

Отчёт по лабораторной работе №2

Дисциплина: Архитектура Компьютеров и Операционные Системы

Барето Виллиан Мануел

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Создание базовой конфигурации для работы с git.	6
3.2	Создание ключ ssh	7
3.3	Создание ключ gpg	7
3.4	Создание локального каталога для выполнения заданий.	11
4	Выводы	13
5	Ответы на контрольные вопросы	14
6	Список литературы	17

Список иллюстраций

3.1	Установка git	6
3.2	Установка gh	6
3.3	имя и email владельца	6
3.4	имя начальной ветки и параметры	7
3.5	Создание ключ ssh	7
3.6	Создание ключ gpg	8
3.7	Настройки ключ gpg	8
3.8	личная информация	8
3.9	аккаунт на git	9
3.10	список ключей	9
3.11	Установка xclip	9
3.12	Копирование ключ gpg	9
3.13	Добавлен ключ gpg	10
3.14	указываю Git	10
3.15	авторизацию в gh	10
3.16	Авторизоваться через браузер.	11
3.17	Завершена авторизация	11
3.18	Создание каталог	11
3.19	Создание каталог	12
3.20	Удаление файла	12
3.21	Созданы необходимых каталогов	12
3.22	Отправление файлы на сервер	12

1 Цель работы

Изучение идеологии, применение средств контроля версий и освоение 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 Выполнение лабораторной работы

3.1 Создание базовой конфигурации для работы с git.

Устанавливаю git используя “dnf install git”:

```
willianmanuelbarreto@willianbarreto:~$ sudo -i
[sudo] senha para willianmanuelbarreto:
root@willianbarreto:~# dnf install git
Updating and loading repositories:
Repositories loaded.
Package "git-2.48.1-1.fc41.x86_64" is already installed.

Nothing to do.
root@willianbarreto:~#
```

Рис. 3.1: Установление git

С помощью dnf install gh, устанавливаю gh:

```
root@willianbarreto:~# dnf install gh
Updating and loading repositories:
Repositories loaded.
Package "gh-2.65.0-1.fc41.x86_64" is already installed.

Nothing to do.
root@willianbarreto:~#
```

Рис. 3.2: Установление gh

В качестве имя и email владельца репозитории задаю свои имя и email и настраиваю utf-8:

```
root@willianbarreto:~# git config --global user.name "WMBarreto"
root@willianbarreto:~# git config --global user.email "manuelbarretowillian@gmail.com"
root@willianbarreto:~# git config --global core.quotePath false
root@willianbarreto:~#
```

Рис. 3.3: имя и email владельца

Задаю имя начальной ветки и параметры autocrlf и safecrlf:

```
root@willianbarreto:~# git config --global init.defaultBranch master
root@willianbarreto:~# git config --global core.autocrlf input
root@willianbarreto:~# git config --global core.safecrlf warn
root@willianbarreto:~#
```

Рис. 3.4: имя начальной ветки и параметры

3.2 Создание ключ ssh

Создаю ключи ssh по алгоритму rsa с размером 4096 бит:

```
root@willianbarreto:~# ssh-keygen -t rsa -b 4096

Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase for "/root/.ssh/id_rsa" (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:wtrLE0b8CTX2YsTQstfKYynKs588rUCisGkbrg3BUHY root@willianbarreto
The key's randomart image is:
+---[RSA 4096]-----+
|  o E               |
| o . .             |
| . o .             |
| o * .             |
|+.. + X S          |
|o* + X *           |
|=o. B.X .          |
|o+++X.=            |
|oo+=+o=            |
+---[SHA256]-----+
```

Рис. 3.5: Создание ключ ssh

3.3 Создание ключ gpg

Генерирую ключ gpg --full-generate-key:

```

root@willianbarreto:~# gpg --full--generate-key
gpg: invalid option "--full--generate-key"
root@willianbarreto:~# gpg --full-generate-key
gpg (GnuPG) 2.4.5; 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.

Selecione o tipo de chave desejado:
  (1) RSA e RSA
  (2) DSA e Elgamal
  (3) DSA (apenas de assinar)
  (4) RSA (apenas de assinar)
  (9) ECC (de assinar e cifrar) *pré-definição*
 (10) ECC (apenas de assinar)
 (14) Chave do cartão existente
Sua opção? 1

