

Лабораторная работа No2. Система контроля версий Git

Архитектура Компьютера

Сущенко Алина Николаевна

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
4	Выводы	13
	Список литературы	14
5	Задание	15
6	Теоретическое введение	16
7	Выполнение лабораторной работы	17
8	Выводы	18
	Список литературы	19

Список иллюстраций

3.1	Использование команды git	8
3.2	Настраиваем utf-8 в выводе сообщения git	8
3.3	Зададим имя «master» для начальной ветки	8
3.4	Генерация ssh ключа	9
3.5	Копирование кода с помощью «cat»	9
3.6	Ключ SSH на сайте github	9
3.7	Создание каталога	9
3.8	Созданные репозитории на сайте github	10
3.9	Открываем терминал и переходим в каталог «Архитектура компьютера»	10
3.10	Клонирование репозитория	10
3.11	Удаление файлов	10
3.12	Создание каталогов	10
3.13	Перенос файлов на сервер	11
3.14	Проверка иерархии рабочего пространства на github	11
3.15	Проверка иерархии рабочего пространства	11
3.16	Копируем первую лабораторную	11
3.17	Проверка наличия лабораторной	12
3.18	Проверка наличия отчета на github	12
3.19	проверка наличия лабораторной работы	12
7.1	Название рисунка	17

Список таблиц

6.1	Описание некоторых каталогов файловой системы GNU Linux . .	16
-----	---	----

1 Цель работы

Изучить применение средств контроля версии. Приобретения навыков работы с «git».

2 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 зави-

симости 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 Выполнение лабораторной работы

1. С помощью команды `git config --global` вводим в терминал владельца репозитория имя и email

```
nsuthenko@dk6n52 ~ $ git config --global user.name "<Alina Sushchenko>"
nsuthenko@dk6n52 ~ $ git config --global user.email "<1132231439@pfur.ru>"
```

Рис.1.1 Использование команды git

Рис. 3.1: Использование команды git

```
nsuthenko@dk6n52 ~ $ git config --global core.quotepath false
```

Рис.1.2 Настройка utf-8

Рис. 3.2: Настраиваем utf-8 в выводе сообщения git

```
nsuthenko@dk6n52 ~ $ git config --global init.defaultBranch master
nsuthenko@dk6n52 ~ $ git config --global core.autocrlf input
nsuthenko@dk6n52 ~ $ git config --global core.safecrlf warn
```

Рис. 3.3: Зададим имя «master» для начальной ветки

2. Используя команду `ssh-keygen -C`, генерируем ключи

Рис. 3.4: Генерация ssh ключа

Рис. 3.5: Копирование кода с помощью «cat»



Рис. 3.7: Создание каталога

5. Переходим по ссылке указанной в работе и создаем репозитории на основе шаблона, выбрав «Use this template». В открывшемся окне задаем имя репозиторию (study_2023-2024_archpc) и создаем репозитории и нажимаем «Create repository from template».

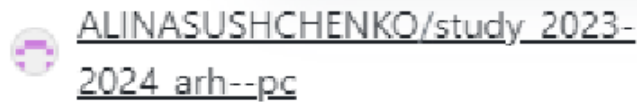


Рис. 3.8: Созданный репозиторий на сайте github

```
ansuthenko@dk4n71 ~ $ cd ~/work/study/2023-2024/"архитектура компьютера"
```

Рис. 3.9: Открываем терминал и переходим в каталог «Архитектура компьютера»

```
ansuthenko@dk4n71 ~ $ cd ~/work/study/2023-2024/"архитектура компьютера"
ansuthenko@dk4n71 ~/work/study/2023-2024/архитектура компьютера $ git clone --recursive git@github.com:ALINASUSHCHENKO/s.git arch-pc
Клонирование в «arch-pc»...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TujHbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? █
```

Рис. 3.10: Клонирование репозитория

6. Удаляем лишние файлы из каталога

```
ansuthenko@dk4n71 ~/work/study/2023-2024/архитектура компьютера/arch-pc $ echo arch-pc > COURSE
ansuthenko@dk4n71 ~/work/study/2023-2024/архитектура компьютера/arch-pc $ make
```

Рис. 3.11: Удаление файлов

```
ansuthenko@dk4n71 ~/work/study/2023-2024/архитектура компьютера $ cd ~/work/study/2023-2024/"архитектура компьютера"/arch-pc
ansuthenko@dk4n71 ~/work/study/2023-2024/архитектура компьютера/arch-pc $ rm package.json
ansuthenko@dk4n71 ~/work/study/2023-2024/архитектура компьютера/arch-pc $ echo arch-pc > COURSE
ansuthenko@dk4n71 ~/work/study/2023-2024/архитектура компьютера/arch-pc $ make
ansuthenko@dk4n71 ~/work/study/2023-2024/архитектура компьютера/arch-pc $ git commit -am 'geat(main):make course structure'
[master 7e13dd7] geat(main):make course structure
2 files changed, 1 insertion(+), 14 deletions(-)
delete mode 100644 package.json
ansuthenko@dk4n71 ~/work/study/2023-2024/архитектура компьютера/arch-pc $ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 289 байтов | 289.00 Киб/с, готово.
Всего 3 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:ALINASUSHCHENKO/study_2023-2024_arch--pc.git
1a592a8..7e13dd7 master -> master
```

