

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

Дисциплина : Архитектура компьютера

Студент : Монхжаргал
Тувшинбаяр Группа :
НКАбд-04-23

МОСКВА
2023 г.

Содержание

1	Цель работы	3
2	Задание	4
3	Теоретическое введение	5
		6
4	Выполнение лабораторной работы	7-16
5	Выводы	17
6	Список литературы	18

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. 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.

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 изменения, сделанные разными участниками, 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. 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. 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. Работа пользователя со своей веткой начинается с проверки и получения изменений из центрального репозитория (при этом в локальное дерево до начала этой процедуры не должно было вноситься изменений). Затем можно вносить изменения в локальном дереве и/или ветке. После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральном репозитории.

4 Выполнение лабораторной работы

Настройка GitHub

Создаю учетную запись на сайте GitHub (рис 1.1)



Рис 1.1: GitHub account

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. (рис 1.2).

```
mtuvshinbayer@mtuvshinbayer-VirtualBox:~$ git config --global user.name "mtuvshinbayer"
mtuvshinbayer@mtuvshinbayer-VirtualBox:~$ git config --global user.email "1032235566@pfur.ru"
```

Рис1.2: Предварительная конфигурация git

Настраиваю utf-8 в выводе сообщений git для корректного отображения символов (рис. 1.3).

```
mtuvshinbayer@mtuvshinbayer-VirtualBox:~$ git config --global core.quotePath false
```

Рис. 1.3: Настройка кодировки

Задаю имя «master» для начальной ветки (рис 1.4)

```
mtuvshinbayer@mtuvshinbayer-VirtualBox:~$ git config --global core.quotePath false
mtuvshinbayer@mtuvshinbayer-VirtualBox:~$ git config --global init.defaultBranch master
```

Рис. 1.4: Создание имени для начальной ветки

Задаю параметр `autocrlf` с значением `input`, так как я работаю в системе Linux, чтобы конвертировать CRLF в LF только при коммитах (рис. 1.5). CR и LF – это символы, которые можно использовать для обозначения разрыва строки в текстовых файлах

```
mtuvshinbayer@mtuvshinbayer-VirtualBox:~$ git config --global core.autocrlf input
```

Рис 1.5

Задаю параметр `safecrlf` со значением `warn`, так Git будет проверять преобразование на обратимость (рис. 1.6). При значении `warn` Git только выведет предупреждение, но будет принимать необратимые конвертации.

```
mtuvshinbayer@mtuvshinbayer-VirtualBox:~$ git config --global core.safecrlf warn
```

Рис 1.6

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. 1.7). Ключ автоматически сохранится в каталоге `~/.ssh/`

```
mtuvshinbayer@mtuvshinbayer-VirtualBox:~$ ssh-keygen -C "Mtuvshinbayer <1032235566@pfur.ru>"
\Generating public/private rsa key pair.
Enter file in which to save the key (/home/mtuvshinbayer/.ssh/id_rsa): documents
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in \documents
Your public key has been saved in \documents.pub
The key fingerprint is:
SHA256:zHRY4w3BXZsmCHb5GwfdXRPem29qhPscYniyPHC/Lt4 Mtuvshinbayer <1032235566@pfur.ru>
The key's randomart image is:
+---[RSA 3072]---+
|      o.*= o.+o|
|      . *o== ooo|
|      o +.oo+..|
|      + . oo. o|
|      S      = o|
|      . ..o . .|
|      oo.+o. o|
|      .o*oo.+ |
```

Рис 1.7

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 -y от имени суперпользователя, введя в начале команды sudo (рис. 1.8).

```
mtuvshinbayer@mtuvshinbayer-VirtualBox:~$ sudo apt-get install -y xclip
[sudo] password for mtuvshinbayer:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libreoffice-ogltrans
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  xclip
0 upgraded, 1 newly installed, 0 to remove and 123 not upgraded.
Need to get 18,3 kB of archives.
After this operation, 60,4 kB of additional disk space will be used.
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy/universe amd64 xclip amd64 0.13-2 [18,3 kB]
Fetched 18,3 kB in 0s (84,6 kB/s)
Selecting previously unselected package xclip.
(Reading database ... 200880 files and directories currently installed.)
Preparing to unpack .../xclip_0.13-2_amd64.deb ...
Unpacking xclip (0.13-2) ...
Setting up xclip (0.13-2) ...
Processing triggers for man-db (2.10.2-1) ...
```

