

Лабораторная работа №2

Архитектура вычислительных систем

Басманова Дарья Кириллова

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Цель работы | 4 |
| 2 | Задание | 5 |
| 3 | Выполнение лабораторной работы | 6 |
| 4 | Выводы | 10 |
| 5 | Ответы на контрольные вопросы. | 11 |

Список иллюстраций

| | | | |
|-----|-----|-----------|---|
| 3.1 | 111 | | 6 |
| 3.2 | 1 | | 6 |
| 3.3 | 2 | | 7 |
| 3.4 | 3 | | 8 |
| 3.5 | 4 | | 8 |
| 3.6 | 5 | | 9 |

1 Цель работы

Целью данной работы являются такие задачи как: 1. Изучить идеологию и применение средств контроля версий. 2. Освоить умения по работе с git.

2 Задание

Заданием на данную работу является: 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

- 1) Для начала работы нам необходимо создать шаблон рабочего пространства. Для нашего учебного года и предмета “Операционные системы” (код предмета os-intro) создание репозитория примет следующий вид:

```
os-intro: bash — Konsole
dkbasmanova@dk3n63 ~$ mkdir -p ~/work/study/2022-2023/"Операционные системы"
dkbasmanova@dk3n63 ~$ cd ~/work/study/2022-2023/"Операционные системы"
dkbasmanova@dk3n63 ~/work/study/2022-2023/Операционные системы$ git repo create study_2022-2023
```

Рис. 3.1: 111

- 2) Сделаем базовую настройку git. Во первых, зададим имя и почту владельца репозитория. Далее настроим utf-8 в выводе сообщений git. Зададим имя начальной ветки. Далее настроим параметр autocrlf и saferlf.

```
os-intro: bash — Konsole
Файл Правка Вид Закладки Модули Настройка Справка
[+] Новая вкладка [x] Разделить окно [K] Копировать [V] Вставить [F] Найти
dkbasmanova@dk3n63 ~$ git config --global user.name "Daria Basmanova"
dkbasmanova@dk3n63 ~$ git config --global user.email "db1486@yandex.ru"
dkbasmanova@dk3n63 ~$ git config --global core.quotepath false
dkbasmanova@dk3n63 ~$ git config --global init.defaultBranch master
dkbasmanova@dk3n63 ~$ git config --global core.autocrlf input
dkbasmanova@dk3n63 ~$ git config --global core.safecrlf warn
dkbasmanova@dk3n63 ~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/d/k/dkbasmanova/.ssh/id_rsa): ssh-keygen -t ed25519
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ssh-keygen -t ed25519
Your public key has been saved in ssh-keygen -t ed25519.pub
The key fingerprint is:
SHA256:7R03oK+h2g517y0c8LpKva7V8skSZ0uk/2ZQhiIwC dkbasmanova@dk3n63
The key's randomart image is:
+----[RSA 4096]-----+
|
|..
|E oo+
|+.B.*
|S /.0.o
|+./.* o
|..o * o
|... * = +
|...o.o.o
+----[SHA256]-----+
dkbasmanova@dk3n63 ~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/d/k/dkbasmanova/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/d/k/dkbasmanova/.ssh/id_ed25519
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/d/k/dkbasmanova/.ssh/id_ed25519.pub
```

Рис. 3.2: 1

3) Создадим Ключи ssh

По алгоритму rsa создадим с ключём размером 4096 бит. А так же по алгоритму ed25519.

4) Создадим ключи pgr. Генерируем ключ с помощью команды:

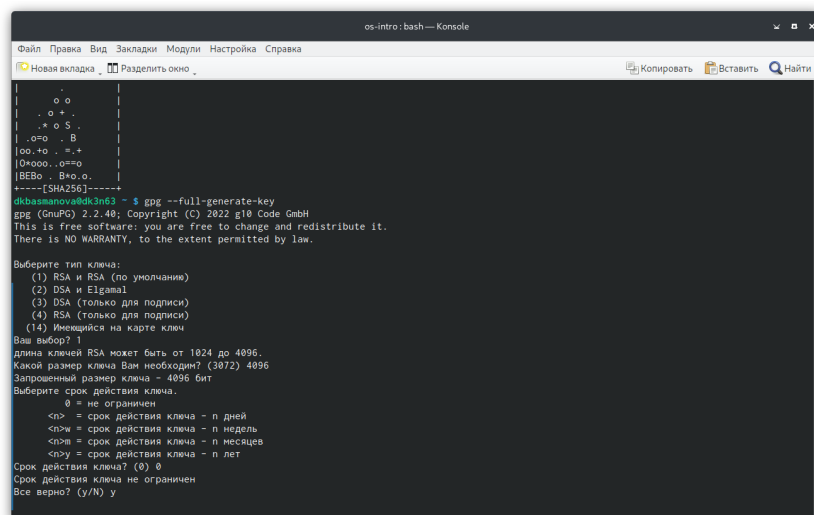
`gpg --full-generate-key` И выбираем из предложенных опций: тип RSA and RSA; размер 4096; выберите срок действия 0 (срок действия не истекает никогда).

