

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: Армихос Гонзалез Карла

Группа: НКАбд-02-24

МОСКВА

2025 г.

Содержание

1. Цель работы.....	4
2. Задание.....	5
3. Теоретические сведения.....	6
4. Задание для самостоятельной работы.....	8
4.1 Создать базовую конфигурацию для работы с git.....	8
4.2 Создать ключ SSH.....	8
4.3 Создать ключ PGP.....	9
4.4 Настроить подписи git.....	11
Настройка автоматических подписей коммитов git. Используя введённый email, укажите Git применять его при подписи коммитов (рис 4.4.1, 4.4.2).....	11
4.5 Зарегистрироваться на Github.....	12
4.6 Создать локальный каталог для выполнения заданий по предмету..	12
5. Выводы.....	14
7. Список используемой литературы.....	15

Список иллюстраций

рис. 4.1.1 Configuracion de Git.....	8
рис. 4.2.1 Создание SSH key.....	8
рис. 4.2.3 Добавить пароль.....	9
рис. 4.3.1 PGP key.....	10
рис 4.3.2 Ключ.....	11
рис 4.3.3.....	11
рис 4.4.1 подписей коммитов git.....	12
рис 4.4.2 подписей коммитов git.....	12
рис. 4.5.1 настройка gh.....	12
рис (4.6.1)новый каталог.....	12
рис 4.6.2 репозитория.....	12
рис 4.6.3 Git clone.....	13
рис 4.6.4 package.json.....	13
рис 4.6.5 Отбавьте файлы на север.....	13

1. Цель работы

- Изучить идеологию и применение средств контроля версий.
- Освоить умения по работе с git.

2. Задание

1. Создать базовую конфигурацию для работы с git.
2. Создать ключ SSH.
3. Создать ключ PGP.
4. Настроить подписи git.
5. Зарегистрироваться на Github.
6. Создать локальный каталог для выполнения заданий по предмету.

3. Теоретические сведения

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.

4. Задание для самостоятельной работы

4.1 Создать базовую конфигурацию для работы с git.

Базовая настройка git, зададим имя и email владельца репозитория (рис. 4.1.1)

```
gkarmikhos@fedora:~$ git config --global user.name "ArmijosKarla"
gkarmikhos@fedora:~$ git config --global user.email "karla.armijos2003@gmail.com"
gkarmikhos@fedora:~$ git config --global core.quotepath false
gkarmikhos@fedora:~$ git config --global init.defaultBranch master
gkarmikhos@fedora:~$ git config --global core.autocrlf input
gkarmikhos@fedora:~$ git config --global core.safecrlf warn
```

рис. 4.1.1 Configuracion de Git.

4.2 Создать ключ SSH

Мы создаем ключ для SSH, по алгоритму rsa с ключом размером 4096 бит: мы должны указать место, где мы хотим, чтобы наш пароль отображался. (рис. 4.2.1)

```
gkarmikhos@fedora:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/gkarmikhos/.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:m1XcHbk4/cbg8WzLY0PvUwnH3qMIFgUdg8sGZrJwpNg gkarmikhos@fedora
The key's randomart image is:
+---[RSA 4096]-----+
|  ..  .+O.  ..|
|  o..o + . +.. o.|
|  . Eo = o o o = o|
|      . = . +O*.*|
|      S o  .*B*|
|      *   .B%|
|      + . . o+=|
|      . . ..|
|      .|
+-----[SHA256]-----+
```

рис. 4.2.1 Создание SSH key

Мы копируем пароль из документов и вставляем его в Github (рис.4.2.3)

Add new SSH Key

Title

Key type

Authentication Key ▾

Key

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQADJ6WWfVejXTnpGmKMrDX8Jb4FLVFG4z4/
BGPnPdYcj1STQ9rhTCcnW6pAAbDN9uFnxFj9GImkVcEGLRMZHOjPDPuJAfStSugs2MOjI2BHDA1Dn/
zqnOLzL08X9orUVP5JVZuoH30dF+Nw3p4DrO4ufgfE1Rj/URbATpY/9vfSQWjBDnuv6QOkVUX8j2dgjgtq86ek73i/
dn1h6t+XgSvX+YK0vv415nuSFIaqwbRnyAAT7SrKPS7Uaz0wQ9mX4c7/rHCyUBUnIm4nCB/
e89sP6MAa5UDP5Qa6JzHbxHmy1Ij8IT4MahAOMfdQGEoiBNI58KwK8hkO/6kx2DkEN54cDxzqSubLSekVY+a72fiu6RHfU
AYyqCnT4cGwbHloYg0Dh9OvhBjQe/
vCnqD4mw6uLjR4RWO6uO7rgKI9lOadX4YA7oQmw7V9JPUFEExx47h3AIRMZnjFD9XLERWd0Hkay1Sciwah+5mSGEq6F8
g3D57TPgiBdodzsjaAD1Kee5wYzWzDogoVhYv11buiii27glDxbKfImzF7o8Y63bnrUuSOlPNNdKFze6bnPbpCBm4nIThhvd
afsbj9/53zwXA6egEQtVgb84MMDqyxyQB1TDpKuZzutVSDFMejjuDtKO6C0PWZEddirHQFywwqdAjjUxgpgHuhtuwn+4cF
```

Add SSH key

рис. 4.2.3 Добавить пароль

4.3 Создать ключ PGP.

Генерируем ключ и мы выбираем варианты, как вы показываете нам на картинке (рис. 4.3.1)