```

Рис. 3.6: Создание ключ gpg

Из предложенных опций выбираю тип RSA and RSA; размер 4096; срок действия 0:

```

Sua opção? 1
As chaves RSA podem estar entre 1024 e 4096 bits de comprimento.
Qual tamanho de chave você quer? (3072) 4096
O tamanho de chave pedido é 4096 bits
Especifique quando a chave expira.
  0 = chave não expira
  <n> = chave expira em n dias
  <n>w = chave expira em n semanas
  <n>m = chave expira em n meses
  <n>y = chave expira em n anos
Quando a chave expira? (0) 0
A chave não expira de forma alguma
Isto está correto? (s/N) s

```

Рис. 3.7: Настройки ключ gpg

GPG запросил личную информацию, которая сохранится в ключе Имя и адрес электронной почты:

```

O GnuPG precisa construir uma ID de utilizador para identificar sua chave.

Nome verdadeiro: WMBarreto
Endereço de email: manuelbarretowillian@gmail.com
Comentário:
Você selecionou este USER-ID:
  "WMBarreto <manuelbarretowillian@gmail.com>"

```

Рис. 3.8: личная информация

У меня уже есть аккаунт на github, поэтому я вхожу в систему:

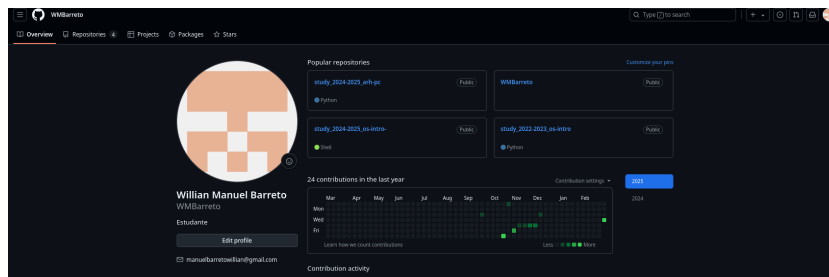


Рис. 3.9: аккаунт на git

Вывожу список ключей:

```
willianmanuelbarreto@willianbarreto:~$ gpg --list-secret-keys --keyid-format LONG
gpg: verificando a trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: profundidade: 0 válidas: 1 assinadas: 0 confiáveis: 0-, 0i, 0n, 0m, 0c, 1p
[keyboxd]
-----
sec   rsa4096/BD61DE0E9D32D1AA 2025-03-05 [SC]
      FD87CF244B78BDCE041B14D8BD61DE0E9D32D1AA
uid    [plena ] WMBBarreto <manuelbarretowillian@gmail.com>
ssb   rsa4096/36825D772AA5FC0 2025-03-05 [E]
```

Рис. 3.10: список ключей

Устанавливаю xclip:

```
willianmanuelbarreto@willianbarreto:~$ dnf install xclip
The requested operation requires superuser privileges. Please log in as a user w
ith elevated rights, or use the "--assumeno" or "--downloadonly" options to run
the command without modifying the system state.
willianmanuelbarreto@willianbarreto:~$
```

Рис. 3.11: Установление xclip

Скопирую сгенерированный gpg ключ в буфер обмена:

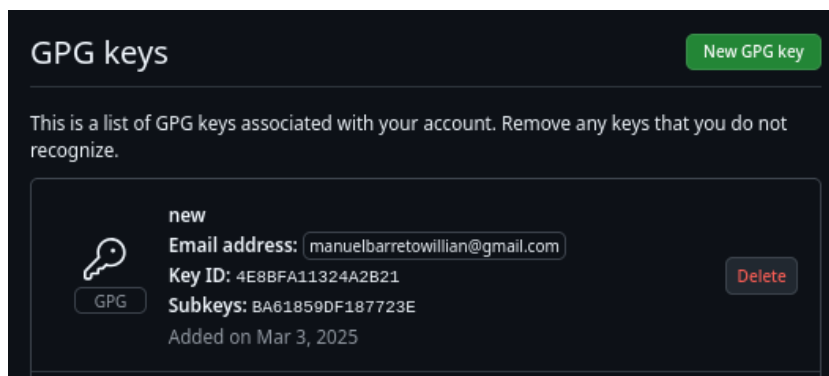


Рис. 3.12: Копирование ключ gpg

Далее перехожу в настройки GitHub, нажимаю на кнопку New GPG key и вставляю полученный ключ:

```
willianmanuelbarreto@willianbarreto:~$ git config --global user.signinkey 4E8BFA11324A2B21
willianmanuelbarreto@willianbarreto:~$ git config --global commit.gpgsign true
willianmanuelbarreto@willianbarreto:~$ git config --global gpg.programm $(which gpg2)
willianmanuelbarreto@willianbarreto:~$
```

