

Лабораторная работа №2

Дисциплина: Архитектура компьютера

Орлов Илья Сергеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Контрольные вопросы	13
6	Выводы	15

Список иллюстраций

4.1	Пример конфигурации git	8
4.2	Генерация ключей	9
4.3	Экспорт ключа	10
4.4	Конфигурация подписей для коммитов	10
4.5	Авторизация на github	11
4.6	Создание директории курса	11
4.7	Настройка директории	12

Список таблиц

1 Цель работы

Целью данной работы является изучение идеологии и применения средств контроля версий и освоение умения по работе с git.

2 Задание

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

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.

4 Выполнение лабораторной работы

Произвожу базовую настройку git. (рис. 4.1)

```
root@isorlov:~# git config --global user.email "orlovilia86@icloud.com"
root@isorlov:~# git config --global user.name "Cocico4kalmao"
root@isorlov:~# git config --global core.quotepath false
root@isorlov:~# git config --global core.autocrlf input
root@isorlov:~# git config --global core.safecrlf warn
root@isorlov:~#
```

Рис. 4.1: Пример конфигурации git

Создаю ssh и gpg ключи. (рис. 4.2)


```

Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
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:
SHA256:4i9wsQYiQDp/EHv3aCZAnLsQgVNm5ra+pPBVeK+x/Dc root@isorlov
The key's randomart image is:
+---[RSA 4096]-----+
|+=B.                  |
|*+oo                  |
|+o+.. .               |
|o+o=.o.o             |
| ooo+.BoS            |
| ....B+o             |
|. o .+o .            |
|. + o ..= E          |
|. o   +oo. .         |
+----[SHA256]-----+
root@isorlov:~# gpg --full-generate-key
gpg (GnuPG) 2.4.7; 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.

gpg: создан каталог '/root/.gnupg'
Выберите тип ключа:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) ECC (sign and encrypt) *default*
 (10) ECC (только для подписи)
 (14) Existing key from card
Ваш выбор? 1
длина ключей RSA может быть от 1024 до 4096.
Какой размер ключа Вам необходим? (3072) 4096

