

Лабораторная работа номер 2

Система контроля версий Git

Хагуров Андрей андреевич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	11
	Список литературы	12

Список иллюстраций

Список таблиц

1 Цель работы

Изучить систему Git. Научиться работать, загружать информацию на github

2 Задание

Создать SSH ключ и рабочего пространства. Подготовить отчёт по лабораторной работе номер 2.

3 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 заблоки- ровать 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

4 Выполнение лабораторной работы

Сначала сделаем предварительную конфигурацию git. Откроем терминал и введём следующие команды, указав имя и email владельца репозитория (рис. ??).

```
aakhagurov@dk8n59 ~ $ git config --global user.name "Andrei Khagurov"
git config --global user.email "<1132239117aakhagurov@dk8n59 ~ $ git config --global user.email "<1132239117@pfur.ru>"
```

??).

Настроим utf-8 в выводе сообщений git (рис. ??).

```
git config --global core.quotePath false
^[[A^[[aakhagurov@dk8n59 ~ $ git config --global core.quotePath false
```

Дададим имя начальной ветки (будем называть её master) (рис. ??).

```
git config --global init.defaultBranch master
aakhagurov@dk8n59 ~ $ git config --global core.quotePath false
git config --global init.defaultBranch master
```

```
git config --global core.autocrlf input
git config --global core.safecrlf warn
использование: git config [<опции>]

Размещение файла конфигурации
--global      использовать глобальный файл конфигурации
--system      использовать системный файл конфигурации
--local       использовать файл конфигурации репозитория
--worktree    use per-worktree config file
-f, --file <файл> использовать указанный файл конфигурации
--blob <идент-двоичн-объекта> прочитать настройки из указанного двоичного объекта
```

Параметр autocrlf (рис. ??).

Параметр safecrlf (рис. ??).

```
aakhagurov@dk8n59 ~ $ git config --global core.autocrlf input
git config --global core.safecrlf warn
```

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый) (рис. ??).

```
aakhagurov@dk8n59 ~ $ ssh-keygen -C "Andrei Khagurov <1132239117@pfur.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/a/a/aakhagurov/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/a/a/aakhagurov/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/a/a/aakhagurov/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:JWI6mNuyLGdA7absiPdGq9EYEzEt38Y2nkLiwt64zYY Andrei Khagurov <1132239117@pfur.ru>
The key's randomart image is:
+----[RSA 3072]-----+
|  .o                    |
|  .o.                   |
| o+Bo+ . .              |
| .O==*. o               |
| .o==o+ oS              |
| ..+o.o                 |
| o+...o                  |
| oE=+o                   |
| o*.                      |
+----[SHA256]-----+
aakhagurov@dk8n59 ~ $
```

Откроем терминал и создадим каталог для предмета «Архитектура компьютера» (рис. ??).

```
aakhagurov@dk8n59 ~ $ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
aakhagurov@dk8n59 ~ $ cd
```

Откроем терминал и перейдем в каталог курса (рис. ??).

```
aakhagurov@dk8n59 ~ $ cd ~/work/study/2023-2024/Архитектура\ компьютера
aakhagurov@dk8n59 ~/work/study/2023-2024/Архитектура\ компьютера $ cd ~/work/study/2023-2024/Архитектура\ компьютера/
```

Клонируем созданный репозиторий (рис. ??).

```
aakhagurov@dk8n59 ~/work/study/2023-2024/Архитектура\ компьютера $ git clone --recursive git://study_2023-2024.arhpc-...
Клонирование в «study_2023-2024.arhpc-...»...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhpZisF/zLDA0zPMsVhDkr4UyCQOu.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? arch-pc
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
```

Перейдем в каталог курса: (рис. ??).

```
aakhagurov@dk8n59 ~ $ cd ~/work/study/2023-2024/Архитектура\ компьютера/
aakhagurov@dk8n59 ~/work/study/2023-2024/Архитектура\ компьютера $ cd ~/work/study/2023-2024/Архитектура\ компьютера/
```

Удалим лишние файлы (рис. ??). 11

```
aakhagurov@dk8n59 ~/work/study/2023-2024/Архитектура\ компьютера/study_2023-2024.arhpc- $ echo arch-pc
arch-pc
aakhagurov@dk8n59 ~/work/study/2023-2024/Архитектура\ компьютера/study_2023-2024.arhpc- $ make
ls
aakhagurov@dk8n59 ~/work/study/2023-2024/Архитектура\ компьютера/study_2023-2024.arhpc- $ ls
CHANGELOG.md  COURSE  LICENSE  prepare  README.en.md  README.md
cxfig         labs    Makefile  presentation  README.git-flow.md  template
```

Создадим необходимые каталоги (рис. ??).

```
aakhagurov@dk8n59 ~/work/study/2023-2024/Архитектура\ компьютера/study_2023-2024.arhpc- $ git add .
aakhagurov@dk8n59 ~/work/study/2023-2024/Архитектура\ компьютера/study_2023-2024.arhpc- $ git commit -am 'feat(main): make course structure'
[master a16693f] feat(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
```

Отправим файлы на сервер (рис. ??).

```
aakhagurov@dk8n59 ~/work/study/2023-2024/Архитектура\ компьютера/study_2023-2024.arhpc- $ git push
Перечислил объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При скатии изменений используется до 6 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.14 КиБ | 2.63 МБ/с, готово.
Всего 35 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:aakhagurov/study_2023-2024.arhpc-.git
 2dcba72..a16693f  master -> master
```

Отправим файлы на сервер (рис. ??).

Создадим отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (рис. ??).

```
aakhagurov@dk8n59 ~/work/study/2023-2024/Архитектура\ компьютера/study_2023-2024.arhpc-/labs/lab02/report $ touch report.md
```

Скопируем отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства (рис. ??).

```
aakhagurov@dk8n59 ~/work/study/2023-2024/Архитектура\ компьютера/study_2023-2024.arhpc-/labs/lab02/report $ ls
bib  image  Makefile  pandoc  report.docx  report.md  report.pdf  report.pptx
```

5 Выводы

В ходе лабораторной работы мы ознакомились с системой Git и создали собственный репозиторий

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 0596009658. — URL: <http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс,
- 11.
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. — БХВ- Петербург, 2010. — 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, — 1120 с. — (Классика Computer Science).