

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ  
ФАКУЛЬТЕТ ФИЗИКО-МАТЕМАТИЧЕСКИХ И  
ЕСТЕСТВЕННЫХ НАУК**

**Лабораторная Работа №2**

**По курсу**

**Операционные Системы**

**Озьяс Стев Икнэль Дани**

**НКНбд-02-21**

**Содержание**

- 1. Цель работы**
- 2. Задание**
- 3. Теоретическое введение**
- 4. Выполнение лабораторной работы**
  - 1. Настройка github**
  - 2. Установка git-flow в Fedora Linux**
  - 3. Установка gh в Fedora Linux**
  - 4. Базовая настройка GIT**
  - 5. Создание ключа gpg**
  - 6. Настройка автоматических подписей коммитов git**
  - 7. Настройка gh**
  - 8. Создание репозитория курса на основе шаблона**
  - 9. Настройка каталога курса**
- 5. Вывод**
- 6. Контрольные вопросы**

## 1 Цель работы

Цели данной работы --- работать с командами системы контроля версий GIT на Linux, создать репозиторий курса на основе шаблона в github.

## 2 Задание

- Создать базовую конфигурацию для работы с git.
- Создать ключ SSH.
- Создать ключ PGP.
- Настроить подписи git.
- Зарегистрироваться на Github.
- Создать локальный каталог для выполнения заданий по предмету

## 3 Теоретическое введение

По определению система контроля версий (VCS) это программное обеспечение для облегчения работы с изменяющейся информацией. Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Git является распределённой системой контроля версий. Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды git с различными опциями.

В табл.1 приведено краткое описание основных команд git .

*Таблица 1: Описание основных команд git*

Команда	Описание действия
git init	создание основного дерева репозитория
git pull	получение обновлений (изменений) текущего дерева из центрального репозитория
git status	просмотр списка изменённых файлов в текущей директории
git push	отправка всех произведённых изменений локального дерева в центральный репозиторий
git diff	просмотр текущих изменения
git add .	добавить все изменённые и/или созданные файлы и/или каталоги
git add	добавить конкретные изменённые и/или созданные файлы

Команда	Описание действия
имена_файло в	и/или каталоги
git rm имена_файло в	удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории)
git commit - am 'Коммит'	сохранить все добавленные изменения и все изменённые файлы
git commit	сохранить добавленные изменения с внесением комментария через встроенный редактор
git checkout - b имя_ветки	создание новой ветки, базирующейся на текущей
git checkout имя_ветки	переключение на некоторую ветку
git push origin имя_ветки	отправка изменений конкретной ветки в центральный репозиторий
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

Более подробно об Unix см. в [1–6].

## 4 Выполнение лабораторной работы

### 4.1 Настройка github

Зарегистрировался на Github. (сайт <https://github.com>) (рис. 1)

