

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

дисциплина: Архитектура компьютера

Студент:

Группа:

МОСКВА

2022 г.

Цел работу:

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

Ход работы:

1. Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab03>report).
2. Скопируйте отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.
3. Загрузите файлы на github.

```
[root@localhost-live liveuser]# git config --global user.name "cagatayiscan"
[root@localhost-live liveuser]# git config --global user.email "alyailhamovha@gmail.com"
[root@localhost-live liveuser]# git config --global core.quotepath false
[root@localhost-live liveuser]# git config --global init.defaultBranch master
[root@localhost-live liveuser]# git config --global core.autocrlf input
[root@localhost-live liveuser]# git config --global core.safecrlf warn
[root@localhost-live liveuser]# ssh-keygen -C "cagatayiscan alyailhamovha@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /root/.ssh/.
Created directory '/root/.ssh'.
/root/.ssh/. already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Saving key "/root/.ssh/." failed: Is a directory
[root@localhost-live liveuser]# ls
desktop Documents Downloads Music Pictures Public Templates Videos
[root@localhost-live liveuser]# touch sshkey.txt
[root@localhost-live liveuser]# ssh-keygen -C "cagatayiscan alyailhamovha@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /root/.ssh/id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Nf1KuAwArTR1ftQ0e926QHQMCOHU1HLwmutsqsuhUmI cagatayiscan alyailhamovha@gmail.com
The key's randomart image is:
----[RSA 3072]-----
oo...o.o+o0++o |
o o .o +.o.++oo|
. o o .- =. =.oo|
. o . o + |
. S . + |
E . . o |
. o . o |
. o . o |
.. +o.oo |
-----[SHA256]-----
[root@localhost-live liveuser]# cat ~/.ssh/id_rsa.pub | xclip -sel clip
bash: xclip: command not found...
[root@localhost-live liveuser]# sudo install xclip
install: missing destination file operand after 'xclip'
Try 'install --help' for more information.
[root@localhost-live liveuser]# sudo apt install xclip
sudo: apt: command not found
[root@localhost-live liveuser]# yum install xclip
Last metadata expiration check: 1:01:17 ago on Mon 19 Dec 2022 11:31:56 PM EST.
Dependencies resolved.
=====
Package Architecture Version Repository Size
=====
Installing:
xclip x86_64 0.13-18.git11cba61.fc37 fedora 37 k
=====
Transaction Summary
=====
Install 1 Package

Total download size: 37 k
Installed size: 63 k
Is this ok [y/N]: y
Downloading Packages:
xclip-0.13-18.git11cba61.fc37.x86_64.rpm 570 kB/s | 37 kB 00:00
Total 32 kB/s | 37 kB 00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
Installing : xclip-0.13-18.git11cba61.fc37.x86_64 1/1
Running scriptlet: xclip-0.13-18.git11cba61.fc37.x86_64 1/1
Verifying : xclip-0.13-18.git11cba61.fc37.x86_64 1/1
Installed:
xclip-0.13-18.git11cba61.fc37.x86_64
Complete!
[root@localhost-live liveuser]# cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

Activate Windows
Go to Settings to activate Windows.

```

[root@localhost-live liveuser]# cd
[root@localhost-live ~]# mkdir work
[root@localhost-live ~]# cd work
[root@localhost-live work]# mkdir study
[root@localhost-live work]# cd study
[root@localhost-live study]# mkdir 2022-2023
[root@localhost-live study]# cd 2022-2023/
[root@localhost-live 2022-2023]# mkdir arch-pc
[root@localhost-live 2022-2023]# cd arch-pc/
[root@localhost-live arch-pc]# mkdir labs
[root@localhost-live arch-pc]# cd labs
[root@localhost-live labs]# mkdir lab01 lab02 lab03
[root@localhost-live labs]# ls
lab01 lab02 lab03
[root@localhost-live labs]# cd \
>
[root@localhost-live ~]# cd work/study/2022-2023/
[root@localhost-live 2022-2023]# mkdir ComputerArchitecture
[root@localhost-live 2022-2023]# mv arch-pc ComputerArchitecture/
[root@localhost-live 2022-2023]# ls
ComputerArchitecture
[root@localhost-live 2022-2023]# mkdir -p ~/work/study/2022-2023/ComputerArchitecture/
[root@localhost-live 2022-2023]# ls
ComputerArchitecture
[root@localhost-live 2022-2023]# cd ComputerArchitecture/
[root@localhost-live ComputerArchitecture]# git clone --recursive
fatal: You must specify a repository to clone.