Вводим личную информацию, которая сохранится в ключе:

Имя (Daria).

Адрес электронной почты: `db1406@yandex.ru`

Комментарий: оставим это поле пустым.



```
os-intro: bash — Konsole
Файл Правка Вид Закладки Модули Настройка Справка
Новая вкладка Разделить окно
Копировать Вставить Найти

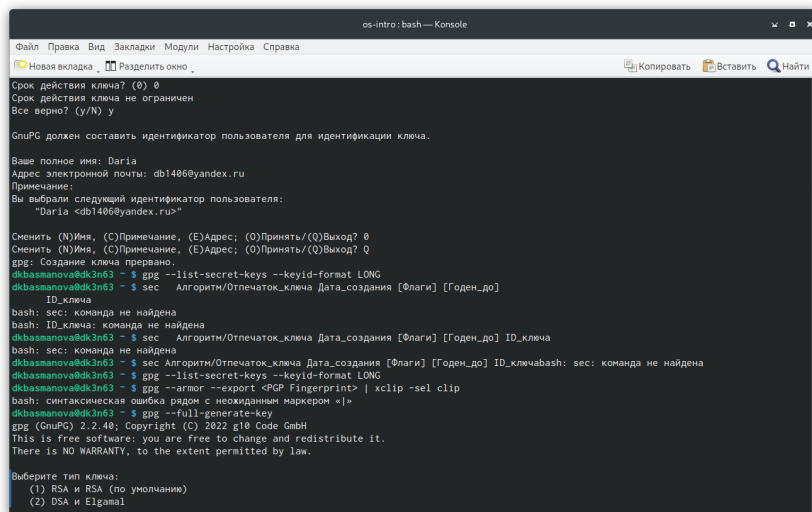
      .
     .o.
    .o+.
   .*o S.
  .o+o .B
 |oo+o .+.
|O+ooo .o+o
|BEO .B+o.o.
+----[SHA256]-----
gpg (GnuPG) 2.2.40; Copyright (C) 2022 g10 Code GmbH
This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Выберите тип ключа:
  (1) RSA и RSA (по умолчанию)
  (2) DSA и Elgamal
  (3) DSA (только для подписи)
  (4) RSA (только для подписи)
  (14) Иные на карте ключ
Ваш выбор? 1
длина ключей RSA может быть от 1024 до 4096.
Какой размер ключа Вам необходим? (3072) 4096
Запрошенный размер ключа - 4096 бит
Выберите срок действия ключа.
  0 = не ограничен
  <n> = срок действия ключа - n дней
  <n>m = срок действия ключа - n недель
  <n>y = срок действия ключа - n месяцев
  <n>u = срок действия ключа - n лет
Срок действия ключа? (0) 0
Срок действия ключа не ограничен
Все верно? (y/N) y
```

Рис. 3.3: 2

5) Мы уже создали учетную запись еще в начале прошлого семестра.

