

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных

наук Кафедра прикладной информатики и теории

вероятностей

ОТЧЕТ

по лабораторной работе № 3. Система контроля версий Git

*Дисциплина: Архитектура
компьютера*

Студент: *Королёв Иван Андреевич*

Группа: *НКАбд-05-22*

МОСКВА 2022г.

Содержание

1. Цель работы:.....	3
2. Теоретическое введение:.....	3
2.1. Системы контроля версий. Общие понятия:.....	3
2.2. Система контроля версий Git:.....	4
3. Выполнение работы:.....	4
3.1. Настройка GitHub:.....	4
3.2. Базовая настройка git:.....	5
3.3. Создание SSH-ключа:.....	5
3.4. Создание рабочего пространства и репозитория курса на основе шаблона:.....	6
3.5. Создание репозитория курса на основе шаблона:.....	7
3.6. Настройка каталога курса:.....	7
4.Выполнение самостоятельной работы:.....	9
5.Вывод:.....	10

1. Цель работы:

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

2. Теоретическое введение:

2.1. 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 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.

2.2. 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 распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

3. Выполнение работы:

3.1. Настройка GitHub:

Создаю учетную запись на GitHub и заполняю основные данные. (Рис.1.1)

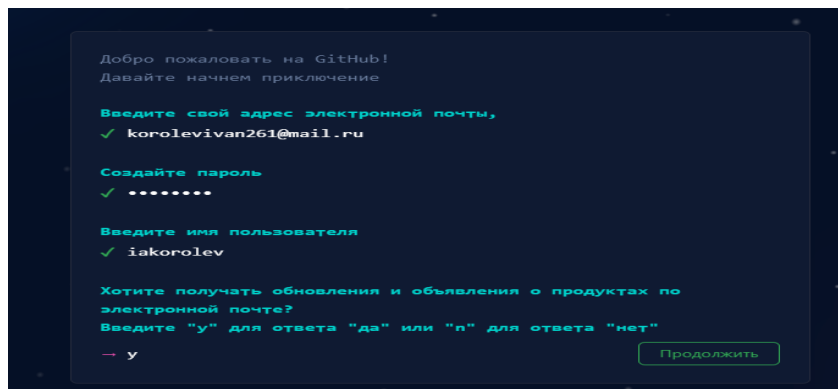


Рис.1.1. Создаю учетную запись на github

3.2. Базовая настройка git:

Сначала сделаю предварительную конфигурацию git. Открою терминал и введу следующие команды, указав имя и email:(Рис.1.2)

```
[iakorolyov@fedora ~]$ git config --global user.name "<iakorolyov>"  
[iakorolyov@fedora ~]$ git config --global user.email "<1032225751@pfur.ru>"
```

Рис.1.2. Указываем имя и email

Настрою utf-8 в выводе сообщений git:(Рис.1.3.)

```
[iakorolyov@fedora ~]$ git config --global core.quotePath false
```

Рис.1.3. Настройка utf-8

Задам имя начальной ветки (master):(Рис.1.4.)

```
[iakorolyov@fedora ~]$ git config --global init.defaultBranch master
```

Рис1.4. Имя начальной ветки

Параметр autocrlf:(Рис.1.5.)

```
[iakorolyov@fedora ~]$ git config --global core.autocrlf input
```

Рис.1.5. Параметр autocrlf

Параметр safecrlf:

```
[iakorolyov@fedora ~]$ git config --global core.safecrlf warn
```

Рис.1.6. Параметр safecrlf

3.3. Создание SSH-ключа:

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый):(Рис1.1.)

```
[iakorolyov@fedora ~]$ ssh-keygen -C "Ivan Korolev <1032225751@pfur.ru>"  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/iakorolyov/.ssh/id_rsa):  
/home/iakorolyov/.ssh/id_rsa already exists.
```

Рис.1.1. Создание ssh-ключа

Ключи сохраняться в каталоге ~/.ssh/.

Далее необходимо загрузить сгенерённый открытый ключ. Для этого захожу на сайт github. Нажимаю на иконку своей учетной записи и перехожу в Settings. Далее нажимаю SSH and GPG key, и нажимаю New Key() (Рис 1.2.)

SSH keys

New SSH key

Рис 1.2. Нажимаю New SSH Key, загружаю сгенерированный ключ

Скопирую из локальной консоли ключ в буфер обмена с помощью команды:(Рис 1.3.)

```
[iakorolyov@fedora ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip  
[iakorolyov@fedora ~]$
```

Рис.1.3. Команда cat

Вставляем ключ в появившееся на сайте поле и указываем для ключа имя(gite).(Рис 1.4.)