```

Рис. 4.2: Генерация ключей

Экспортирую gpg ключ для авторизации на github. (рис. 4.3)

```

9xPWS8C6GZrpSbeUElGEVYP1KxSpzeeKRMCLFnCZzPMm2Nhg1EuUSk90ePCfHZ
/qIEmLnRXPzG8NlyHMw8IsWTmP4oY5+EvA1uwmHnPMV1gW08eaASrcypIQARAQAB
tCZDb2NpY280a2FsbWVlIDxvcmxvdmlsaWEwNk8pY2xvdWQuY29tPokCUQQTAAQAA
0xYhBeOLgwzT/u0NxoEtpySUxp+6qMNoBQJomiIRAhSDBQsJCAcCAiICBhUKCQGL
AgQWAgMBAh4HAheAAAOJECsUxp+6qMNoYNIQAL/wF+nN1JGuQwXretkw3hn8SS23
CsjaY2S14g0PQZgVXFRTBgycNtjens8PSPXX+Q8uk/SAXvrUpldpz19ak4fbvwNe
ihcpJbSJ2HF1omS0/1817Zw/HWGMG67cP0n9F8G8qAgUfFwT1JJ+F6EmANzDNv1
hvM+vuC+EwbpyI5QIoz93NBaX+0gTJ7biqS1JbxwVDdIfR2khvgqXVHk3z/8KyOZ
EO+jtn+2omdMDdQnd0fHs6iBB8AbgQbTF6nUxzkephg28YI/LI5HJNQLqbHir+tr
7o4gf01tjyhhjhqZ8BYMTSlh6o2xEEsi71ogtNH10mFrcxS7Cu6fcjJvJd7jd1qS
TvQP7eBw4IXCfYnIVmM72vklHvFTziEfTx79JN29pv4Kh3adcKI00SUjF0qKYxT
BjeKCfrdA5Io+CnRtT7TC4pDzzvozlrTu1DMFcVvriFC1LP6VXEbXj6/XqqK7dw
4tF7VgjrBv7BMEwRLsDPoqTjOFxHYo4nIgJ90ZKnnGKBMDH6L0NHS8Dg6YwaHn/w
/tXH40eAun7r5klyqvHeL/Htn58Iigr4t/jWdbewT4E7x0QsJXh4EzmC5jda2186
2EooPe7v47P2hHyeWB3rEkCFdNGQ9Q14No4u0kBoqgWPynRUV6D/yTW0HNTyrL
746UDfn2WbRWUavUQIN8G1aKJEBCALBGfZe31x2XgMTotinxSR03769+rm12B
OCTZUyphqoELpar+Ja3xgw0510q0Kj3xv+WnZcVxyC40T1bJNYrFuLpww3+uv2UZ
GzANmMgRTSfxyedcVuMo95Vq/wUq09EqaJX6BP8Hu5e03bSuPhj0TeFYrF1Cm
2oUD1Ih18+q/YW6MMND0bweLj/yWwD3oM+McU0XRlXm4YUm117wmAWN2Hc166Gh
0DyBps28Zc6zzatRvsAZ6a5561R2cTxfrvRdqkE/ET/rj3zafvR0kZ270nPB3ZDv
Vc42Gee/4kgIJHZFj5TyCYj5gWVz8jRNHmg7gf7TebmHTa2glQST0Ls31TNYjzYU
VGQqvUqkPHCXjph70hX1PCd1ZXQLUnfnbgQG17QLLCWc0QaPa1lt0cbEAa3yBjQk
PiWUHR1GUj5eln7kX/us465oZDns1ljV4xAgCdp7Nv8vxbPf4f+hIrTuChJxuIEa
jibAwgq/x6jKFzV+Tzr0bWDCAvX2TlVJd1gGK9b/TQFPBu4KprTLdc1Gln6vTid/
acycdj5qmFeDZ+JGyiMxis3FjaHUVbSPZbUNXxc720uo1MrZxUxng7E+47CvHWgr
64JXzjpCbsfBD4tD0j4q0ZKHhyYlA6vy2YLYBMu9mHFfvgnF+cIg5JUrb5rt8gkD
o/qU0FQfEwARAQABiQI2B8gBCgAgFiEESguDDNP+7Q3E4S2nJJTGn7qow2gFamia
KJEC6wwACgkQJJTGn7qow2jVVw/+12K+T92Sy16YNcMS88mUt0nGmA7+U08YRDtt
Mm5fvCJAK2IuFD3pE0cJczYz5RS0WD/FC09CIQYze38TGnX8teqFyiuHXCWZs3Q8
BQHlPYtNvpeD0x6SwlIT9Tmmymim610+mUis2lbzJ3ejeGRxYn0UAApWPNB3Ka
mA3Naye3jrrAOQYsWueARXTtmz03VTQivBHf/PJyZdAJ1s/LmhrJ4q900Dmr1QUL
sy3XqJm8VgxWE1V63BR6hx++3vwdqR7y093M9j6MX9rAydbyEQVUf268KW639jze
3ZybhS38Bj+zHwB6P4IGmmQ91hPdLJ+wUy9w6futerT1eg4d+ZR5/VbwEHn2hgT
rXZnPILA2h6T0qZx0jorqTkR1Evfo9qdBPjPjw28u4M1wcWJQW3yX76yy3aTP0G
v2TDKd6NK+XUlnqAt7P6uF5b1S6D/+H6Z+YjK4Zyo0MntVKqgbpevAJK870a8Y
fLsMVD2YDt3ULHJ85UyWGe2hJ6wjvJHMgABJmrGwT+He6mzoD6CcPsAhb9u3W/GT
F+9Rq1C59eydFxsVPLvSd+7PePqP90VDW4Xtj/yH1sLs5wjF51BzVCyFHgsP4CX1
LzwFRMwSNwrp0a++awsTc1EjfMW2hEhJ01gJ1bTsL/JCupDm6k/dBqrZdvWjUfC
xHWKgdQ=
=6GCS
-----END PGP PUBLIC KEY BLOCK-----
root@isorlov:~#

```

Рис. 4.3: Экспорт ключа

Настраиваю автоматические подписи для коммитов. (рис. 4.4)

```

root@isorlov:~# git config --global user.signkey
root@isorlov:~# git config --global user.signkey 2494C69FBAAC8C368
root@isorlov:~# git config --global commit.gpgsign true
root@isorlov:~# git config --global gpg.program $(which gpg2)
root@isorlov:~#

```

Рис. 4.4: Конфигурация подписей для коммитов

Авторизуюсь на github для работы через терминал. (рис. 4.5)

```

isortlov@isortlov:~$ gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? SSH
? Generate a new SSH key to add to your GitHub account? Yes
? Enter a passphrase for your new SSH key (Optional): *****
? Title for your SSH key: gpg key
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: E6C2-9B35
Press Enter to open https://github.com/login/device in your browser... ~[[B^[[B^[[B^[[B^[[B^
restorecon: SELinux: Could not get canonical path for /home/isortlov/.mozilla/firefox/*/gmp-widevinecdm/
✓ Authentication complete.
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Uploaded the SSH key to your GitHub account: /home/isortlov/.ssh/id_ed25519.pub
✓ Logged in as Cocico4kalmao