Рис. 1.8: Установка утилиты xclip

Копирую открытый ключ из директории, в которой он был сохранен, с помощью утилиты xclip (рис. 1.9).

```
mtuvshinbayer@mtuvshinbayer-VirtualBox:~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

Рис 1.9 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. 1.10).

SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

 SSH	mtuvshinbayar0407 SHA256: oN0Q6KM47ZHRKr7biByeTRgPCHV1fnoGGQyf8rAY1h0 Added on Feb 5, 2024 Last used within the last week — Read/write	Delete
 SSH	Tuvshinbayar SHA256: Muy8cX1+mgR+ae6MX7XRTxKYtX4kt5S856iKBnnhAGE Added on Feb 6, 2024 Last used within the last week — Read/write	Delete

Рис. 1.10: Окно SSH

Вставляю скопированный ключ в поле «Key». В поле Title указываю имя для ключа. (рис 1.11)

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys


 SSH	mtuvshinbayar0407 SHA256:0N0Q6KM47ZHRKr7biByeTRgPCHV1fnoGGQyf8rAY1h0 Added on Feb 5, 2024 Last used within the last week — Read/write	Delete
 SSH	Tuvshinbayar SHA256:Muy8cX1+mgR+ae6MX7XRTxKYtX4kt5S856iKBnnhAGE Added on Feb 6, 2024 Last used within the last week — Read/write	Delete

Рис. 1.11: Добавление ключа

Создание рабочего пространства и репозитория курса на основе шаблона

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 каталоги (рис. 1.12).



```
mtuvshinbayer@mtuvshinbayer-VirtualBox:~$ mkdir -p work/study/2023~2024/"Computer Architect
mtuvshinbayer@mtuvshinbayer-VirtualBox:~$ ls
Desktop      '\documents.pub'  parantdir3      parentdir3      Public          test           Videos
'\documents' Downloads          parentdir1      perentdir       snap           tmp            work
Documents    Music              parentdir2      Pictures         Templates       tR
```

Рис. 1.12: Создание рабочего пространства

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

yamadharmacourse-directory-student-template

Type to search

>

+

<>Code

Issues

Pull requests

Actions

Projects

Security

Insights

course-directory-student-template

Public template

Watch2

Fork24

Star2

Use this template

master

2 Branches

7 Tags

Go to file

t

+

<>Code

yamadharmamerge branch 'release/1.0.6'

ec12a6a · 3 months ago

44 Commits

config	feat(course): add computer-practice	3 months ago
template	chore(main): update submodules	3 months ago
.gitattributes	Initial commit	2 years ago
.gitignore	Initial commit	2 years ago
.gitmodules	chore(main): add conventional changelog support	2 years ago
CHANGELOG.md	chore(main): update changelog	3 months ago
COURSE	feat(script): add script for auto-determine name of co...	2 years ago
LICENSE	Initial commit	2 years ago
Makefile	feat(courses): add net-os-admin course	4 months ago
	chore(main): update submodules	2 years ago

About

Course Catalog Template for Student

Readme

CC-BY-4.0 license

Activity

2 stars

2 watching

24 forks

Report repository

Releases6

v1.0.6Latest

on Nov 5, 2023

+ 5 releases

Languages

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

Рис 1.13

В открывшемся окне задаю имя репозитория (Repository name): study_2022–2023_arh-pc и создаю репозиторий, нажимаю на кнопку «Create repository» (рис. 1.14).

The screenshot shows the GitHub 'Create repository' form. At the top, the 'Owner' is set to 'mtuvshinbayar' and the 'Repository name' is 'study_2023-2024_arch-pc'. A green checkmark indicates the name is available. Below this, a tip suggests great repository names are short and memorable, with an example 'super-duper-octo-lamp'. A 'Description (optional)' text box is present. Under the 'Visibility' section, 'Public' is selected with a radio button, and 'Private' is unselected. A note at the bottom states 'You are creating a public repository in your personal account.' A green 'Create repository' button is at the bottom right.

Owner * / Repository name *

mtuvshinbayar / study_2023-2024_arch-pc

✓ study_2023-2024_arch-pc is available.

Great repository names are short and memorable. Need inspiration? How about **super-duper-octo-lamp** ?

Description (optional)


☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

(i) You are creating a public repository in your personal account.

Create repository

Рис 1.14
Репозиторий создан (рис.1.15)(забыла сделать скриншот)


study_2023-2024_arch-pc
Public

generated from [yamadharm/course-directory-student-template](#)

Pin
Unwatch 1

master
1 Branch
0 Tags

+
Code

About

No de
provid

Re











CC

Ac

0 s

1 v

0 f

 mtuvshinbayar Initial commit 15bca70 · now 1 Commits		
	config	Initial commit now
	template	Initial commit now
	.gitattributes	Initial commit now
	.gitignore	Initial commit now
	.gitmodules	Initial commit now
	CHANGELOG.md	Initial commit now
	COURSE	Initial commit now
	LICENSE	Initial commit now
	Makefile	Initial commit now

Relea

No relea
[Create a](#)

Packa

No pack

Рис 1.15

Через терминал перехожу в созданный каталог курса с помощью утилиты `cd` (рис. 1.16).