Рис. 3.12: Создание каталогов

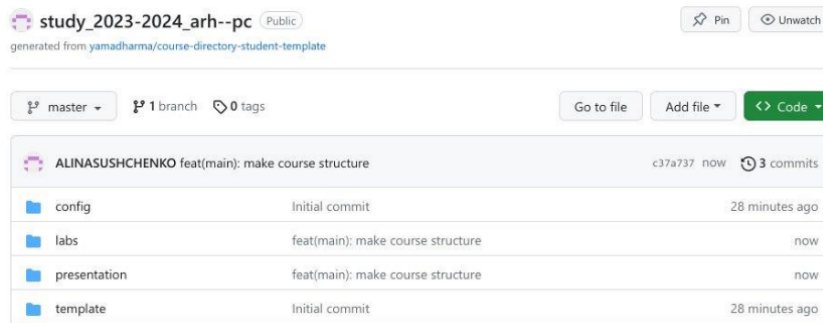


Рис. 3.13: Перенос файлов на сервер

Name	Last commit message
..	
lab01	feat(main): make course structure
lab02	feat(main): make course structure
lab03	feat(main): make course structure
lab04	feat(main): make course structure
lab05	feat(main): make course structure
lab06	feat(main): make course structure
lab07	feat(main): make course structure
lab08	feat(main): make course structure
lab09	feat(main): make course structure
lab10	feat(main): make course structure
lab11	feat(main): make course structure
README.md	feat(main): make course structure
README.ru.md	feat(main): make course structure

Рис. 3.14: Проверка иерархии рабочего пространства на github

```
ansuthenko@dk4n71 ~ $ cp Загрузки/П01.pdf work/study/2023-2024/'архитектура компьютера'/arch-pc/labs/lab01/report
```

Рис. 3.15: Проверка иерархии рабочего пространства

7. Копируем первую лабораторную из каталога «Загрузки» в каталог «labs/lab01/report»

```
ansuthenko@dk4n71 ~ $ cp Загрузки/П01.pdf work/study/2023-2024/'архитектура компьютера'/arch-pc/labs/lab01/report
ansuthenko@dk4n71 ~ $ cd ~/work/study/2023-2024/'архитектура компьютера'/arch-pc/labs/lab01/report
ansuthenko@dk4n71 ~/work/study/2023-2024/'архитектура компьютера'/arch-pc/labs/lab01/report $ ls
bib image Makefile pandoc report.md П01.pdf
ansuthenko@dk4n71 ~/work/study/2023-2024/'архитектура компьютера'/arch-pc/labs/lab01/report $
```

Рис. 3.16: Копируем первую лабораторную

```
ansuthenko@dk4n71 ~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab01/report $ git commit -am 'add lab01'
[master e0a1909] add lab01
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/LO1.pdf
ansuthenko@dk4n71 ~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab01/report $ git push
Переисчисление объектов: 10, готово.
Подсчет объектов: 100% (10/10), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 391.05 Киб | 3.13 Миб/с, готово.
Всего 6 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:ALINASUSHCHENKO/study_2023-2024_arh--pc.git
 c37a737..e0a1909 master -> master
ansuthenko@dk4n71 ~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab01/report $
```

Рис. 3.17: Проверка наличия лабораторной

ALINASUSHCHENKO add lab01

Name	Last commit message
..	
bib	feat(main): make course structure
image	feat(main): make course structure
pandoc	feat(main): make course structure
Makefile	feat(main): make course structure
report.mdi	feat(main): make course structure
LO1.pdf	add lab01

Рис. 3.18: Проверка наличия отчета на github

проверка наличия лабораторной работы

Рис. 3.19: проверка наличия лабораторной работы

4 Выводы

Мы приобрели практические навыки по работе с системой git, а так же изучили применение средств контроля версии.

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.

5 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

6 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы. Например, в табл. 6.1 приведено краткое описание стандартных каталогов Unix.

Таблица 6.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно об Unix см. в [1–6].

7 Выполнение лабораторной работы

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 7.1).

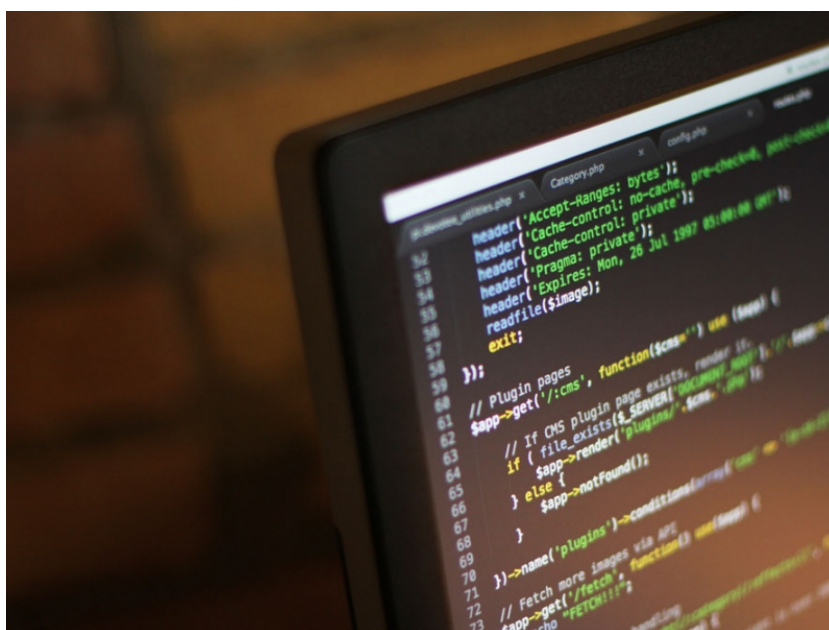


Рис. 7.1: Название рисунка

8 Выводы

Здесь кратко описываются итоги проделанной работы.

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.