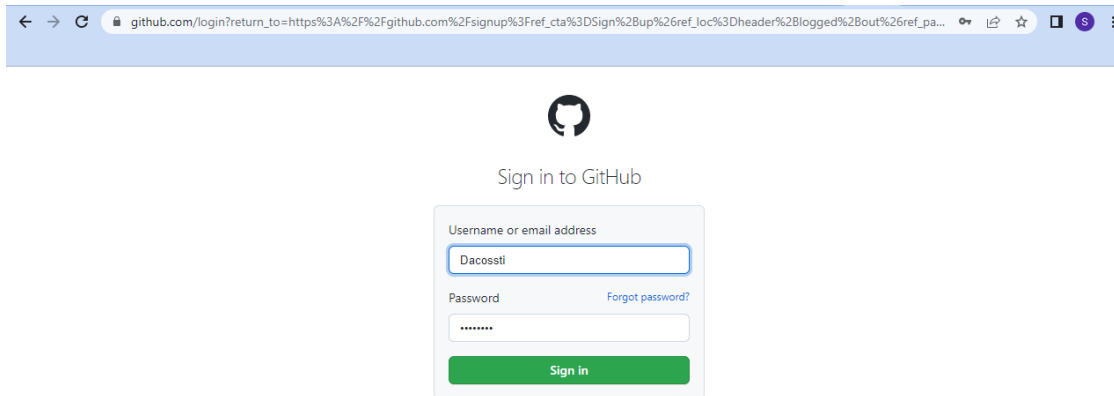


Рис. 1: Регистрация в github

## 4.2 Установка git-flow в Fedora Linux

Установил его вручную, введя следующие команды: (рис. 2)

```
Activités Visionneur d'images 26 avril 22:02
Capture d'écran de 2022-04-26 18:07:49.png

[sozjyas@fedora ~]$ cd /tmp
[sozjyas@fedora tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[sozjyas@fedora tmp]$ chmod +x gitflow-installer.sh
[sozjyas@fedora tmp]$ sudo ./gitflow-installer.sh install stable

Nous espérons que vous avez reçu de votre administrateur système local
les consignes traditionnelles. Généralement, elles se concentrent sur ces trois éléments :

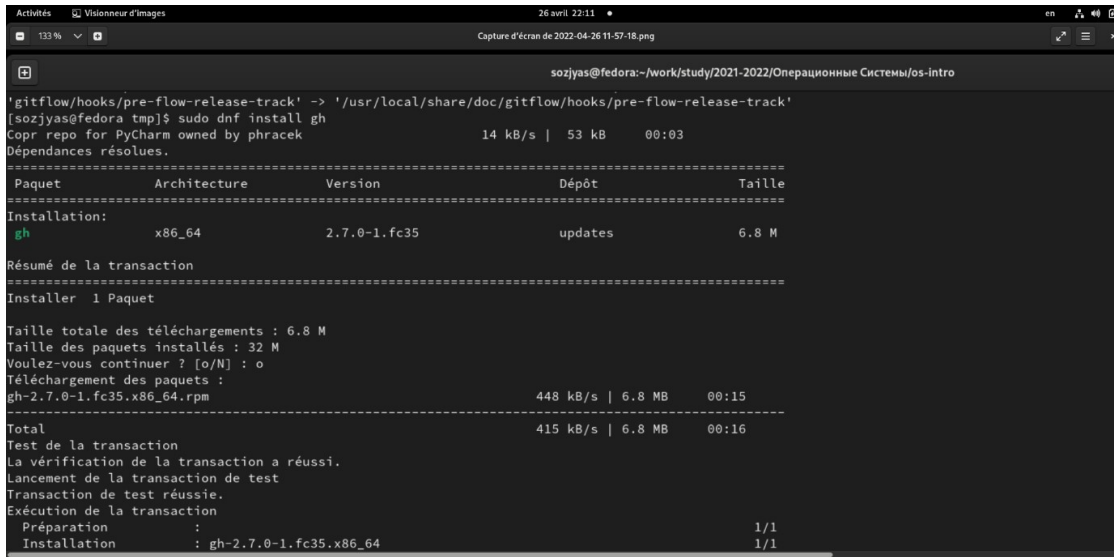
#1) Respectez la vie privée des autres.
#2) Réfléchissez avant d'utiliser le clavier.
#3) De grands pouvoirs confèrent de grandes responsabilités.

[sudo] Mot de passe de sozjyas :
## git-flow no-make installer ##
Installing git-flow to /usr/local/bin
Cloning repo from GitHub to gitflow
Cloning dans 'gitflow'...
remote: Enumerating objects: 4270, done.
remote: Total 4270 (delta 0), reused 0 (delta 0), pack-reused 4270
Réception d'objets: 100% (4270/4270), 1.74 Mio | 497.00 Kio/s, fait.
Résolution des deltas: 100% (2533/2533), fait.
Déjà à jour.
La branche 'master' est paramétrée pour suivre la branche distante 'master' depuis 'origin'.
Basculement sur la nouvelle branche 'master'
install: création du répertoire '/usr/local/share/doc'
install: création du répertoire '/usr/local/share/doc/gitflow'
install: création du répertoire '/usr/local/share/doc/gitflow/hooks'
'gitflow/git-flow' -> '/usr/local/bin/git-flow'
'gitflow/git-flow-init' -> '/usr/local/bin/git-flow-init'
'gitflow/git-flow-feature' -> '/usr/local/bin/git-flow-feature'
'gitflow/git-flow-bugfix' -> '/usr/local/bin/git-flow-bugfix'
'gitflow/git-flow-hotfix' -> '/usr/local/bin/git-flow-hotfix'
'gitflow/git-flow-release' -> '/usr/local/bin/git-flow-release'
'gitflow/git-flow-support' -> '/usr/local/bin/git-flow-support'
'gitflow/git-flow-version' -> '/usr/local/bin/git-flow-version'
'gitflow/gitflow-common' -> '/usr/local/bin/gitflow-common'
'gitflow/gitflow-shflags' -> '/usr/local/bin/gitflow-shflags'
'gitflow/git-flow-config' -> '/usr/local/share/doc/gitflow/hooks/filter-flow-hotfix-finish-tag-message'
```

Рис. 2: Установка git-flow в Fedora Linux

## 4.3 Установка gh в Fedora Linux

Использовал следующую команду: (рис. 3)



```
sozjyas@fedora:~/work/study/2021-2022/Операционные Системы/os-intro
'gitflow/hooks/pre-flow-release-track' -> '/usr/local/share/doc/gitflow/hooks/pre-flow-release-track'
[sozjyas@fedora tmp]$ sudo dnf install gh
Copr repo for PyCharm owned by phracek                               14 kB/s | 53 kB   00:03
Дépendances résolues.
=====
Paquet      Architecture  Version      Dépôt      Taille
=====
Installation:
gh          x86_64        2.7.0-1.fc35 updates    6.8 M
=====
Résumé de la transaction
=====
Installer 1 Paquet
=====
Taille totale des téléchargements : 6.8 M
Taille des paquets installés : 32 M
Voulez-vous continuer ? [o/N] : o
Téléchargement des paquets :
gh-2.7.0-1.fc35.x86_64.rpm                                         448 kB/s | 6.8 MB   00:15
-----
Total                                                             415 kB/s | 6.8 MB   00:16
-----
Test de la transaction
La vérification de la transaction a réussi.
Lancement de la transaction de test
Transaction de test réussie.
Exécution de la transaction
Préparation : 1/1
Installation : gh-2.7.0-1.fc35.x86_64 1/1
```

Рис. 3: Установка gh в Fedora Linux

#### 4.4 Базовая настройка GIT

– Задал имя и email владельца репозитория и настроил utf-8: (рис. 4)

