

Отчет по лабораторной работе №1: Система контроля версий Git

дисциплина: Математическое моделирование

Сасин Ярослав Игоревич, НФИбд-03-18

Содержание

Введение.....	1
Цель работы	1
Задачи работы.....	1
Объект и предмет исследования.....	2
Терминология. Условные обозначения	2
Выполнение лабораторной работы	2
Установка имени и электронной почты.....	2
Параметры установки окончаний строк.....	2
Установка отображения unicode	3
Создание SSH-ключа с последующей выгрузкой на GitHub.....	3
Создание каталога для последующей работы и его выгрузка на GitHub	4
Работа с git-flow.....	7
Выводы	8

Введение

Цель работы

Основной целью лабораторной работы можно считать ознакомление с системой контроля версий Git.

Задачи работы

Можно выделить три основные задачи данной лабораторной работы: 1. ознакомление с системой контроля версий Git; 2. создание каталогов для хранения данных последующих лабораторных работ с их выгрузкой на Github; 3. ознакомление с набором расширений git-flow.

Объект и предмет исследования

Объектом исследований для данной лабораторной работы является система контроля версий git, предметом же исследования можно считать непосредственное взаимодействие с git, его преимущества и особенности, а также использование платформы GitHub для размещения git-репозитория.

Терминология. Условные обозначения

git - это бесплатная распределенная система контроля версий с открытым исходным кодом, предназначенная для быстрого и эффективного управления всеми проектами любого размера.

GitHub - крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Веб-сервис основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc.

git-flow — это набор расширений git предоставляющий высокоуровневые операции над репозиторием для поддержки модели ветвления Vincent Driessen.

Homebrew — утилита командной строки в macOS и Linux, которая позволяет устанавливать пакеты и приложения (менеджер пакетов). Распространяется как свободное программное обеспечение с открытым кодом.

Выполнение лабораторной работы

Для того, чтобы использовать git на своем устройстве, для начала необходимо его установить. Так как в моем случае git уже был установлен, данный пункт работы был пропущен. Но, в случае, если бы на моем устройстве не был бы установлен данный пакет, я бы воспользовался менеджером пакетов Homebrew для установки:

```
brew install git
```

Установка имени и электронной почты

Для того, чтобы git узнал имя и адрес электронной почты пользователя, необходимо выполнить следующие команды(рис. @fig:001):

```
git config --global user.name ""  
git config --global user.email "1032182505@pf.ru"
```

Параметры установки окончаний строк

Настроил core.autocrlf с параметром input для того, чтобы все переводы строк текстовых файлов в главном репозитории одинаковы были одинаковы. Конвертация CRLF в LF будет производиться только при коммитах. При настроенном core.safecrlf в true, git будет проверять, если преобразование является обратимым

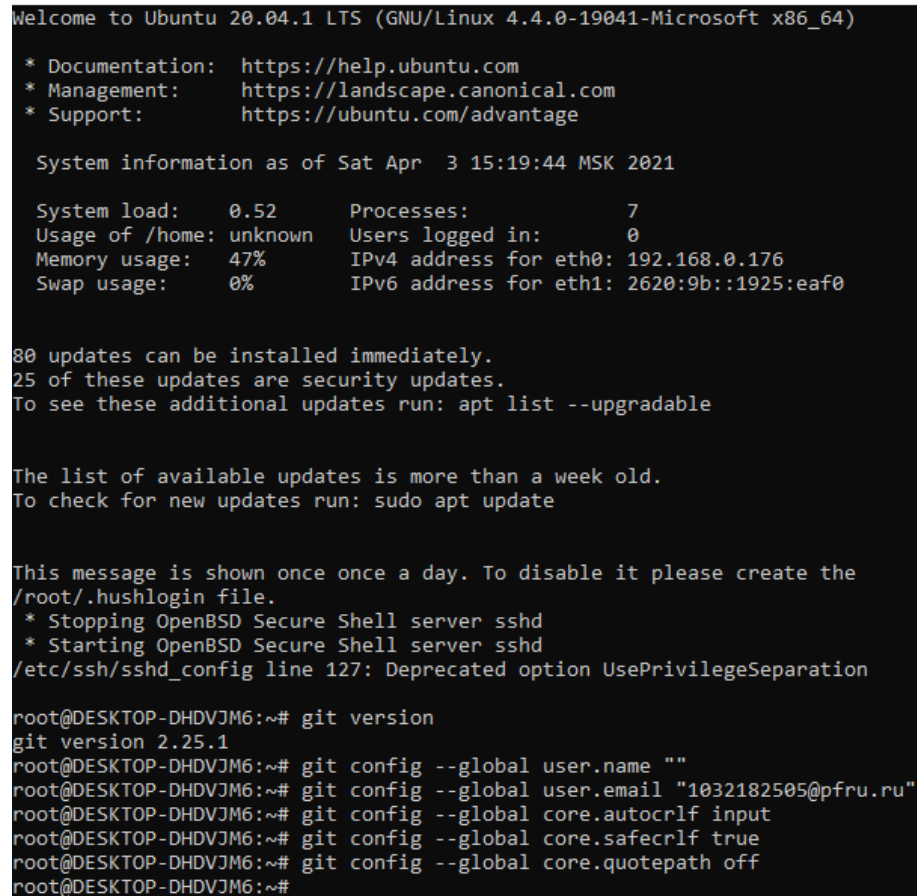
для текущей настройки core.autocrlf, то есть core.safecrlf true - отвержение необратимого преобразования lf<->crlf(рис. @fig:001):

```
git config --global core.autocrlf input
git config --global core.safecrlf true
```