```

Рис. 4.5: Авторизация на github

Создаю директорию курса по шаблону (рис. 4.6)

```

isortlov@isortlov:~$ mkdir -p ~/work/study/2024-2025/os
isortlov@isortlov:~$ cd ~/work/study/2024-2025/os
isortlov@isortlov:~/work/study/2024-2025/os$ gh repo create study_2024-2025_os-intro --template-yamadharma/course-directory-student-template --public
nowin flag: --template-yamadharma/course-directory-student-template

gh: gh repo create [<name>] [<flags>]

qs:
--add-readme      Add a README file to the new repository
c, --clone        Clone the new repository to the current directory
d, --description string  Description of the repository
--disable-issues  Disable issues in the new repository
--disable-siki    Disable siki in the new repository
g, --gitignore string  Specify a gitignore template for the repository
h, --homepage URL    Repository home page URL
--include-all-branches  Include all branches from template repository
--internal         Make the new repository internal
l, --license string  Specify an Open Source license for the repository
--private          Make the new repository private
--public           Make the new repository public
--push             Push local commits to the new repository
r, --remote string  Specify remote name for the new repository
s, --source string  Specify path to local repository to use as source
t, --team name      The name of the organization team to be granted access
p, --template repository  Make the new repository based on a template repository

isortlov@isortlov:~/work/study/2024-2025/os$ git clone --recursive git@github.com:Cocico4kalmao/study_2024-2025_os-intro.git os-intro
Cloning into 'os-intro'...
remote: Enumerating objects: 1, done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), done.
isortlov@isortlov:~/work/study/2024-2025/os$ git clone --recursive git@github.com:Cocico4kalmao/study_2024-2025_os-intro.git os-intro
Cloning into 'os-intro'...
remote: Enumerating objects: 1, done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), done.

```

Рис. 4.6: Создание директории курса

Настраиваю рабочую директорию (рис. 4.7)

```

isorlov@isorlov:~/work/study/2024-2025/os$ cd os-intro
isorlov@isorlov:~/work/study/2024-2025/os/os-intro$ rm package.json
rm: невозможно удалить 'package.json': Нет такого файла или каталога
isorlov@isorlov:~/work/study/2024-2025/os/os-intro$ nano COURSE
isorlov@isorlov:~/work/study/2024-2025/os/os-intro$ make prepare
make: «prepare» не требует обновления.
isorlov@isorlov:~/work/study/2024-2025/os/os-intro$ make
Usage:
  make <target>

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

isorlov@isorlov:~/work/study/2024-2025/os/os-intro$
isorlov@isorlov:~/work/study/2024-2025/os/os-intro$ git add.
git: «add.» не является командой git. Смотрите «git --help».

Самые похожие команды:
  add
isorlov@isorlov:~/work/study/2024-2025/os/os-intro$ git add .
isorlov@isorlov:~/work/study/2024-2025/os/os-intro$ git commit -m "feat(make):make course structure"
Author identity unknown

*** Пожалуйста, скажите мне кто вы есть.

Запустите

  git config --global user.email "you@example.com"
  git config --global user.name "Ваше Имя"

для указания идентификационных данных аккаунта по умолчанию.
Пропустите параметр --global для указания данных только для этого репозитория.

fatal: не удалось выполнить автоопределение адреса электронной почты (получено «isorlov@isorlov.(none)»)
isorlov@isorlov:~/work/study/2024-2025/os/os-intro$ git push

```

Рис. 4.7: Настройка директории

5 Контрольные вопросы

1.VCS - системы для контроля изменений в файлах. Нужны для сохранения истории, совместной работы и отката изменений.

2.Основные понятия:

Хранилище - база версий файлов

Коммит - сохранение изменений

История - все коммиты

Рабочая копия - локальные файлы

3.Виды VCS:

Централизованные (SVN) - одно хранилище

Децентрализованные (Git) - у каждого полная копия

4.Работа в одиночку:

`git init`

`git add`

`git commit`

`git log`

5.Совместная работа:

`git clone`

`git pull`

`git push`

Решение конфликтов

6.Задачи Git:

Контроль версий

Ветвление

Совместная работа

7.Основные команды:

init, clone, add, commit

push, pull, branch, merge

checkout, status, log

8.Примеры: Локально: git init → add → commit Удалённо: clone → push/pull

9.Ветки - для параллельной работы над разными задачами. Пример: git branch
feature

10..gitignore - чтобы не коммитить ненужные файлы (логи, бинарники). Пример:
*.log tmp/

6 Выводы

В результате выполнения данной лабораторной работы я приобрел необходимые навыки работы с гит, научился созданию репозитория, gpg и ssh ключей, настроил каталог курса и авторизовался в gh.