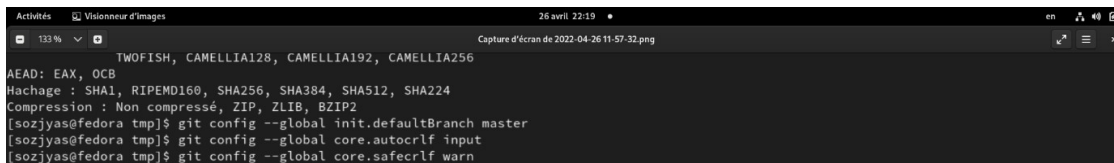


```
Installé:
gh-2.7.0-1.fc35.x86_64

Terminé !
[sozjyas@fedora tmp]$ git config --global user.name "Osiias Dacossti"
[sozjyas@fedora tmp]$ git config --global user.email "osiasstave15@gmail.com"
[sozjyas@fedora tmp]$ git config --global core.quotepath false
```

Рис. 4: Настройка имени и email владельца репозитория и utf-8

– Задал имя начальной ветки, настроил параметры autocrlf и safecrlf: (рис. 5)



```
TWOFISH, CAMELLIA128, CAMELLIA192, CAMELLIA256
AEAD: EAX, OCB
Hashage : SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression : Non compressé, ZIP, ZLIB, BZIP2
[sozjyas@fedora tmp]$ git config --global init.defaultBranch master
[sozjyas@fedora tmp]$ git config --global core.autocrlf input
[sozjyas@fedora tmp]$ git config --global core.safecrlf warn
```

Рис. 5: Настройка имени начальной ветки и параметров autocrlf и safecrlf

– Установил пакет xclip для ввода команды “xclip” для того, чтобы создать ключ. gpg (рис. 6)



```
Activités Visionneur d'images 26 avril 22:21 en
133 % Capture d'écran de 2022-04-26 11:58-08.png
Install package 'xclip' to provide command 'xclip'? [N/y] y

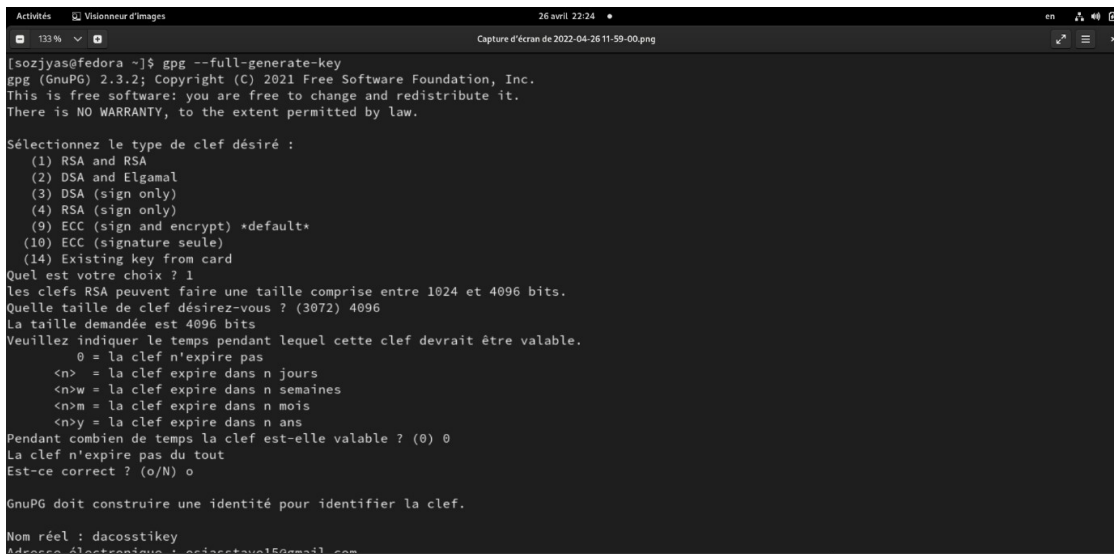
* Waiting in queue...
* Loading list of packages...
The following packages have to be installed:
xclip-0.13-15.git11c6a61.fc35.x86_64 Command line clipboard grabber
Proceed with changes? [N/y] y

* Waiting in queue...
* Waiting for authentication...
* Waiting in queue...
* Downloading packages...
* Requesting data...
* Testing changes...
* Installing packages...
```