usage: git clone [<options>] [--] <repo> [<dir>]

    -v, --verbose            be more verbose
    -q, --quiet              be more quiet
    --progress               force progress reporting
    --reject-shallow          don't clone shallow repository
    -n, --no-checkout         don't create a checkout
    --bare                   create a bare repository
    --mirror                 create a mirror repository (implies bare)
    -l, --local              to clone from a local repository
    --no-hardlinks            don't use local hardlinks, always copy
    -s, --shared              setup as shared repository
    --recurse-submodules[=<pathspec>]
                             initialize submodules in the clone
    --recursive ...          alias of --recurse-submodules
    -j, --jobs <n>           number of submodules cloned in parallel
    --template <template-directory>
                             directory from which templates will be used
    --reference <repo>       reference repository
    --reference-if-able <repo>
                             reference repository
    --dissociate              use --reference only while cloning
    -o, --origin <name>     use <name> instead of 'origin' to track upstream
    -b, --branch <branch>   checkout <branch> instead of the remote's HEAD
    -u, --upload-pack <path>
                             path to git-upload-pack on the remote
    --depth <depth>          create a shallow clone of that depth
    --shallow-since <time>   create a shallow clone since a specific time
    --shallow-exclude <revision>
                             deepen history of shallow clone, excluding rev
    --single-branch           clone only one branch, HEAD or --branch
    --no-tags                 don't clone any tags, and make later fetches not to follow them
    --shallow-submodules      any cloned submodules will be shallow
    --separate-git-dir <gitdir>
                             separate git dir from working tree
    -c, --config <key=value>
                             set config inside the new repository
    --server-option <server-specific>
                             option to transmit
    -4, --ipv4                use IPv4 addresses only
    -6, --ipv6                use IPv6 addresses only
    --filter <args>           object filtering
    --also-filter-submodules  apply partial clone filters to submodules
    --remote-submodules       any cloned submodules will use their remote-tracking branch
    --sparse                  initialize sparse-checkout file to include only files at root

```