```
mtuvshinbayer@mtuvshinbayer-VirtualBox:~$ cd ~/work/study/2023-2024/'Computer Architecture'  
mtuvshinbayer@mtuvshinbayer-VirtualBox:~/work/study/2023-2024/Computer Architecture$
```

Рис 1.16

Клонирую созданный репозиторий с помощью команды `git clone --recursive git@github.com:/study_2023-2024_arh-pc.git arch-pc` (рис. 1.17).

```
mtuvshinbayer@mtuvshinbayer-VirtualBox:~/work/study/2023-2024/Computer Architecture$ git clone --recursive git@github.com:mtuvshinbayer/study_2023-2024_arh-pc.git arch-pc  
Cloning into 'arch-pc'...  
remote: Enumerating objects: 41, done.  
remote: Counting objects: 100% (41/41), done.  
remote: Compressing objects: 100% (39/39), done.  
remote: Total 41 (delta 3), reused 17 (delta 0), pack-reused 0  
Receiving objects: 100% (41/41), 5.17 MiB | 2.13 MiB/s, done.  
Resolving deltas: 100% (3/3), done.  
Submodule 'template/presentation' (https://github.com/yamadharm/academic-presentation-markdown-template.git) registered for path 'template/presentation'  
Submodule 'template/report' (https://github.com/yamadharm/academic-laboratory-report-template.git) registered for path 'template/report'
```

Рис 1.17

Копирую ссылку для клонирования на странице созданного репозитория, сначала перейдя в окно «code», далее выбрав в окне вкладку «SSH» (рис. 1.18).

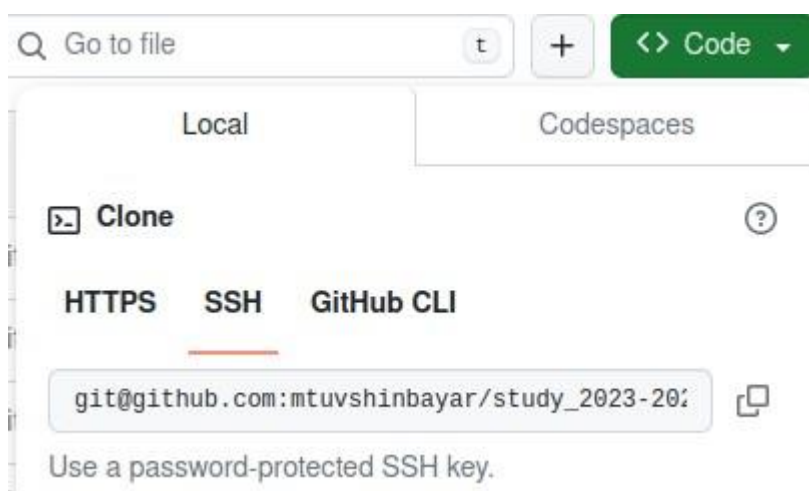


Рис 1.18

Настройка каталога курса

Перехожу в каталог arch-pc с помощью утилиты cd (рис. 1.18).