Установка отображения unicode

Что бы избежать нечитаемых строк, установил соответствующий флаг (рис. @fig:001):

```
git config --global core.quotepath off
```



```
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 4.4.0-19041-Microsoft x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Sat Apr  3 15:19:44 MSK 2021

System load:  0.52      Processes:            7
Usage of /home: unknown  Users logged in:      0
Memory usage: 47%      IPv4 address for eth0: 192.168.0.176
Swap usage:   0%       IPv6 address for eth1: 2620:9b::1925:eaf0

80 updates can be installed immediately.
25 of these updates are security updates.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

This message is shown once once a day. To disable it please create the
/root/.hushlogin file.
* Stopping OpenBSD Secure Shell server sshd
* Starting OpenBSD Secure Shell server sshd
/etc/ssh/sshd_config line 127: Deprecated option UsePrivilegeSeparation

root@DESKTOP-DHDVJM6:~# git version
git version 2.25.1
root@DESKTOP-DHDVJM6:~# git config --global user.name ""
root@DESKTOP-DHDVJM6:~# git config --global user.email "1032182505@pf.ru"
root@DESKTOP-DHDVJM6:~# git config --global core.autocrlf input
root@DESKTOP-DHDVJM6:~# git config --global core.safecrlf true
root@DESKTOP-DHDVJM6:~# git config --global core.quotepath off
root@DESKTOP-DHDVJM6:~#
```

Установка имени и электронной почты и параметры установки окончаний строк

Создание SSH-ключа с последующей выгрузкой на GitHub

Так как у меня уже имеется учетная запись на GitHub, то мне просто нужно сгенерировать SSH-ключ и выгрузить его на GitHub.

Генерация ключа и его последующее “изыятие” из файла, в котором он записан, происходит при помощи следующих команд (рис. @fig:002):

```
ssh-keygen -t rsa
cat /Users//.ssh/Id-rsa.pub
```

```
root@DESKTOP-DH0VJME:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_rsa
Your public key has been saved in id_rsa.pub
The key fingerprint is:
SHA256:6Rkq14jP8MUI1g3W6W3k1AP1j8Nt020V11m33JHY root@DESKTOP-DH0VJME
The key's randomart image is:
+--[RSA 3872]-----+
|  .+ .+ .+          |
| o+o .o             |
| o.o .o             |
| .+ .S              |
| o.o + o.+          |
| o+ +oooo.E         |
| oG .+++.o          |
|_._+_.+_.+          |
+-----[SHA256]-----+
root@DESKTOP-DH0VJME:~# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQC9Xp4K8F5na/TBzWNfeh6gU2UGHbFujjOkJgqMe3M/E+Q4M5x28evtZ+
XFFvAhJkm75ZHAVBO/OG4XyAdBpxN1lgLIRBP+KrdGACJZ81gCwaV4pnlbAt2rlPxxWW7UQRp4S7OE4wt7ApdWt6Ye
KzdagiT0Z41xmwUUDg6j4Rbj5w/P+ZegruYfRDuIEoCTKlaTX9dYZMn+GnFmUxo59nFclh6aj4h8sdBoKb/naJ9FxRTgXh
p8Dz2FjM6e1Qp0JDxHPRL2SOnr8XjYnPMxSD6dedFuxudg9RGf3TyNIGVQpWnE2w8SKw8x8jVgd/ztblcSwgOqISpG
NnFUniyuZrSGjyax8rpQHCPc1xMeodi6HN7p+E7rVuXvOcAiwKip0D6EYRH3D1R5ZC1V/WvgJQ5rth9/V4s2yHNPMN
L4kRYX0Z/OFEK3FT+DvRC1p69j8vQdijUR1QfXM0h4ErZtzYqjcd2DCIV8qT7KHxewWF6OcP4tTLDymU=
root@DESKTOP-DH0VJME:~#
```

Генерация и чтение SSH-ключа

Далее в настройках профиля на GitHub, во вкладке “SSH and GPG keys” необходимо добавить новый SSH-ключ. Это необходимо для быстрого доступа к репозиториям, которые созданы в моей учетной записи (рис. @fig:003).