```
[root@localhost-live ComputerArchitecture]# git clone --recursive git@github.com:cagatayiscan/study_2022-2023_arh.pc.git arch-pc
fatal: destination path 'arch-pc' already exists and is not an empty directory.
[root@localhost-live ComputerArchitecture]# rm -r arch-pc
rm: descend into directory 'arch-pc'? y
rm: descend into directory 'arch-pc/labs'? y
rm: remove directory 'arch-pc/labs/lab03'? y
rm: remove directory 'arch-pc/labs/lab02'? y
rm: remove directory 'arch-pc/labs/lab01'? y
rm: remove directory 'arch-pc/labs'? y
rm: remove directory 'arch-pc'? y
[root@localhost-live ComputerArchitecture]# git clone --recursive git@github.com:cagatayiscan/study_2022-2023_arh.pc.git arch-pc
Cloning into 'arch-pc'...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+D1Y3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCoQU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.
```

Please make sure you have the correct access rights
and the repository exists.

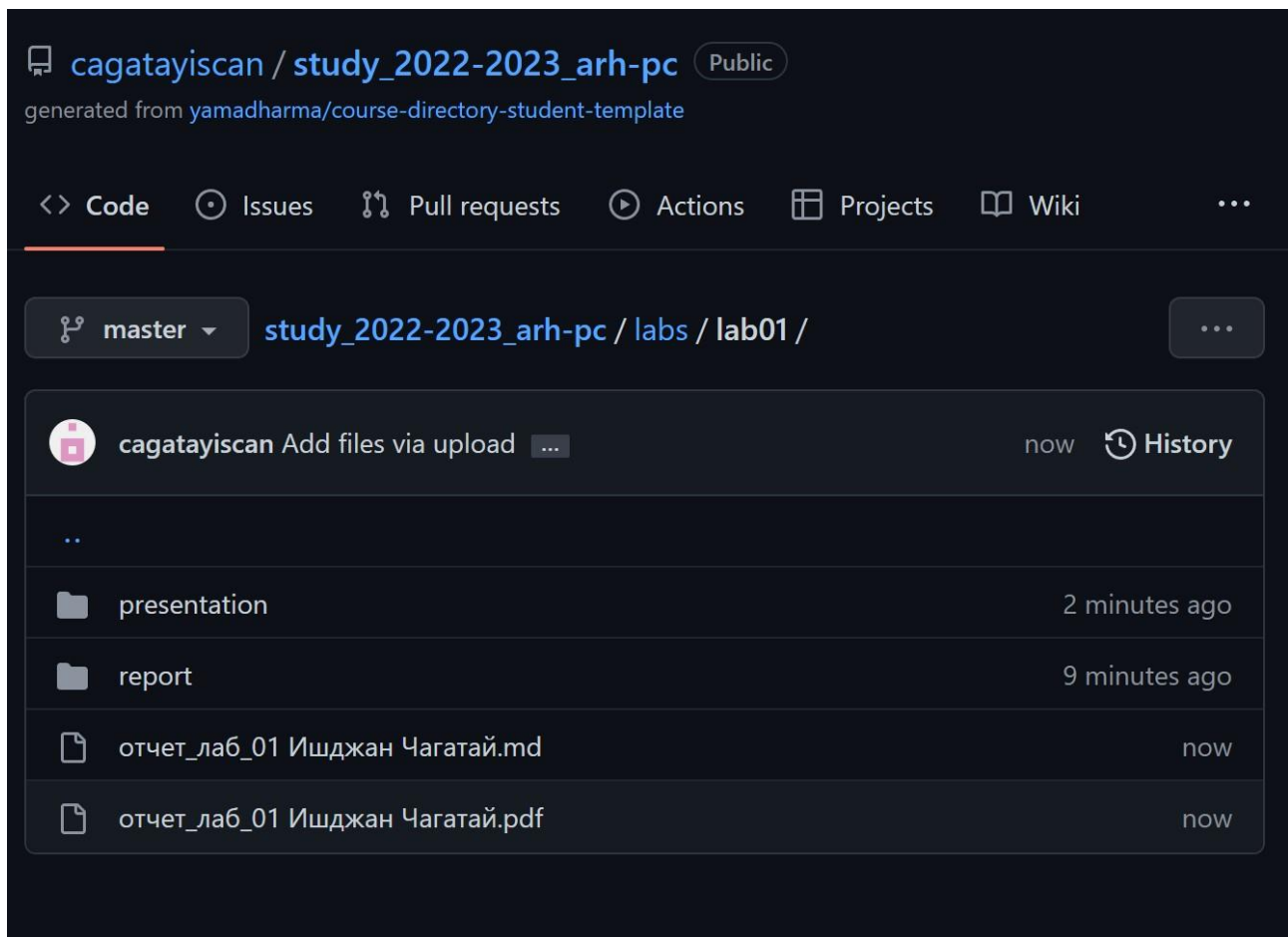
```
[root@localhost-live ComputerArchitecture]# git clone --recursive git@github.com:cagatayiscan/study_2022-2023_arh-pc.git arch-pc
Cloning into 'arch-pc'...
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.
```

Please make sure you have the correct access rights
and the repository exists.

```
[root@localhost-live ComputerArchitecture]# cd
[root@localhost-live ~]# ls
anaconda-ks.cfg  work
[root@localhost-live ~]# cd //
[root@localhost-live //]# cd
[root@localhost-live ~]# cd //
[root@localhost-live //]# ls
afs  bin  boot  dev  etc  home  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[root@localhost-live //]# cd /.ssh/
bash: cd: /.ssh/: No such file or directory
[root@localhost-live //]# cd usr
[root@localhost-live usr]# ls
bin  games  i686-w64-mingw32  include  java  lib  lib64  libexec  local  sbin  share  src  tmp  x86_64-w64-mingw32
[root@localhost-live usr]# cd src
[root@localhost-live src]# ls
debug  kernels
[root@localhost-live src]# cd usr
bash: cd: usr: No such file or directory
[root@localhost-live src]# cd .
[root@localhost-live src]# cd ..
[root@localhost-live usr]# cd
[root@localhost-live ~]# cd .ssh/
[root@localhost-live .ssh]# ls
id_rsa  id_rsa.pub  known_hosts
[root@localhost-live .ssh]# cat id_rsa
```