Рис. 6: Установка пакета *xclip*

## 4.5 Создание ключа gpg

- Сгенерировал ключ (рис. 7) :



```
Activités Visionneur d'images 26 avril 22:24 en
133 % Capture d'écran de 2022-04-26 11:59-00.png
[sozjyas@fedora ~]$ gpg --full-generate-key
gpg (GnuPG) 2.3.2; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

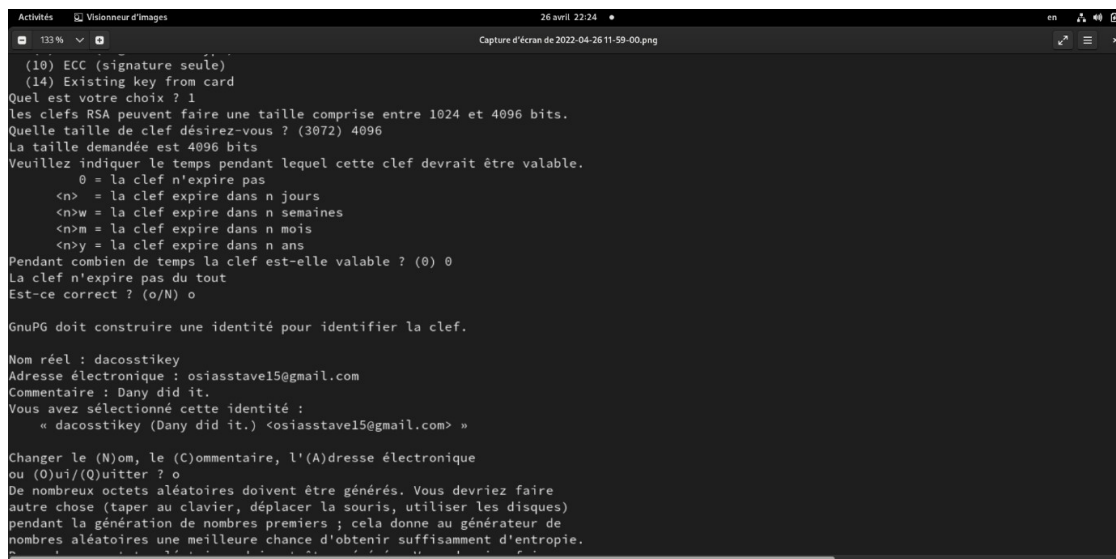
Sélectionnez le type de clef désiré :
(1) RSA and RSA
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
(9) ECC (sign and encrypt) *default*
(10) ECC (signature seule)
(14) Existing key from card
Quel est votre choix ? 1
les clefs RSA peuvent faire une taille comprise entre 1024 et 4096 bits.
Quelle taille de clef désirez-vous ? (3072) 4096
La taille demandée est 4096 bits
Veuillez indiquer le temps pendant lequel cette clef devrait être valable.
  0 = la clef n'expire pas
  <n> = la clef expire dans n jours
  <n>w = la clef expire dans n semaines
  <n>m = la clef expire dans n mois
  <n>y = la clef expire dans n ans
Pendant combien de temps la clef est-elle valable ? (0) 0
La clef n'expire pas du tout
Est-ce correct ? (o/N) o

GnuPG doit construire une identité pour identifier la clef.

Nom réel : dacosstikey
Adresse électronique : sofia@taul15@gmail.com
```

Рис. 7: Генерирование ключа GPG

- И установил необходимые настройки (рис. 8):



```
Activités Visionneur d'images 26 avril 22:24
133 % Capture d'écran de 2022-04-26 11:59-00.png

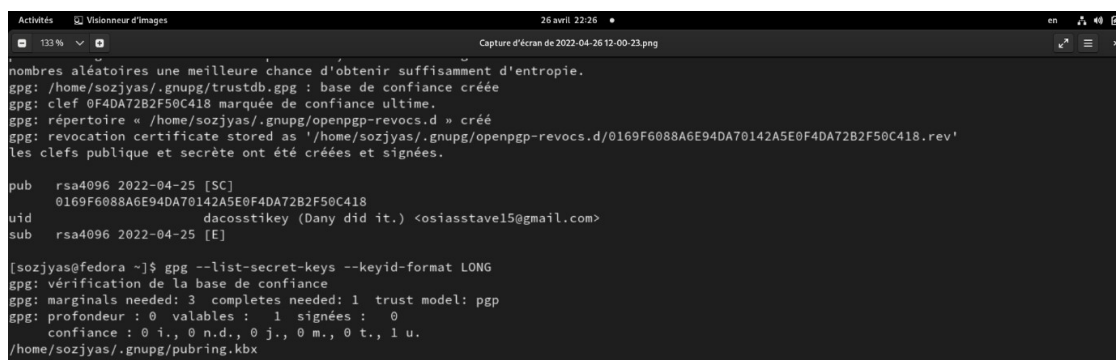
(10) ECC (signature seule)
(14) Existing key from card
Quel est votre choix ? 1
les clefs RSA peuvent faire une taille comprise entre 1024 et 4096 bits.
Quelle taille de clef désirez-vous ? (3072) 4096
La taille demandée est 4096 bits
Veuillez indiquer le temps pendant lequel cette clef devrait être valable.
  0 = la clef n'expire pas
  <n> = la clef expire dans n jours
  <n>w = la clef expire dans n semaines
  <n>m = la clef expire dans n mois
  <n>y = la clef expire dans n ans
Pendant combien de temps la clef est-elle valable ? (0) 0
La clef n'expire pas du tout
Est-ce correct ? (o/N) o

GnuPG doit construire une identité pour identifier la clef.
Nom réel : dacosstikey
Adresse électronique : osiasstavel5@gmail.com
Commentaire : Dany did it.
Vous avez sélectionné cette identité :
  « dacosstikey (Dany did it.) <osiasstavel5@gmail.com> »

Changer le (N)om, le (C)ommentaire, l'(A)dresse électronique
ou (O)ui/(Q)uitter ? o
De nombreux octets aléatoires doivent être générés. Vous devriez faire
autre chose (taper au clavier, déplacer la souris, utiliser les disques)
pendant la génération de nombres premiers ; cela donne au générateur de
nombres aléatoires une meilleure chance d'obtenir suffisamment d'entropie.
```