6) Добавим PGP ключа в Github.



```
os-intro: bash — Konsole
Файл Правка Вид Закладки Модули Настройка Справка
[+] Новая вкладка [M] Разделить окно
Срок действия ключа? (0) 0
Срок действия ключа не ограничен
Все верно? (y/N) y

GnuPG должен составить идентификатор пользователя для идентификации ключа.

Ваше полное имя: Daria
Адрес электронной почты: db1406@yandex.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
  "Daria <db1406@yandex.ru>"

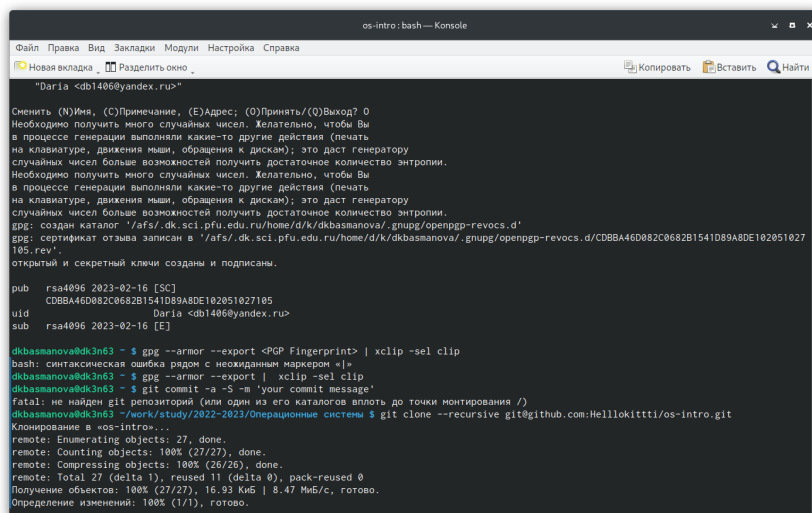
Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? 0
Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? Q
gpg: Создание ключа преверено.
dkbasmanova@dk3n63 ~$ gpg --list-secret-keys --keyid-format LONG
dkbasmanova@dk3n63 ~$ gpg --sec Алгоритм/Отпечаток_ключа Дата_создания [Флаги] [Годен_до]
ID_ключа
bash: sec: команда не найдена
bash: ID_ключа: команда не найдена
dkbasmanova@dk3n63 ~$ gpg --sec Алгоритм/Отпечаток_ключа Дата_создания [Флаги] [Годен_до] ID_ключа
bash: sec: команда не найдена
dkbasmanova@dk3n63 ~$ gpg --sec Алгоритм/Отпечаток_ключа Дата_создания [Флаги] [Годен_до] ID_ключа
bash: sec: команда не найдена
dkbasmanova@dk3n63 ~$ gpg --list-secret-keys --keyid-format LONG
dkbasmanova@dk3n63 ~$ gpg --armor --export <PGP Fingerprint> | xclip -sel clip
bash: синтаксическая ошибка рядом с неожиданным маркером «|»
dkbasmanova@dk3n63 ~$ gpg --full-generate-key
gpg (GnuPG) 2.2.40; Copyright (C) 2022 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.

Выберите тип ключа:
(1) RSA и RSA (по умолчанию)
(2) DSA и ElGamal
```

Рис. 3.4: 3

Отпечаток ключа — это последовательность байтов, используемая для идентификации более длинного, по сравнению с самим отпечатком ключа.