```
[root@localhost-live .ssh]# cd ~/work/study/2022-2023/ComputerArchitecture/
[root@localhost-live ComputerArchitecture]# git clone --recursive git@github.com:cagatayiscan/study_2022-2023_arh-pc.git arch-pc
Cloning into 'arch-pc'...
remote: Enumerating objects: 26, done.
remote: Counting objects: 100% (26/26), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 26 (delta 0), reused 17 (delta 0), pack-reused 0
Receiving objects: 100% (26/26), 16.39 KiB | 4.10 MiB/s, done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-template.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory-report-template.git) registered for path 'template/report'
Cloning into '/root/work/study/2022-2023/ComputerArchitecture/arch-pc/template/presentation'...
remote: Enumerating objects: 71, done.
remote: Counting objects: 100% (71/71), done.
remote: Compressing objects: 100% (49/49), done.
remote: Total 71 (delta 23), reused 68 (delta 20), pack-reused 0
Receiving objects: 100% (71/71), 88.89 KiB | 1.37 MiB/s, done.
Resolving deltas: 100% (23/23), done.
Cloning into '/root/work/study/2022-2023/ComputerArchitecture/arch-pc/template/report'...
remote: Enumerating objects: 78, done.
remote: Counting objects: 100% (78/78), done.
remote: Compressing objects: 100% (52/52), done.
remote: Total 78 (delta 31), reused 69 (delta 22), pack-reused 0
Receiving objects: 100% (78/78), 292.27 KiB | 1.29 MiB/s, done.
Resolving deltas: 100% (31/31), done.
Submodule path 'template/presentation': checked out '2703b47423792d472694aaf7555a5626dce51a25'
Submodule path 'template/report': checked out 'df7b2ef80f8def3b9a496f8695277469a1a7842a'
[root@localhost-live ComputerArchitecture]# cd ~/work/study/2022-2023/ComputerArchitecture/arch-pc/
[root@localhost-live arch-pc]# rm package.json
rm: remove regular file 'package.json'? y
[root@localhost-live arch-pc]# echo arch-pc > COURSE
[root@localhost-live arch-pc]# make
[root@localhost-live arch-pc]# git add .
[root@localhost-live arch-pc]# git commit -am 'feat(main): make course structure'
```

```
[root@localhost-live arch-pc]# git push
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 2 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (20/20), 310.95 KiB | 2.41 MiB/s, done.
Total 20 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:cagatayiscan/study_2022-2023_arh-pc.git
   04deb24..bd76df6  master -> master
[root@localhost-live arch-pc]#
```



Контрольные вопросы:

- Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

Контроль версий, также известный как управление исходным кодом, — это практика отслеживания изменений программного кода и управления ими. Системы контроля версий — это программные инструменты, помогающие командам разработчиков управлять изменениями в исходном коде с течением времени.

- Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

контроль версий - это класс систем, ответственных за управление изменениями в компьютерных программах, документах, крупных веб-сайтах или других коллекциях информации. Контроль версий - это компонент управления конфигурацией программного обеспечения.

- Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

централизованной:

1-Традиционные VCS были созданы для бэкапирования, отслеживания и синхронизации файлов

2-Все изменения проходят через центральный сервер

Децентрализованная система:

1-В DVCS у каждого есть свой полноценный репозиторий

2-DVCS были созданы для обмена изменениями

3-При использовании DVCS нет какой-то жестко заданной структуры репозитория с центральным сервером

- Опишите действия с VCS при единоличной работе с хранилищем.

Хранилище (repository), или репозиторий, — место хранения файлов и их версий, служебной информации.

- Каковы основные задачи, решаемые инструментальным средством git?

У Git две основных задачи: первая — хранить информацию о всех изменениях в вашем коде, начиная с самой первой строчки, а вторая — обеспечение удобства командной работы над кодом.

- Назовите и дайте краткую характеристику командам git.

git checkout

-b имя_ветки

создание новой ветки, базирующейся на текущей

git checkout

имя_ветки

переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)

git push

origin

имя_ветки

отправка изменений конкретной ветки в центральный репозиторий

git merge

--no-ff

имя_ветки

слияние ветки с текущим деревом

git branch -d

имя_ветки

удаление локальной уже слитой с основным деревом ветки

git branch -D

имя_ветки

принудительное удаление локальной ветки

git push

origin

:имя_ветки

удаление ветки с центрального репозитория