Рис. 3.13: Добавлен ключ gpg

Используя введённый email, указываю Git применять его при подписи коммитов:

```
willianmanuelbarreto@willianbarreto:~$ 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/willianmanuelbarreto/.ssh/id_ed25519.pub
? Title for your SSH key: GitHub CLI
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: E6D2-5E03
Press Enter to open https://github.com/login/device in your browser...
```

Рис. 3.14: указываю Git

Начинаю авторизацию в gh используя gh auth login:

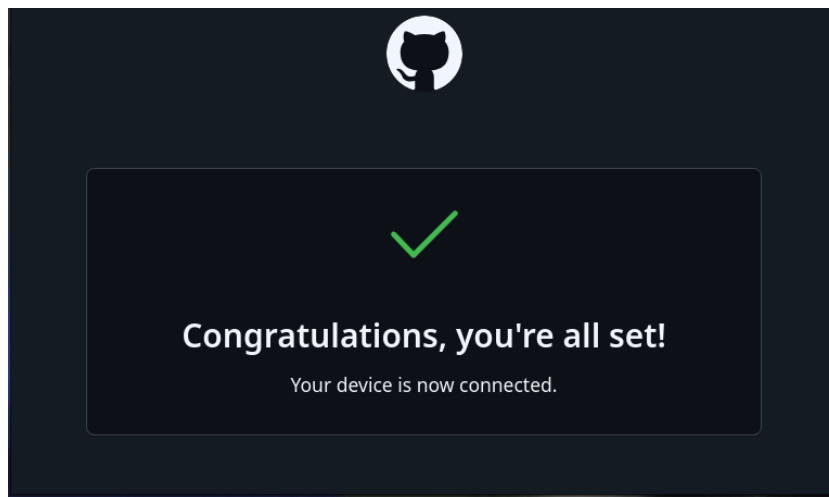


Рис. 3.15: авторизацию в gh

Завершаю авторизацию на броузер:

```
! First copy your one-time code: 3597-E9C2
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
✓ SSH key already existed on your GitHub account: /home/willianmanuelbarreto/.ssh/id_rsa.pub
✓ Logged in as WMBBarreto
```

Рис. 3.16: Авторизоваться через браузер.

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

Рис. 3.17: Завершена авторизация

3.4 Создание локального каталога для выполнения заданий.

Создаю каталог “mkdir -p ~/work/study/2022-2023/”Операционные системы”:

```
willianmanuelbarreto@willianbarreto:~/work/study/2024-2025/Операционные системы$ gh repo create study_2022-2023_os-intro --template=yamadharm/course-directory-student-template --public
✓ Created repository WMBBarreto/study_2022-2023_os-intro on GitHub
https://github.com/WMBBarreto/study_2022-2023_os-intro
willianmanuelbarreto@willianbarreto:~/work/study/2024-2025/Операционные системы$ git clone --recursive git@github.com:WMBBarreto/study_2022-2023_os-intro
.git os-intro
Clonando para '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)
A receber objetos: 100% (36/36), 19.38 KiB | 3.23 MiB/s, concluído.
A resolver deltas: 100% (1/1), concluído.
Submódulo 'template/presentation' (https://github.com/yamadharm/academic-presentation-markdown-template.git) registrado para path 'template/presentation'
Submódulo 'template/report' (https://github.com/yamadharm/academic-laboratory-report-template.git) registrado para path 'template/report'
Clonando para '/home/willianmanuelbarreto/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)
```

Рис. 3.18: Создание каталог

Перехожу в созданный каталог:

```
willianmanuelbarreto@willianbarreto:~/work/study/2024-2025/Операционные системы$ cd ~/work/study/2024-2025/"Операционные системы"/os-intro

willianmanuelbarreto@willianbarreto:~/work/study/2024-2025/Операционные системы/os-intro$ rm package.json
willianmanuelbarreto@willianbarreto:~/work/study/2024-2025/Операционные системы/os-intro$ echo os-intro > COURSE
willianmanuelbarreto@willianbarreto:~/work/study/2024-2025/Операционные системы/os-intro$
```