Рис. 8: Установка настроек

- Выводил список ключей (рис. 9)



```
Activités Visionneur d'images 26 avril 22:26
133 % Capture d'écran de 2022-04-26 12:00-23.png

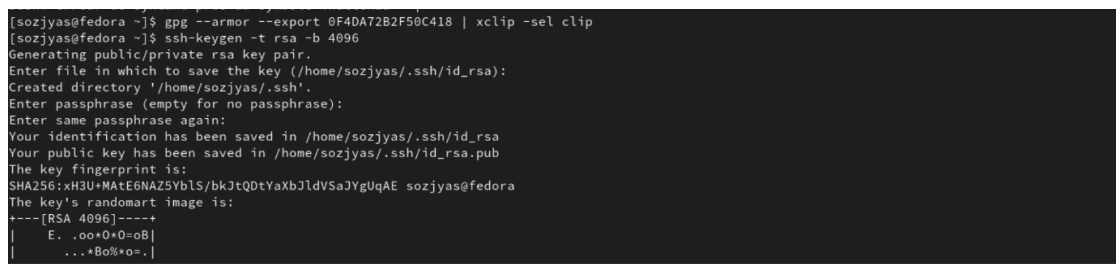
nombres aléatoires une meilleure chance d'obtenir suffisamment d'entropie.
gpg: /home/sozjyas/.gnupg/trustdb.gpg : base de confiance créée
gpg: clef 0F4DA72B2F50C418 marquée de confiance ultime.
gpg: répertoire « /home/sozjyas/.gnupg/openpgp-revocs.d » créé
gpg: revocation certificate stored as '/home/sozjyas/.gnupg/openpgp-revocs.d/0169F6088A6E94DA70142A5E0F4DA72B2F50C418.rev'
les clefs publique et secrète ont été créées et signées.

pub  rsa4096 2022-04-25 [SC]
    0169F6088A6E94DA70142A5E0F4DA72B2F50C418
uid          dacosstikey (Dany did it.) <osiasstavel5@gmail.com>
sub  rsa4096 2022-04-25 [E]

[sozjyas@fedora ~]$ gpg --list-secret-keys --keyid-format LONG
gpg: vérification de la base de confiance
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: profondeur : 0 valables : 1 signées : 0
confiance : 0 i., 0 n.d., 0 j., 0 m., 0 t., 1 u.
/home/sozjyas/.gnupg/pubring.kbx
```

Рис. 9: Список ключей

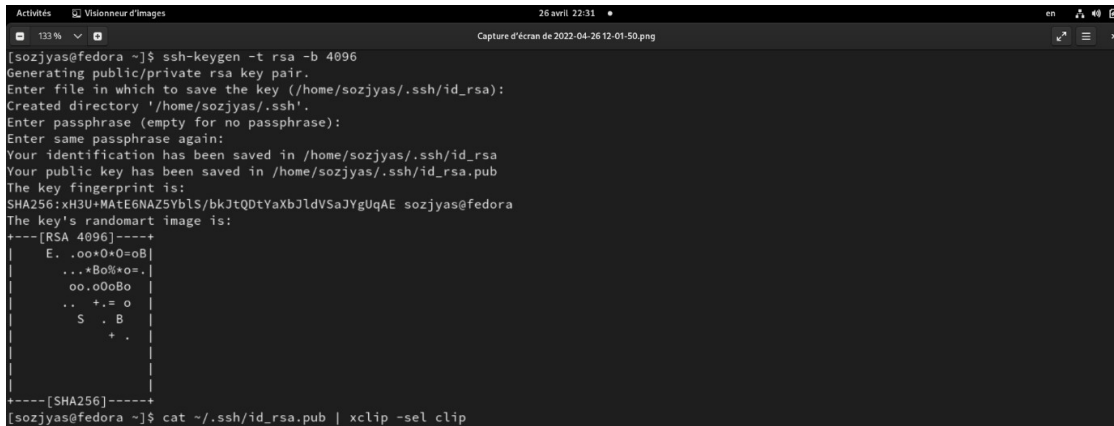
- Скопировал сгенерированный ключ PGP в буфер обмена (рис. 10)



```
[sozjyas@fedora ~]$ gpg --armor --export 0F4DA72B2F50C418 | xclip -sel clip
[sozjyas@fedora ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/sozjyas/.ssh/id_rsa):
Created directory '/home/sozjyas/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/sozjyas/.ssh/id_rsa
Your public key has been saved in /home/sozjyas/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:xH3U+MatE6NAZ5Yb1S/bk3tQDtYaXbJldV5aJYgUqAE sozjyas@fedora
The key's randomart image is:
+--[RSA 4096]-----+
|  E. .oo*o*o=oB|
| ...*Bo%*o=. |
```

