

## 1 Использование системы контроля версий и CodeReview

### Требуемое программное обеспечение:

1. GIT

Ссылка на скачивание: <https://git-scm.com/downloads>

Установить в предложенную папку.

### Требуемое аппаратное обеспечение:

1. Относительно прямые руки.

### Дополнительные требования:

1. Почта ХАИ;

### 1.1 Регистрация в системе

Целевой сайт – <https://ci.csn.khai.edu/gerrit/>

Зайдите на него и зарегистрируйтесь, используя почту ХАИ.

На вкладке Settings - Profile установите Username и Full Name на вкладке Settings - Contact Information.

1. Откройте Git Bash (Вверху прописан путь: он должен быть следующим – `c/<пользователи>/<имя_пользователя>`<sup>1</sup>).
2. Настроим Git. Введите команды:

```
git config --global user.email "example@example.com"
git config --global user.name "example_user_name"
```

где `example@example.com` – ваша почта ХАИ,

`example_user_name` – Username, который вы выбрали в Gerrit.

При наборе команды `git config -l` в списке будут указаны выбранные вами почта и имя.

Настройка Git закончена.

3. Установка SSH Ключей в Gerrit

```
ssh-keygen -t rsa -C "your_email@youremail.com"
```

где `your_email@youremail.com` – ваша почта ХАИ;

После ввода команды появится запрос на выбор директории – просто нажмите ввод. Потом появится запрос на ввод секретной фразы (грубо говоря пароля) – введите его. Не беспокойтесь, что символы не появляются – так и должно быть. Каждый нажатый символ записывается и вносится в секретную фразу. Как закончите ввод пароля – нажмите ввод. После этого будет запрос на повтор секретной фразы – ну сами догадаетесь что сделать нужно.

Ура, мы сгенерировали ключ.

---

<sup>1</sup> Если вы поставили Git в другую папку автор не несет ответственности за ваши страдания и верит в ваши навыки.

```
preilly@vumi:/mw/examples$ ssh-keygen -t rsa -C "your_email@youremail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/preilly/.ssh/id_rsa):

Created directory '/home/preilly/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/preilly/.ssh/id_rsa.
Your public key has been saved in /home/preilly/.ssh/id_rsa.pub.
The key fingerprint is:
46:f0:61:f0:69:45:ad:a8:89:91:ca:f8:f3:c5:0a:36 your_email@youremail.com
The key's randomart image is:
+---[ RSA 2048]-----+
|      o.o.o.      |
|      = + .      |
|      . *. .      |
|      o o. .      |
| o . o oS        |
| . o ..o.        |
| .E  o           |
| .oo o           |
|      oo         |
+-----+
preilly@vumi:/mw/examples$
```

Рисунок 1 – Пример работы команды

Найдите два сгенерированных файла (id\_rsa и id\_rsa.pub) на диске. Папка, в которой они находятся, скрыта. Откройте id\_rsa.pub. Скопируете содержимое этого файла. Вернитесь на сайт в раздел Settings - SSH Public Keys и в окошко вставьте содержимое файла и нажмите Add.

Вернитесь в Git Bash.

Введите команды:

```
eval `ssh-agent`
ssh-add .ssh/id_rsa
```

Проверьте подключение:

```
ssh -p 29418 <username>@ci.csn.khai.edu
```

Если будет выведено сообщение, начинающиеся с Welcome to Gerrit Code Review – это ваш маленький успех на пути к заливанию лабы.

В случае неудачи обратитесь к вашему системному администратору (или самостоятельно читайте сайты, указанные в источниках).

4. Зайдите в папку .ssh и создайте в ней файл config (без расширения) и внесите в него следующие:

```
Host <ваше_имя_хоста>
    Hostname ci.csn.khai.edu
    Port 29418
    User <ваш_username>
```

Ваше имя хоста нужно придумать. Я буду использовать my\_gerrit как имя хоста в дальнейших командах.

## 1.2 Загрузка проекта

В инструкции указано как запустить тестовое сообщение в Sandbox (песочницу), которую можно найти по ссылке <https://ci.csn.khai.edu/gerrit/#/q/project:cicsnkhaidedu-dev/sandbox> . Все успешные попытки отображены в представленном списке.

1. Скопируем Sandbox на локальную машину:

```
git clone my_gerrit:cicsnkhaidedu-dev/sandbox
```

2. Перейдем в эту папку командой cd.

Если вы находитесь в нужной папке, рядом с ее названием в командной строке будет голубая надпись (master)

3. Ручками перенесите в эту папку нужный файл.

4. Пропишите команды:

```
git add <имя_файла>
```

```
scp -p -P 29418 <username>@ci.csn.khai.edu:hooks/commit-msg .git/hooks/
```

```
git commit --amend
```

Если в итоге одной из этих команд появится новое окно – пропишите wq и ввод (я не помню в каком случае оно появляется). Если окно не появилось помните, что вы не особенный.

5. Время загрузить изменения:

```
git push my_gerrit:cicsnkhaidedu-dev/sandbox HEAD:refs/for/master
```

В случае удачи по ссылке в начале этого раздела в списке коммитов будет и ваш.

Если вы хотите залить лабу – нужно копировать папку с вашей фамилией (она должна быть в списке Projects – List) вместо sandbox. При входе в свою папку можно найти ссылку для ее копирования.

## 2 Требования к проекту

Для лабораторных работ нужно загружать только vhd файлы (код работы и код тестов). При верной загрузке работы в поле V вашего коммита будет зеленая галочка. Это проверка кода встроенной системой – Jenkins. Добиться этого тяжело, так как система очень требовательна к форматированию кода. Лучший вариант – смотреть на ошибки, выданные компилятором Jenkins и постепенно исправлять код (следовательно, коммитить заново).

Известные требования к коду:

1. Каждый порт, сигнал, переменная пишутся на отдельной строке;
2. Символ табуляции недопустим;
3. Каждая секция должна быть закончена полной формой. Например:

```
entity laba_1 is
  port(
    x1 : in  std_logic;
    x2 : in  std_logic;
    x3 : in  std_logic;
    x4 : in  std_logic;
    x5 : in  std_logic;
    y1 : out std_logic
  );
end entity laba_1;
```

4. Каждый процесс должен иметь метку-имя. Например:

```
proc_1: process
  begin
    x1 <= not x1;
    wait for 50 ns;
  end process proc_1;
```

### **Список использованных источников**

1. <https://www.mediawiki.org/wiki/Gerrit/Tutorial>
2. <https://ci.csn.khai.edu/gerrit/Documentation/user-upload.html>