```

gkarmikhos@fedora:~$ gpg --full-generate-key
gpg (GnuPG) 2.4.4; Copyright (C) 2024 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) ECC (sign and encrypt) *default*
 (10) ECC (sign only)
 (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
    0 = key does not expire
    <n> = key expires in n days
    <n>w = key expires in n weeks
    <n>m = key expires in n months
    <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: honorab
Email address: karla.armijos2003@gmail.com
Comment: -
You selected this USER-ID:
    "honorab (-) <karla.armijos2003@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? n
Real name: karla
You selected this USER-ID:
    "karla (-) <karla.armijos2003@gmail.com>"

```

рис. 4.3.1 PGP key

Выводим список ключей и копируем отпечаток приватного ключа (рис 4.3.2)

```

Real name: karla
You selected this USER-ID:
    "karla (-) <karla.armijos2003@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: revocation certificate stored as '/home/gkarmikhos/.gnupg/openpgp-revocs.d/BA403C45A2F2A67200D80BF3E
E29D.rev'
public and secret key created and signed.

pub   rsa4096 2025-02-28 [SC]
      BA403C45A2F2A67200D80BF3B4D6543D85E1E29D
uid           karla (-) <karla.armijos2003@gmail.com>
sub   rsa4096 2025-02-28 [E]

gkarmikhos@fedora:~$ gpg --list-secret-keys --keyid-format LONG
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid: 2  signed: 0  trust: 0-, 0q, 0n, 0m, 0f, 2u
[keyboxd]
-----
sec   rsa4096/B4D6543D85E1E29D 2025-02-28 [SC]
      BA403C45A2F2A67200D80BF3B4D6543D85E1E29D
uid           [ultimate] karla (-) <karla.armijos2003@gmail.com>
ssb   rsa4096/4C577F015FD4E64D 2025-02-28 [E]

sec   rsa4096/B273F5BD552C67EA 2025-02-20 [SC]
      C4E775D66C9B78E38BB3FAF1B273F5BD552C67EA
uid           [ultimate] karla (kedi) <karla.armijos2003@gmail.com>
ssb   rsa4096/067A71165CD79159 2025-02-20 [E]

gkarmikhos@fedora:~$ gpg --armor --export <PGP Fingerprint> | xclip -sel clip

```

рис 4.3.2 Ключ

Скопируйте ваш сгенерированный PGP ключ в буфер обмена (рис 4.3.3)

```

gkarmikhos@fedora:~$ gpg --armor --export AD8344422F7E2BC4 2025-02-28 | xclip -sel clip
gkarmikhos@fedora:~$ kedi
bash: kedi: command not found...
gkarmikhos@fedora:~$ gpg --armor --export AD8344422F7E2BC4 2025-02-28 | xclip -sel clip
gkarmikhos@fedora:~$
gkarmikhos@fedora:~$
gkarmikhos@fedora:~$ git config --global user.signingkey AD8344422F7E2BC4 2025-02-28

```

рис 4.3.3

4.4 Настроить подписи git.

Настройка автоматических подписей коммитов git. Используя введенный email, укажите Git применять его при подписи коммитов (рис 4.4.1, 4.4.2)

```
gkarmikhos@fedora:~$ git config --global user.signingkey AD8344422F7E2BC4
gkarmikhos@fedora:~$ git config --global commit.gpgsign true
```

рис 4.4.1 подписей коммитов git

```
gkarmikhos@fedora:~$ git config --global commit.gpgsign true
gkarmikhos@fedora:~$ git config --global gpg.program $(which gpg2)
gkarmikhos@fedora:~$ S
```

рис 4.4.2 подписей коммитов git

4.5 Зарегистрироваться на Github.

Для начала необходимо авторизоваться (рис. 4.5.1)

```
gkarmikhos@fedora:~/work/study/2024-2025/Операционные системы$ gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? SSH
? Upload your SSH public key to your GitHub account? /home/gkarmikhos/.ssh/id_ed25519.pub
? Title for your SSH key: holli
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 42D5-F359
Press Enter to open https://github.com/login/device in your browser...