7) Скопируем наш сгенерированный PGP ключ в буфер обмена



```
os-intro: bash — Konsole
Файл Правка Вид Закладки Модули Настройка Справка
[+] Новая вкладка [M] Разделить окно
"Daria <db1406@yandex.ru>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? 0
Необходимо получить много случайных чисел. Хотелось бы, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печатать
на клавиатуре, движения мыши, обращения к диску); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
Необходимо получить много случайных чисел. Хотелось бы, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печатать
на клавиатуре, движения мыши, обращения к диску); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
gpg: создан каталог "/afs/dk.sci.pfu.edu.ru/home/d/k/dkbasmanova/.gnupg/openpgp-revocs.d"
gpg: сертификат отозван в "/afs/dk.sci.pfu.edu.ru/home/d/k/dkbasmanova/.gnupg/openpgp-revocs.d/C0BBA46D082C0682B1541D89A8DE102051027105.rev"
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2023-02-16 [SC]
      C0BBA46D082C0682B1541D89A8DE102051027105
uid    Daria <db1406@yandex.ru>
sub    rsa4096 2023-02-16 [E]

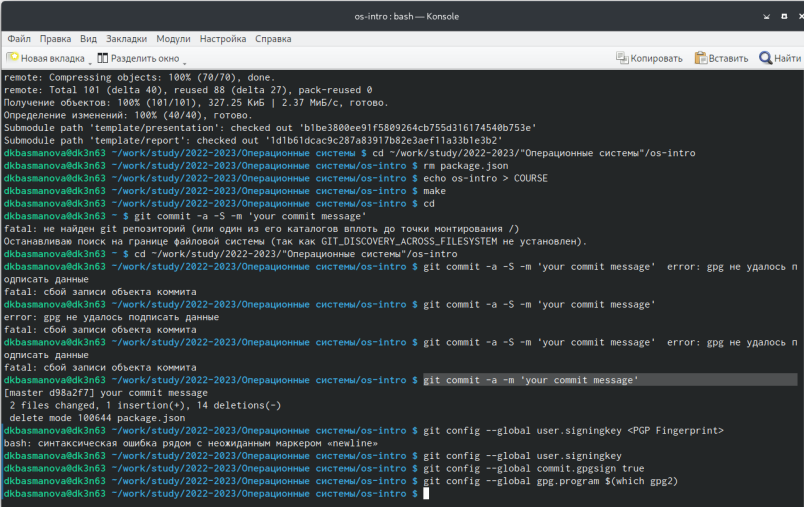
dkbasmanova@dk3n63 ~$ gpg --armor --export <PGP Fingerprint> | xclip -sel clip
bash: синтаксическая ошибка рядом с неожиданным маркером «|»
dkbasmanova@dk3n63 ~$ gpg --armor --export | xclip -sel clip
dkbasmanova@dk3n63 ~$ git commit -a -S -m 'your commit message'
fatal: не найден git репозиторий (или один из его каталогов вплоть до точки монтирования /)
dkbasmanova@dk3n63 ~/work/study/2022-2023/Операционные системы $ git clone --recursive git@github.com:Helllokititi/os-intro.git
Клонирование в «os-intro»...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Получение объектов: 100% (27/27), 16.93 KiB | 8.47 МБ/с, готово.
Определение изменений: 100% (1/1), готово.
```

Рис. 3.5: 4

8) Далее переходим в настройки GitHub (<https://github.com/settings/keys>), нажимаем на кнопку New GPG key и вставляем полученный ключ в поле ввода.

9) Настройка автоматических подписей коммитов git

Используя введенный email, укажите Git применять его при подписи коммитов.



```
os-intro: bash — Konsole
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
[+] Новая вкладка  [x] Разделить окно  [K] Копировать  [V] Вставить  [F] Найти

remote: Compressing objects: 100% (70/70), done.
remote: Total 101 (delta 40), reused 88 (delta 27), pack-reused 0
Получение объектов: 100% (101/101), 327.25 KiB | 2.37 MiB/c, готово.
Определение изменений: 100% (40/40), готово.
Submodule path 'template/presentation': checked out 'b1be3880ee91f5809264cb755d316174548b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3aef11a33b1e3b2'
dkbasmanova@dk3n63 ~/work/study/2022-2023/Операционные системы $ cd ~/work/study/2022-2023/Операционные системы/os-intro
dkbasmanova@dk3n63 ~/work/study/2022-2023/Операционные системы/os-intro $ rm package.json
dkbasmanova@dk3n63 ~/work/study/2022-2023/Операционные системы/os-intro $ echo os-intro > COURSE
dkbasmanova@dk3n63 ~/work/study/2022-2023/Операционные системы/os-intro $ make
dkbasmanova@dk3n63 ~/work/study/2022-2023/Операционные системы/os-intro $ cd
dkbasmanova@dk3n63 ~ $ git commit -a -S -m 'your commit message'
fatal: не найден git репозиторий (или один из его каталогов вплоть до точки конфигурирования /)
Остановились поиск на границе файловой системы (так как GIT_DISCOVERY_ACROSS_FILESYSTEM не установлен).
dkbasmanova@dk3n63 ~ $ cd ~/work/study/2022-2023/Операционные системы/os-intro
dkbasmanova@dk3n63 ~/work/study/2022-2023/Операционные системы/os-intro $ git commit -a -S -m 'your commit message' error: gpg не удалось подписать данные
fatal: сбой записи объекта коммита
dkbasmanova@dk3n63 ~/work/study/2022-2023/Операционные системы/os-intro $ git commit -a -S -m 'your commit message'
error: gpg не удалось подписать данные
fatal: сбой записи объекта коммита
dkbasmanova@dk3n63 ~/work/study/2022-2023/Операционные системы/os-intro $ git commit -a -S -m 'your commit message' error: gpg не удалось подписать данные
fatal: сбой записи объекта коммита
dkbasmanova@dk3n63 ~/work/study/2022-2023/Операционные системы/os-intro $ git commit -a -m 'your commit message'
[master d98a2ff] your commit message
2 files changed, 1 insertion(+), 14 deletions(-)
delete mode 100644 package.json
dkbasmanova@dk3n63 ~/work/study/2022-2023/Операционные системы/os-intro $ git config --global user.signingkey <PGP Fingerprint>
bash: синтаксическая ошибка рядом с неожиданным маркером «newline»
dkbasmanova@dk3n63 ~/work/study/2022-2023/Операционные системы/os-intro $ git config --global user.signingkey
dkbasmanova@dk3n63 ~/work/study/2022-2023/Операционные системы/os-intro $ git config --global commit.gpgsign true
dkbasmanova@dk3n63 ~/work/study/2022-2023/Операционные системы/os-intro $ git config --global gpg.program $(which gpg2)
dkbasmanova@dk3n63 ~/work/study/2022-2023/Операционные системы/os-intro $
```