Рис. 10: Копирование ключа GPG в буфер обмена

- Сразу создал ключ ssh (рис. 11)



```
[sozjyas@fedora ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/sozjyas/.ssh/id_rsa):
Created directory '/home/sozjyas/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/sozjyas/.ssh/id_rsa
Your public key has been saved in /home/sozjyas/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:xH3U+MatE6NAZ5Yb1S/bk3tQtYaxb1dVsaJYgUqAE sozjyas@fedora
The key's randomart image is:
+--[RSA 4096]-----+
|  E. .oo+0+0=0=0B|
|  ...*Bo%*0=+ |
|  oo.o0oBo      |
|  .. +.= o      |
|  S . B          |
|  + .            |
+-----+
[SHA256]-----+
[sozjyas@fedora ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

Рис. 11: Генерирование ключа SSH

- И добавил их в GitHub (<https://github.com/settings/keys>) (рис. 12)

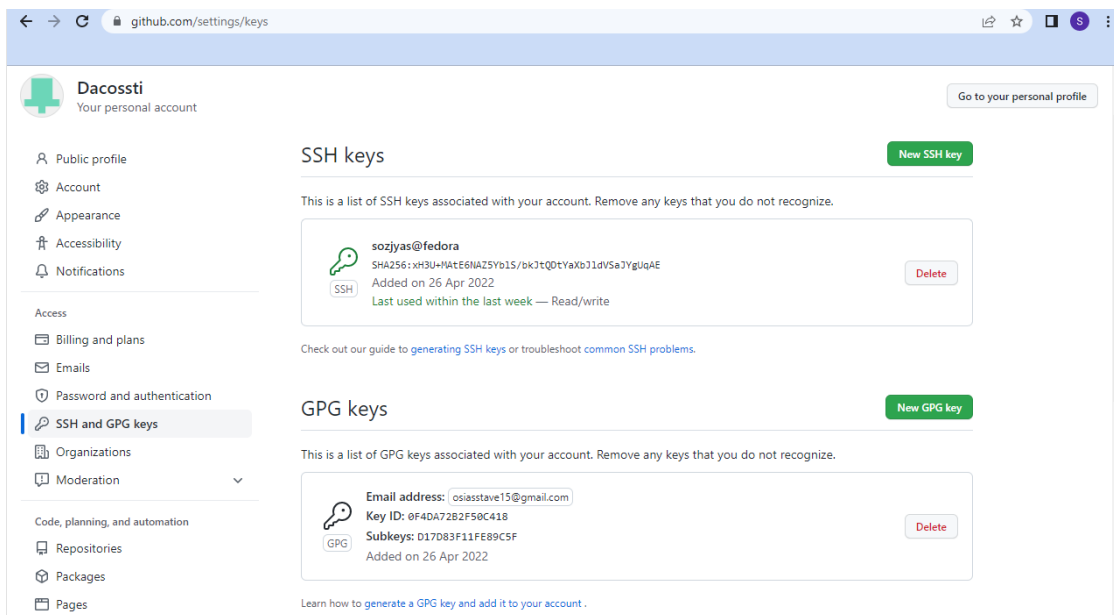


Рис. 12: Добавление ключа GPG и ключа SSH в GitHub

## 4.6 Настройка автоматических подписей коммитов git

- Настроил автоматические подписи коммитов git: (рис. 13)



```
[sozjyas@fedora ~]$ git config --global user.signingkey 0F4DA72B2F50C418
[sozjyas@fedora ~]$ git config --global commit.gpgsign true
[sozjyas@fedora ~]$ git config --global gpg.program $(which gpg)
[sozjyas@fedora ~]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

First copy your one-time code: 5B84-5F20
Press Enter to open github.com in your browser...
/ Authentication complete.
- gh config set -h github.com git_protocol https
/ Configured git protocol
/ Logged in as Dacossti
[sozjyas@fedora ~]$ mkdir -p ~/work/study/2021-2022/"Операционные Системы"
```

Рис. 13: Подписание коммитов git

## 4.7 Настройка gh

- Затем настроил gh (рис. 14)

```
Activities Visionneur d'images 26 avril 22:33
133% Capture d'écran de 2022-04-26 12:02:34.png
[sozjyas@fedora ~]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

First copy your one-time code: 5B84-5F20
Press Enter to open github.com in your browser...
/ Authentication complete.
- gh config set -h github.com git_protocol https
/ Configured git protocol
/ Logged in as Dacossti
```

Рис. 14: Настройка gh

## 4.8 Создание репозитория курса на основе шаблона

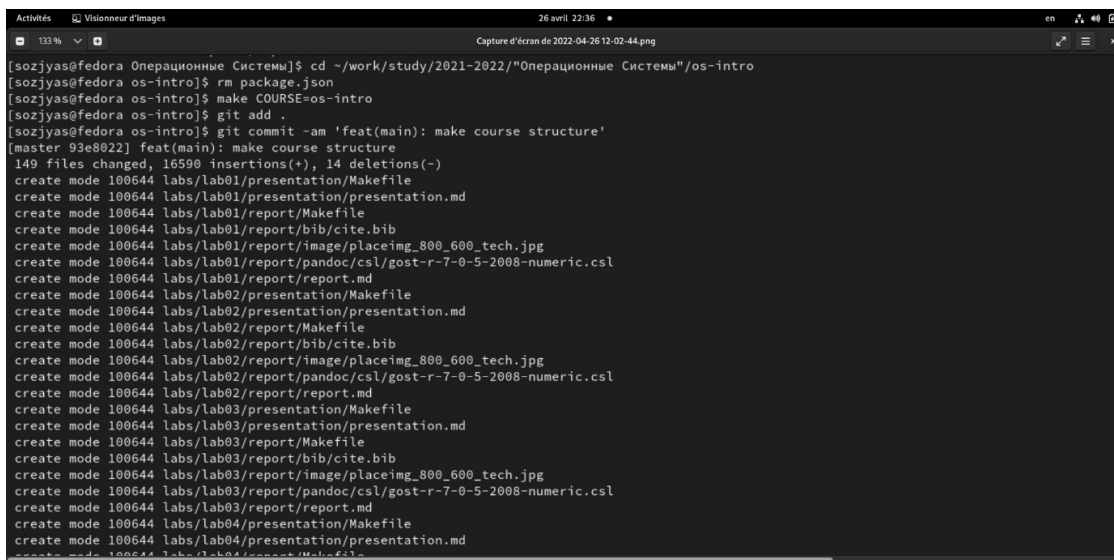
- Создал шаблон рабочего пространства (рис. 15)