SSH keys / Add new

Title

Key type

Authentication Key

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key

Рис1.4. Создание ssh-ключа

3.4. Создание рабочего пространства и репозитория курса на основе шаблона:

Открою терминал и создам каталог для предмета «Архитектура компьютера»:(Рис.1.1)

```
[iakorolyov@fedora ~]$ mkdir -p ~/work/study/2022-2023/"Архитектура компьютера"  
[iakorolyov@fedora ~]$
```

Рис 1.1. Создание каталога

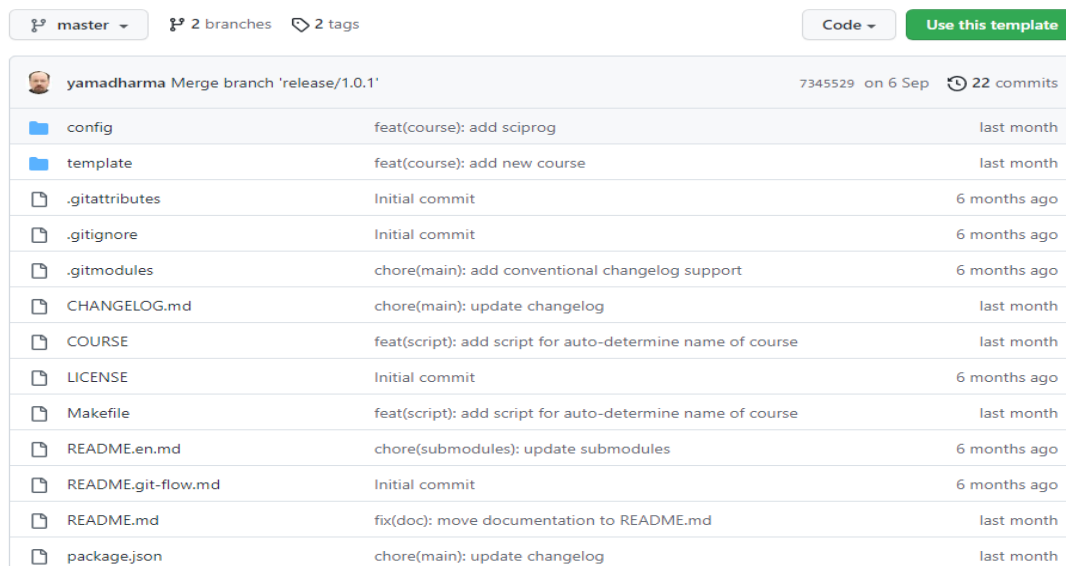
3.5. Создание репозитория курса на основе шаблона:

Репозиторий на основе шаблона можно создать через web-интерфейс github.

Перейду на страницу репозитория с шаблоном курса

<https://github.com/yamadharma/course-directory-student-template>.

Далее выберу Use this template



File/Folder	Commit Message	Commit Date
config	feat(course): add sciprog	last month
template	feat(course): add new course	last month
.gitattributes	Initial commit	6 months ago
.gitignore	Initial commit	6 months ago
.gitmodules	chore(main): add conventional changelog support	6 months ago
CHANGELOG.md	chore(main): update changelog	last month
COURSE	feat(script): add script for auto-determine name of course	last month
LICENSE	Initial commit	6 months ago
Makefile	feat(script): add script for auto-determine name of course	last month
README.en.md	chore(submodules): update submodules	6 months ago
README.git-flow.md	Initial commit	6 months ago
README.md	fix(doc): move documentation to README.md	last month
package.json	chore(main): update changelog	last month

Рис1.1. Создаю репозиторий

В открывшемся окне задаю имя репозитория (Repository name) study_2022–2023_arh-pc и создаю репозиторий (кнопка Create repository from template).

Открою терминал и перейду в каталог:(Рис.1.2.)

```
cd ~/work/study/2022-2023/"Архитектура компьютера"
```

Рис1.2. Команда cd

Клонирую созданный репозиторий:(Рис1.3.)

```
git clone --recursive git@github.com:iakorolev/study_2022-2023_arc-pc.git arch-pc
```

Рис1.3. Клонирую созданный репозиторий

3.6. Настройка каталога курса:

Перейду в каталог курса:(Рис1.4.)

```
[iakorolyov@fedora Архитектура компьютера]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc  
[iakorolyov@fedora arch-pc]$
```

Рис1.4. Перейду в каталог курса

Удалю лишние файлы, создам необходимые каталоги: (Рис 1.5.)

```
[iakorolyov@fedora arch-pc]$ rm package.json
[iakorolyov@fedora arch-pc]$ echo arch-pc > COURSE
[iakorolyov@fedora arch-pc]$ MAKE
bash: MAKE: команда не найдена...
Аналогичная команда: 'make'
[iakorolyov@fedora arch-pc]$ make
[iakorolyov@fedora arch-pc]$ git add .
[iakorolyov@fedora arch-pc]$ git commit -am 'feat(main): make course structure'
[master f183889] feat(main): make course structure
91 files changed, 8229 insertions(+), 14 deletions(-)
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab01/report/report.md
create mode 100644 labs/lab02/presentation/Makefile
create mode 100644 labs/lab02/presentation/image/kulyabov.jpg
```