Рис. 3.19: Создание каталог

Удаляю лишние файлы:

Удаление файла

Рис. 3.20: Удаление файла

Создаю еще необходимые каталоги:

```
willianmanuelbarreto@willianbarreto:~/work/study/2024-2025/Операционные системы/os-intro$ make prepare
willianmanuelbarreto@willianbarreto:~/work/study/2024-2025/Операционные системы/os-intro$ git add .
```

Рис. 3.21: Создани необходимых каталогов

Отправляю Файлы на сервер:

```
willianmanuelbarreto@willianbarreto:~/work/study/2024-2025/Операционные системы/os-intro$ git add .
willianmanuelbarreto@willianbarreto:~/work/study/2024-2025/Операционные системы/os-intro$ git commit -am 'feat(main): make course structure'
[master f619a1c] feat(main): make course structure
403 files changed, 98412 insertions(+)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/.projectile
create mode 100644 labs/lab01/presentation/.texlabroot
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
```

Рис. 3.22: Отправление файлы на сервер

4 Выводы

При выполнении лабораторной работы я изучила идеологию, применение средств контроля версий и освоила умение по работе с git.

5 Ответы на контрольные вопросы

1. Системы Контроля Версий - Программные инструменты, помогающие командам разработчиков управлять изменениями в исходном коде с течением времени.
2. Хранилище - в нем хранятся все документы, включая историю их изменение и прочей служебной информацией.

commit - отслеживание изменений сохраняет разницу в изменениях.

история - Хранит все изменения в проекте и позволяет при необходимости обратиться к нужным данным.

рабочая копия- копия проекта основанная на версии из хранилища.
3. В централизованном VCS например AccuRev, каждый пользователь копирует себе необходимые ему файлы из репозитория, изменяет их а затем добавляет изменения обратно в хранилище. В децентрализованном VCS например Git, есть возможность добавлять и забирать изменения из любого репозитория.
4. Сначала создается и подключается удаленный репозиторий, затем по мере изменения проекта эти изменения отправляются на сервер.
5. Участник проекта перед началом работы получает нужную ему версию проекта в хранилище, с помощью определенных команд, после внесения изменений пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются. К ним можно вернуться в любой момент.

6. Хранение информации о всех изменениях в вашем коде, обеспечение удобства командной работы над кодом.

Создание основного дерева репозитория: `git init`

7. Получение обновлений (изменений) текущего дерева из центрального репозитория: `git pull`

Отправка всех произведённых изменений локального дерева в центральный репозиторий: `git push`

Просмотр списка изменённых файлов в текущей директории: `git status`

Просмотр текущих изменений: `git diff`

Сохранение текущих изменений: добавить все изменённые и/или созданные файлы и/или каталоги: `git add .`

добавить конкретные изменённые и/или созданные файлы и/или каталоги:
`git add имена_файлов`

удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории): `git rm имена_файлов`

Сохранение добавленных изменений:

сохранить все добавленные изменения и все изменённые файлы: `git commit -am 'Описание коммита'`

сохранить добавленные изменения с внесением комментария через встроенный редактор: `git commit`

создание новой ветки, базирующейся на текущей: `git checkout -b имя_ветки`

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

отправка изменений конкретной ветки в центральный репозиторий: `git push origin имя_ветки`

слияние ветки с текущим деревом: `git merge --no-ff имя_ветки`

Удаление ветки:

удаление локальной уже слитой с основным деревом ветки: `git branch -d имя_ветки`

принудительное удаление локальной ветки: `git branch -D имя_ветки`

удаление ветки с центрального репозитория: `git push origin :имя_ветки`

8. `git push -all` отправляем из локального репозитория все сохраненные изменения в центральный репозиторий, предварительно создав локальный репозиторий и сделав предварительную конфигурацию.
9. Ветвление - один из параллельных участков в одном хранилище, исходящих из одной версии, обычно есть главная ветка. Между ветками, т. е. их концами возможно их слияние. Используются для разработки новых функций.
10. Во время работы над проектом могут создаваться файлы, которые не следуют добавлять в репозиторий. Например, временные файлы. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл `.gitignore` с помощью сервисов.

6 Список литературы

::: Архитектура ЭВМ :::