```
Activities Visionneur d'images 26 avril 22:35
133% Capture d'écran de 2022-04-26 12:02:34.png
/ Logged in as Dacossti
[sozjyas@fedora ~]$ mkdir -p ~/work/study/2021-2022/"Операционные Системы"
[sozjyas@fedora ~]$ cd ~/work/study/2021-2022/"Операционные Системы"
[sozjyas@fedora Операционные Системы]$ gh repo create study_2021-2022_os-intro --template=yamadharma/course-directory-student-template --public
/ Created repository Dacossti/study_2021-2022_os-intro on GitHub
[sozjyas@fedora Операционные Системы]$ git clone --recursive git@github.com:Dacossti/study_2021-2022_os-intro.git os-intro
Clonage dans 'os-intro'...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
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.
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 20 (delta 2), reused 15 (delta 2), pack-reused 0
Réception d'objets: 100% (20/20), 12.49 Kio | 2.50 Mio/s, fait.
Résolution des deltas: 100% (2/2), fait.
Sous-module 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-template.git) enregistré pour le chemin 'temp
Sous-module 'template/report' (https://github.com/yamadharma/academic-laboratory-report-template.git) enregistré pour le chemin 'template/repor
Clonage dans '/home/sozjyas/work/study/2021-2022/Операционные Системы/os-intro/template/presentation'...
remote: Enumerating objects: 42, done.
remote: Counting objects: 100% (42/42), done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 42 (delta 9), reused 40 (delta 7), pack-reused 0
Réception d'objets: 100% (42/42), 31.19 Kio | 638.00 Kio/s, fait.
Résolution des deltas: 100% (9/9), fait.
Clonage dans '/home/sozjyas/work/study/2021-2022/Операционные Системы/os-intro/template/report'...
remote: Enumerating objects: 78, done.
remote: Counting objects: 100% (78/78), done.
```

Рис. 15: Создание репозитория курса

## 4.9 Настройка каталога курса

- Внутри каталога курса удалил ненужные файлы, создал необходимые каталоги и отправил файлы на сервер (рис. 16)