Рис. 3.6: 5

10) Отправим все сделанные файлы на свой GitHub.

4 Выводы

Я изучила идеологию и применение средств контроля версий и освоила умения по работе с git.

5 Ответы на контрольные вопросы.

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

Система контроля версий (VCS) – это система, регистрирующая изменения в одном или нескольких файлах. Например, дизайнер и хотели бы хранить каждую версию изображения или макета – а этого вам не хватает. Вы можете потерять файл, сделать ошибку, и многое другое. Вообще, если, пользуясь, вы всё испортите или потеряете файл.

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

Хранилище-система, которая обеспечивает хранение всех существовавших вариантов файла. Команда commit фиксирует изменения. История-список предыдущих ревизий. Рабочая копия-копия другой ветки. Командой commit можно передать сообщение, описывающее изменения, например, message. Можно также вводить сообщения, состоящие из нескольких строк; например, "добавлен первый файл".

3. Что представляют собой и чем отличаются централизованные и децентрализованные системы контроля версий.

Системы контроля версий. Централизованная система контроля версий Subversion и децентрализованная Git.

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

Традиционные системы управления версиями используют централизованную модель, когда все изменения хранятся в одном месте. Например, компрессия – такой способ хранения документов, при котором сохраняются только изменения.

5. Опишите порядок работы с общим хранилищем VCS.

Традиционные системы управления версиями используют централизованную модель, когда

6. Каковы основные задачи, решаемые инструментальным средством git?

Устанавливает единственную новую команду, git. Все возможности предоставляются че

Основные задачи: создание ветки, размещение веток, просмотр изменений, фиксация изменений, сообщение из текстового редактора, выборочная фиксация, удаление зафиксированных изменений, игнорирование файлов, просмотр истории, статистика ветки, контроль файлов и каталогов, ветвление, объединение веток, публикация ветки.

7. Назовите и дайте краткую характеристику командам git.

Обновление рабочей копии По мере внесения изменений в проект рабочая копия на ком

8. Приведите примеры использования при работе с локальным и удалённым репозитория

Мы создаем новую ветку выполнив git init в уже созданном каталоге: % mkdir tutorial
a ./ ../ % pwd /home/mbp/work/bzr.test/tutorial % % git init % ls -
aF ./ ../ .git/ % Мы обычно обращаемся к веткам на нашем компьютере просто переда
vcs.org git // git.dev/ git log sftp://bazaarvcs.org/bzr/bzr.dev/ Установив для g
01-01 00:00:00 +0000 +++ hello.txt 2005-10-18 14:23:29 +00006.2. Указания к лабор
0,0 +1,1 @@ +hello world Команде commit можно передать сообщение описывающее изме
m "добавлен первый файл" Если вы передадите список имен файлов, или каталогов пос
m "исправления документации" commit.py Если вы сделали какие-либо изменения и не
forward делает тоже самое, но в хронологическом порядке, показывая более поздние
r src % remove -v hello.txt ? hello.txt % status removed: hello.txt src/ src/simp

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

Часто вместо того что бы начинать свой собственный проект, вы хотите предложить и

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

Нет проблем если шаблон для игнорирования подходит для файла под контролем версий
м "Добавлены шаблоны для игнорирования". Многие деревья с исходным кодом содержат