```
mtuvshinbayar@mtuvshinbayar-VirtualBox:~/work/study/2023-2024/Computer Architecture$ cd ~/work/study/2023-2024/Computer\ Architecture/arch-pc
mtuvshinbayar@mtuvshinbayar-VirtualBox:~/work/study/2023-2024/Computer Architecture/arch-pc$
```

Рис 1.18

Удаляю лишние файлы с помощью утилиты rm (рис. 1.19).

```
mtuvshinbayar@mtuvshinbayar-VirtualBox:~/work/study/2023-2024/Computer Architecture/arch-pc$ rm package.json
```

Рис 1.19

Создаю необходимые каталоги (рис. 1.20).

```
mtuvshinbayar@mtuvshinbayar-VirtualBox:~/work/study/2023-2024/Computer Architecture/arch-pc$ echo arch-pc > COURSE
mtuvshinbayar@mtuvshinbayar-VirtualBox:~/work/study/2023-2024/Computer Architecture/arch-pc$ make
```

Рис 1.20

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. 1.21).

```

mtuvshinbayar@mtuvshinbayar-VirtualBox:~/work/study/2023-2024/Computer Architecture/arch-pc$ git add .
mtuvshinbayar@mtuvshinbayar-VirtualBox:~/work/study/2023-2024/Computer Architecture/arch-pc$ git commit -am 'feat(main): make course structure'
[master 532192d] feat(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
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 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 labs/lab01/report/report.md
create mode 100644 labs/lab02/presentation/Makefile
create mode 100644 labs/lab02/presentation/image/kulyabov.jpg
create mode 100644 labs/lab02/presentation/presentation.md
create mode 100644 labs/lab02/report/Makefile
create mode 100644 labs/lab02/report/bib/cite.bib
create mode 100644 labs/lab02/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab02/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_fignos.py

```



Рис 1.21

Отправляю все на сервер с помощью push (рис. 1.22).

```
mtuvshinbayer@mtuvshinbayer-VirtualBox:~/work/study/2023-2024/Computer Architecture/arch-pc$ git push
Enumerating objects: 37, done.
Counting objects: 100% (37/37), done.
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 342.13 KiB | 2.46 MiB/s, done.
Total 35 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:mtuvshinbayer/study_2023-2024_arh-pc.git
1d3a3f1..532192d master -> master
```

Рис 1.22

Проверяю правильность выполнения работы сайте GitHub(рис. 1.23).

 **mtuvshinbayer** feat(main): make course structure 532192d · 1 minute ago  History

Name	Last commit message	Last commit date
..		
lab01	feat(main): make course structure	1 minute ago
lab02	feat(main): make course structure	1 minute ago
lab03	feat(main): make course structure	1 minute ago
lab04	feat(main): make course structure	1 minute ago
lab05	feat(main): make course structure	1 minute ago
lab06	feat(main): make course structure	1 minute ago
lab07	feat(main): make course structure	1 minute ago
lab08	feat(main): make course structure	1 minute ago
lab09	feat(main): make course structure	1 minute ago
lab10	feat(main): make course structure	1 minute ago
lab11	feat(main): make course structure	1 minute ago
README.md	feat(main): make course structure	1 minute ago

Рис 1.23

Выводы

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

Список литературы

1. Архитектура ЭВМ
2. Git - gitattributes Документация