```
[sozjyas@fedora Операционные Системы]$ cd ~/work/study/2021-2022/"Операционные Системы"/os-intro
[sozjyas@fedora os-intro]$ rm package.json
[sozjyas@fedora os-intro]$ make COURSE=os-intro
[sozjyas@fedora os-intro]$ git add .
[sozjyas@fedora os-intro]$ git commit -am 'feat(main): make course structure'
[master 93e8022] feat(main): make course structure
149 files changed, 16590 insertions(+), 14 deletions(-)
create mode 100644 labs/lab01/presentation/Makefile
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/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 100644 labs/lab02/report/report.md
create mode 100644 labs/lab03/presentation/Makefile
create mode 100644 labs/lab03/presentation/presentation.md
create mode 100644 labs/lab03/report/Makefile
create mode 100644 labs/lab03/report/bib/cite.bib
create mode 100644 labs/lab03/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab03/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab03/report/report.md
create mode 100644 labs/lab04/presentation/Makefile
create mode 100644 labs/lab04/presentation/presentation.md
create mode 100644 labs/lab04/report/Makefile
```

Рис. 16: Настройка каталога курса

## 4.10 ВЫВОД

Таким образом, мы только что создали репозиторий курса "Операционные Системы", основанный на шаблоне профессора на github.

## Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. [Learning the bash Shell: Unix Shell Programming](#). O'Reilly Media, 2005. 354 с.
3. Zarrelli G. [Mastering Bash](#). Packt Publishing, 2017. 502 с.
4. Robbins A. [Bash Pocket Reference](#). O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.

## 5 Контрольные вопросы

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

- Система контроля версий (VCS) - программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.
- Такие системы наиболее широко используются при разработке программного обеспечения для хранения исходных кодов разрабатываемой программы. Однако они могут с успехом применяться и в других областях, в которых ведётся работа с большим количеством непрерывно изменяющихся электронных документов.

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

- Хранилище – репозиторий - место хранения всех версий и служебной информации.
- Commit - это команда для записи индексированных изменений в репозиторий.
- История – место, где сохраняются все коммиты, по которым можно посмотреть данные о коммитах.
- Рабочая копия – текущее состояние файлов проекта, основанное на версии, загруженной из хранилища.

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

- Централизованные системы – это системы, в которых одно основное хранилище всего проекта, и каждый пользователь копирует необходимые ему файлы, изменяет и вставляет обратно.
- Пример – Subversion.

- Децентрализованные системы – система, в которой каждый пользователь имеет свой вариант репозитория и есть возможность добавлять и забирать изменения из репозитория.
- Например, Git.

#### 4. Опишите действия с VCS при единоличной работе с хранилищем.

- Сначала сделаем предварительную конфигурацию, указав имя и email владельца репозитория: 1 `git config --global user.name "Имя Фамилия"` 2 `git config --global user.email "work@mail"` и настроив utf-8 в выводе сообщений `git: 1 git config --global quotepath false` Для инициализации локального репозитория, расположенного, например, в каталоге `~/tutorial`, необходимо ввести в командной строке: Кулябов Д. С. и др. Операционные системы 25 1 `cd` 2 `mkdir tutorial` 3 `cd tutorial` 4 `git init` После это в каталоге `tutorial` появится каталог `.git`, в котором будет храниться история изменений. Создадим тестовый текстовый файл `hello.txt` и добавим его в локальный репозиторий:
  1. `echo 'hello world' > hello.txt`
  2. `git add hello.txt`
  3. `git commit -am 'Новый файл'`
- Воспользуемся командой `status` для просмотра изменений в рабочем каталоге, сделанных с момента последней ревизии: 1 `git status` Во время работы над проектом так или иначе могут создаваться файлы, которые не требуется добавлять в последствии в репозиторий. Например, временные файлы, создаваемые редакторами, или объектные файлы, создаваемые компиляторами.
- Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл `.gitignore` с помощью сервисов. Для этого сначала нужно получить список имеющихся шаблонов:
  1. `curl -L -s https://www.gitignore.io/api/list` Затем скачать шаблон, например, для C и C++
  2. `curl -L -s https://www.gitignore.io/api/c >> .gitignore`
  3. `curl -L -s https://www.gitignore.io/api/c++ >> .gitignore`

#### 5. Опишите порядок работы с общим хранилищем VCS.

- Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый):
  1. `ssh-keygen -C "Имя Фамилия"`
- Ключи сохраняются в каталоге `~/.ssh/`. Существует несколько доступных серверов репозитория с возможностью бесплатного размещения данных. Например, <https://github.com/>. Для работы с ним необходимо сначала зайти на сайте <https://github.com/> учётную запись. Затем необходимо загрузить сгенерённый нами ранее открытый ключ. Для этого зайти на сайт <https://github.com/> под своей учётной записью и перейти в меню GitHub setting . После этого выбрать в боковом меню GitHub setting SSH-ключи и нажать кнопку Добавить ключ . Скопировав из локальной консоли ключ в буфер обмена
  1. `cat ~/.ssh/id_rsa.pub | xclip -sel clip`
- И так вставляем ключ в появившееся на сайте поле. После этого можно создать на сайте репозиторий, выбрав в меню Репозитории Создать репозиторий , дать ему название и сделать общедоступным (публичным). Для загрузки репозитория из локального каталога на сервер выполняем следующие команды:
  - 1 `git remote add origin`
  2. `ssh://git@github.com//.git`
  3. `git push -u origin master`
- Далее на локальном компьютере можно выполнять стандартные процедуры для работы с git при наличии центрального репозитория.

## 6. Каковы основные задачи, решаемые инструментальным средством git?

- У Git две основных задачи:
  1. Первая — хранить информацию обо всех изменениях в вашем коде, начиная с самой первой строчки,
  2. А вторая — обеспечение удобства командной работы над кодом.

## 7. Назовите и дайте краткую характеристику командам git.

Команда	Описание действия
git init	создание основного дерева репозитория
git pull	получение обновлений (изменений) текущего дерева из центрального репозитория
git status	просмотр списка изменённых файлов в текущей директории
git push	отправка всех произведённых изменений локального дерева в центральный репозиторий
git diff	просмотр текущих изменения
git add .	добавить все изменённые и/или созданные файлы и/или каталоги
git add имена_файло в	добавить конкретные изменённые и/или созданные файлы и/или каталоги
git rm имена_файло в	удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории)
git commit - am 'Коммит'	сохранить все добавленные изменения и все изменённые файлы
git commit	сохранить добавленные изменения с внесением комментария через встроенный редактор
git checkout - b имя_ветки	создание новой ветки, базирующейся на текущей
git checkout имя_ветки	переключение на некоторую ветку
git push origin имя_ветки	отправка изменений конкретной ветки в центральный репозиторий
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	удаление ветки с центрального репозитория

## 8. Приведите примеры использования при работе с локальным и удалённым репозиториями

- Для инициализации локального репозитория, расположенного, например, в каталоге ~/tutorial, необходимо ввести в командной строке:
  1. `cd`
  2. `mkdir tutorial`
  3. `cd tutorial`
  4. `git init`
- После это в каталоге tutorial появится каталог .git, в котором будет храниться история изменений. Создадим тестовый текстовый файл hello.txt и добавим его в локальный репозиторий:
  1. `echo 'hello world' > hello.txt`
  2. `git add hello.txt` 3 `git commit -am 'Новый файл'` Воспользуемся командой `status` для просмотра изменений в рабочем каталоге, сделанных с момента последней ревизии:
  3. `git status`

## 9. Что такое и зачем могут быть нужны ветви (branches)?

- Ветка (англ. branch) — это последовательность коммитов, в которой ведётся параллельная разработка какого-либо функционала. Ветки нужны, чтобы несколько программистов могли вести работу над одним и тем же проектом или даже файлом одновременно, при этом не мешая друг другу. Кроме того, ветки используются для тестирования экспериментальных функций: чтобы не повредить основному проекту, создается новая ветка специально для экспериментов.

## 10. Как и зачем можно игнорировать некоторые файлы при commit?

- Игнорируемые файлы — это, как правило, артефакты сборки и файлы, генерируемые машиной из исходных файлов в вашем репозитории, либо файлы, которые по какой-либо иной причине не должны попадать в коммиты. В Git нет специальной команды для указания игнорируемых файлов: вместо этого необходимо вручную отредактировать файл .
- Временно игнорировать изменения в файле можно с командой `git update-index--assumeunchanged<file>`