Рис 1.5. Удаление лишнего и создание необходимых файлов

Отправлю файлы на сервер:

```
[iakorolyov@fedora 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), 310.94 КиБ | 1.18 МиБ/с, готово.
Всего 20 (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:iakorolev/study_2022-2023_arh-pc.git
  12edf76..f183889  master -> master
[iakorolyov@fedora arch-pc]$
```

Рис 1.6. Команда git push

Проверю правильность создания иерархии рабочего пространства в локальном репозитории и на странице github. (Рис 1.7.)

iakorolev / study_2022-2023_arh-pc (Public)

generated from yamadharm/course-directory-student-template

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

Go to file Add file Code

commit	message	hash	date	commits
config	Initial commit		yesterday	
labs	feat(main): make course structure		yesterday	
template	Initial commit		yesterday	

iakorolev feat(main): make course structure f183889 yesterday 2 commits

Рис 1.7. Создал рабочее пространство

4.Выполнение самостоятельной работы:

Загружу все отчеты по выполненным работам в каталоги рабочего пространства(labs>lab01>report), и так для каждой лабораторной работы.(Рис 1,2,3)

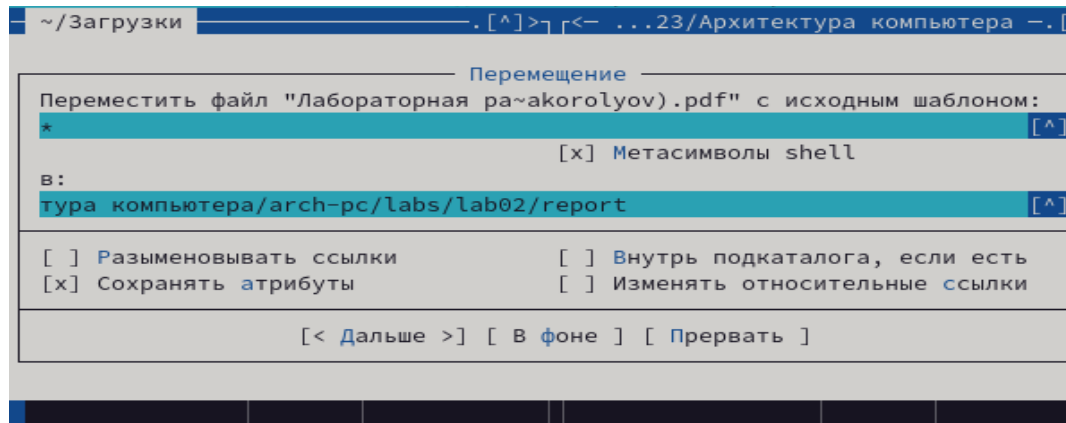


Рис 1. Перещая загруженную лабораторную работу

```
[iakorolyov@fedora arch-pc]$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
Распаковка объектов: 100% (3/3), 740 байтов | 17.00 КиБ/с, готово.
Из github.com:iakorolev/study_2022-2023_arh-pc
7875363..a15df33 master -> origin/master
Обновление 7875363..a15df33
Fast-forward
 labs/lab01/report/Лабораторная работа №1(iakorolyov).docx | Bin 646842 -> 0 bytes
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 labs/lab01/report/Лабораторная работа №1(iakorolyov).docx
[iakorolyov@fedora arch-pc]$ cd labs/lab01/report
[iakorolyov@fedora report]$ git add .
[iakorolyov@fedora report]$ git commit -am 'feat(main):make course structure'
[master 404b0f1] feat(main):make course structure
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 labs/lab01/report/Лабораторная работа №1(iakorolyov).docx
[iakorolyov@fedora report]$ git push
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 528 байтов | 31.00 КиБ/с, готово.
Всего 5 (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:iakorolev/study_2022-2023_arh-pc.git
 a15df33..404b0f1 master -> master
[iakorolyov@fedora report]$ ls
bib image Makefile pandoc report.md 'Лабораторная работа №1(iakorolyov).docx'
```

Рис 2. С помощью команд git pull, add, commit, push, загружаю все лабораторные в репозиторий

master study_2022-2023_arh-pc / labs / lab01 / report /		
iakorolev feat(main):make course structure		
..		
folder	bib	feat(main): make course structure
folder	image	feat(main): make course structure
folder	pandoc/csl	feat(main): make course structure
file	Makefile	feat(main): make course structure
file	report.md	feat(main): make course structure
file	Лабораторная работа №1(iakorolyov).docx	feat(main):make course structure

Рис.3. Лабораторная работа

Такие действия, я проделываю для каждой лабораторной работы, загружаю все на github.

Моя ссылка на репозиторий: https://github.com/iakorolev/study_2022-2023_arh-pc

5.Вывод:

Я изучил идеологию и применение средств контроля версий. Приобрел практические навыки по работе с системой git.