. 3.14: Страница шаблона для репозитория

В открывшемся окне задаю имя репозитория (Repository name): `study_2022–2023_arh-рс` и создаю репозиторий, нажимаю на кнопку «Create repository from template» (рис. 3.15).

Окно создания репозитория

Рис. 3.15: Окно создания репозитория

Репозиторий создан (рис. 3.16).

Созданный репозиторий

Рис. 3.16: Созданный репозиторий

Через терминал перехожу в созданный каталог курса с помощью утилиты `cd` (рис. 3.17).

Перемещение между директориями

Рис. 3.17: Перемещение между директориями

Клонирую созданный репозиторий с помощью команды `git clone –recursive git@github.com:/study_2022–2023_arh-pc.git` arch-pc (рис. 3.18).

Клонирование репозитория

Рис. 3.18: Клонирование репозитория

Копирую ссылку для клонирования на странице созданного репозитория, сначала перейдя в окно «code», далее выбрав в окне вкладку «SSH» (рис. 3.19).

Окно с ссылкой для копирования репозитория

Рис. 3.19: Окно с ссылкой для копирования репозитория

3.6 Настройка каталога курса

Перехожу в каталог arch-pc с помощью утилиты `cd` (рис. 3.20).

Перемещение между директориями

Рис. 3.20: Перемещение между директориями

Удаляю лишние файлы с помощью утилиты `rm` (рис. 3.21).

Удаление файлов

Рис. 3.21: Удаление файлов

Создаю необходимые каталоги (рис. 3.22).

Создание каталогов

Рис. 3.22: Создание каталогов

Отправляю созданные каталоги с локального репозитория на сервер: добавляю все созданные каталоги с помощью `git add`, комментирую и сохраняю изменения на сервере как добавление курса с помощью `git commit` (рис. 3.23).

Добавление и сохранение изменений на сервере

Рис. 3.23: Добавление и сохранение изменений на сервере

Отправляю все на сервер с помощью push (рис. 3.24).

Выгрузка изменений на сервер

Рис. 3.24: Выгрузка изменений на сервер

Проверяю правильность выполнения работы сначала на самом сайте GitHub (рис. 3.25).

Страница репозитория

Рис. 3.25: Страница репозитория

3.7 Выполнение заданий для самостоятельной работы

1. Перехожу в директорию labs/lab03/report с помощью утилиты cd. Создаю в каталоге файл для отчета по третьей лабораторной работе с помощью утилиты touch (рис. 3.26).

Создание файла

Рис. 3.26: Создание файла

Оформить отчет я смогу в текстовом процессоре LibreOffice Writer, найдя его в меню приложений (рис. 3.7).

Меню приложений+

После открытия текстового процессора открываю в нем созданный файл и могу начать в нем работу над отчетом (рис. 3.27).

Работа с отчетом в текстовом процессоре

Рис. 3.27: Работа с отчетом в текстовом процессоре

2. Перехожу из подкаталога lab03/report в подкаталог lab01/report с помощью утилиты cd (рис. 3.28).

Перемещение между директориями

Рис. 3.28: Перемещение между директориями

Проверяю местонахождение файлов с отчетами по первой и второй лабораторным работам. Они должны быть в подкаталоге домашней директории «Загрузки», для проверки использую команду `ls` (рис. 3.29).

Проверка местонахождения файлов

Рис. 3.29: Проверка местонахождения файлов

Копирую первую лабораторную с помощью утилиты `cp` и проверяю правильность выполнения команды `cp` с помощью `ls` (рис. 3.30).

Копирование файла

Рис. 3.30: Копирование файла

Перехожу из подкаталога `lab01/report` в подкаталог `lab02/report` с помощью утилиты `cd` (рис. 3.31).

Перемещение между директориями

Рис. 3.31: Перемещение между директориями

Копирую вторую лабораторную с помощью утилиты `cp` и проверяю правильность выполнения команды `cp` с помощью `ls` (рис. 3.32).

Копирование файла

Рис. 3.32: Копирование файла

3. Добавляю с помощью команды `git add` в коммит созданные файлы: Л02_Дворкина отчет (рис. 3.33). и

Добавление файла на сервер

Рис. 3.33: Добавление файла на сервер

Перехожу в директорию, в которой находится отчет по первой лабораторной работе с помощью `cd` (рис. 3.34).

Перемещение между директориями

Рис. 3.34: Перемещение между директориями

Добавляю файл Л01_Дворкина_отчет (рис. 3.35).

Добавление файла на сервер

Рис. 3.35: Добавление файла на сервер

Сохраняю изменения на сервере командой `git commit -m “...”`, поясняя, что добавила файлы.

То же самое делаю для отчета по третьей лабораторной работе: перехожу в директорию `labs/lab03/report` с помощью `cd`, добавляю с помощью `git add` нужный файл, сохраняю изменения с помощью `git commit` (рис. 3.36).

Подкаталоги и файлы в репозитории

Рис. 3.36: Подкаталоги и файлы в репозитории

Отправляю в центральный репозиторий сохраненные изменения командой `git push -f origin master` (рис. 3.37).

Отправка в центральный репозиторий сохраненных изменений

Рис. 3.37: Отправка в центральный репозиторий сохраненных изменений

Проверяю на сайте GitHub правильность выполнения заданий. Вижу, что пояснение к совершенным действиям отображается (рис. 3.38).

Страница каталога в репозитории

Рис. 3.38: Страница каталога в репозитории

При просмотре изменений так же вижу, что были добавлены файлы с отчетами по лабораторным работам (рис. 3.39).

Страница последних изменений в репозитории

Рис. 3.39: Страница последних изменений в репозитории

Вижу, что отчеты по лабораторным работам находятся в соответствующих каталогах репозитория: отчет по первой - в lab01/report (рис. 3.40), по второй – в lab02/report (рис. 3.41), по третьей в - lab03/report (рис. 3.42).

Каталог lab01/report

Рис. 3.40: Каталог lab01/report

Каталог lab02/report

Рис. 3.41: Каталог lab02/report

Каталог lab03/report

Рис. 3.42: Каталог lab03/report

4 Выводы

При выполнении данной лабораторной работы я изучила идеологию и применение средств контроля версий, а также приобрела практические навыки по работе с системой git.