✓ Authentication complete.
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Uploaded the SSH key to your GitHub account: /home/gkarmikhos/.ssh/id_ed25519.pub
✓ Logged in as ArmijosKarla
gkarmikhos@fedora:~/work/study/2024-2025/Операционные системы$
```

рис. 4.5.1 настройка gh

4.6 Создать локальный каталог для выполнения заданий по предмету.

Мы создали каталог для нашего курса (рис. 4.6.1)

```
gkarmikhos@fedora:~$ mkdir -p ~/work/study/2024-2025/"Операционные системы"
gkarmikhos@fedora:~$ cd ~/work/study/2024-2025/"Операционные системы"
```

рис (4.6.1)новый каталог

Использование репозитория yamadharma (рис 4.6.2)

```
gkarmikhos@fedora:~/work/study/2024-2025/Операционные системы$
gkarmikhos@fedora:~/work/study/2024-2025/Операционные системы$ gh repo create study_2024-2025_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository ArmijosKarla/study_2024-2025_os-intro on GitHub
https://github.com/ArmijosKarla/study_2024-2025_os-intro
gkarmikhos@fedora:~/work/study/2024-2025/Операционные системы$ git clone --recursive git@github.com:<owner>/study_2024-2025_os-intro.git os-intro
bash: owner: No such file or directory
gkarmikhos@fedora:~/work/study/2024-2025/Операционные системы$
```

рис 4.6.2 репозитория

Мы клонируем файл (рис 4.6.3)

```

gkarmikhos@fedora:~/work/study/2024-2025/Операционные системы$ git clone --recursive git@github.com:ArmijosKarla/study_2024-2025_os-intro.git os-intro
Cloning into 'os-intro'...
remote: Enumerating objects: 36, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 36 (delta 1), reused 21 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (36/36), 19.38 KiB | 62.00 KiB/s, done.
Resolving deltas: 100% (1/1), 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 '/home/gkarmikhos/work/study/2024-2025/Операционные системы/os-intro/template/presentation'...
remote: Enumerating objects: 111, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (77/77), done.
remote: Total 111 (delta 42), reused 100 (delta 31), pack-reused 0 (from 0)
Receiving objects: 100% (111/111), 102.17 KiB | 88.00 KiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/gkarmikhos/work/study/2024-2025/Операционные системы/os-intro/template/report'...
remote: Enumerating objects: 142, done.
remote: Counting objects: 100% (142/142), done.
remote: Compressing objects: 100% (97/97), done.
remote: Total 142 (delta 60), reused 121 (delta 39), pack-reused 0 (from 0)
Receiving objects: 100% (142/142), 341.09 KiB | 98.00 KiB/s, done.
Resolving deltas: 100% (60/60), done.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'

```

рис 4.6.3 Git clone

Удалите лишние файлы рис (4.6.4)

```

gkarmikhos@fedora:~/work/study/2024-2025/Операционные системы$ cd os-intro
gkarmikhos@fedora:~/work/study/2024-2025/Операционные системы/os-intro$ rm package.json

```

рис 4.6.4 package.json

Создайте необходимые каталоги (рис 4.6.5)

```

gkarmikhos@fedora:~/work/study/2024-2025/Операционные системы/os-intro$ echo os-intro > COURSE
gkarmikhos@fedora:~/work/study/2024-2025/Операционные системы/os-intro$ make
Usage:
  make <target>

Targets:
  list           List of courses
  prepare        Generate directories structure
  submodule      Update submodules

gkarmikhos@fedora:~/work/study/2024-2025/Операционные системы/os-intro$ git add .
gkarmikhos@fedora:~/work/study/2024-2025/Операционные системы/os-intro$ git commit -am 'subir'
[master 346239b] subir
 2 files changed, 1 insertion(+), 14 deletions(-)
 delete mode 100644 package.json
gkarmikhos@fedora:~/work/study/2024-2025/Операционные системы/os-intro$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 272 bytes | 136.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:ArmijosKarla/study_2024-2025_os-intro.git
   a65b19e..346239b  master -> master
gkarmikhos@fedora:~/work/study/2024-2025/Операционные системы/os-intro$

```

рис 4.6.5 Отбавьте файлы на север

5. Выводы

В этой лаборатории с помощью команд мы создаем новый репозиторий. Мы также смогли создать новый каталог с информацией о новом курсе.

Как связать нашу учетную запись Github, чтобы мы могли работать, а также как сохранять, а затем загружать документы.

6. Ответы на контрольные вопросы.

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

Системы контроля версий (VCS) — это инструменты, которые позволяют управлять изменениями в коде, документах или других файлах в проекте. Они решают следующие задачи:

2. Отслеживание изменений файлов.
3. Хранение истории изменений.
4. Совместная работа над проектом.
5. Возможность отката к предыдущим версиям.
6. Разрешение конфликтов при одновременном редактировании.
7. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.
8. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.
9. Опишите действия с VCS при единоличной работе с хранилищем.
10. Опишите порядок работы с общим хранилищем VCS.
11. Каковы основные задачи, решаемые инструментальным средством **git**?
12. Назовите и дайте краткую характеристику командам **git**.
13. Приведите примеры использования при работе с локальным и удалённым репозиториями.
14. Что такое и зачем могут быть нужны ветви (branches)?
15. Как и зачем можно игнорировать некоторые файлы при commit

7. Список используемой литературы

- 1) os-intro__02.03.00: Лабораторная работа № 2
- 2) <https://github.com/yamadharm/course-directory-student-template>.