Добавление SSH-ключа на GitHub

Создание каталога для последующей работы и его выгрузка на GitHub

Для дальнейшей работы с лабораторными работами и проектами необходимо создать директорию с определенной иерархией, которая была предложена ранее. Также была добавлена папка для файлов первой (данной) лабораторной работы (рис. @fig:004):

```
cd Documents
mkdir work
```

```
mkdir work/laboratory
mkdir work/laboratory/lab01
```

```
root@DESKTOP-DHDVJM6:~# mkdir work
root@DESKTOP-DHDVJM6:~# mkdir work/laboratory
root@DESKTOP-DHDVJM6:~# mkdir work/laboratory/lab01
root@DESKTOP-DHDVJM6:~#
```

Создание директории

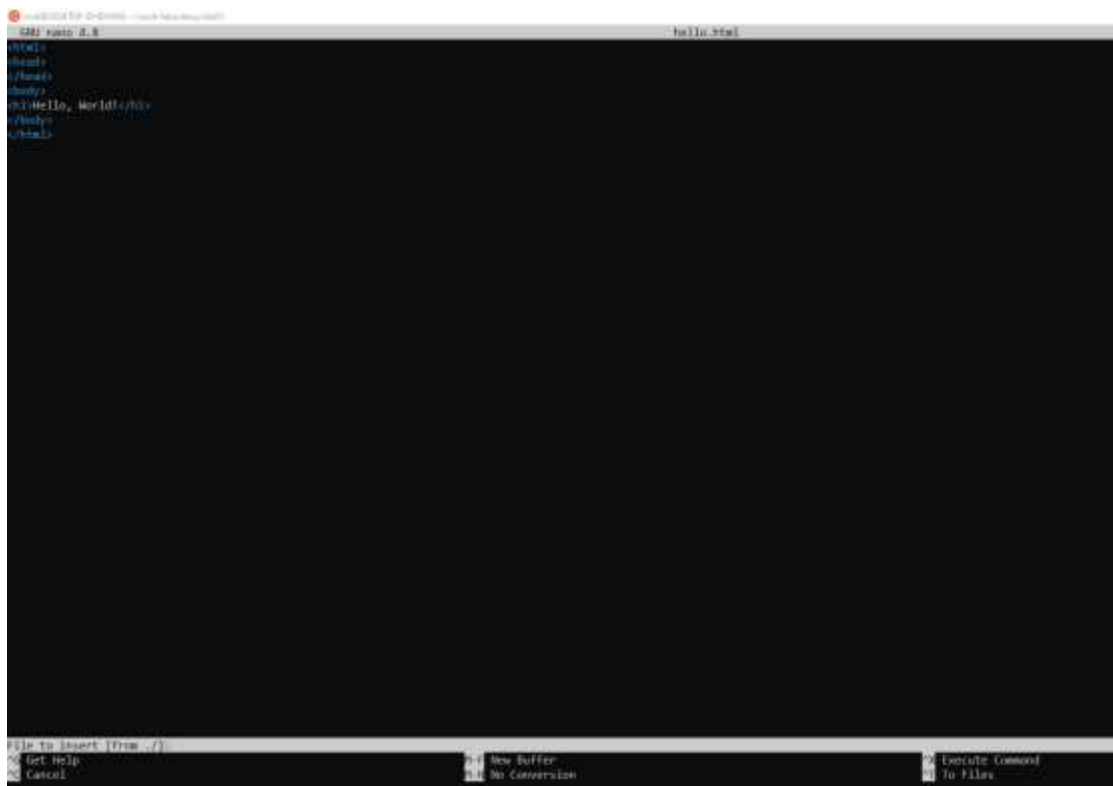
Далее необходимо создать новый репозиторий в GitHub.

Для того, чтобы не инициализировать пустой репозиторий, добавим следующий файл с текстом “Hello, World!” (рис. @fig:005, рис. @fig:006):

```
cd work/laboratory/lab01
mkdir hello
touch hello.html
nano hello.html
```

```
root@DESKTOP-DHDVJM6:~# cd work/laboratory/lab01
root@DESKTOP-DHDVJM6:~/work/laboratory/lab01# mkdir hello
root@DESKTOP-DHDVJM6:~/work/laboratory/lab01# touch hello.html
root@DESKTOP-DHDVJM6:~/work/laboratory/lab01# nano hello.html
root@DESKTOP-DHDVJM6:~/work/laboratory/lab01#
```

Добавление папки hello и файла hello.html



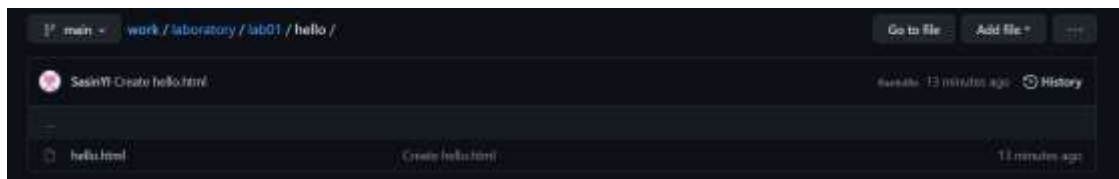
Изменение файла hello.html

Далее необходимо было выполнить. 1. создание репозитория git ('git init');

2. добавление всех созданных файлов в репозиторий('git add .');
3. коммит этого этапа ('git commit -m "Initial Commit"');
4. проверка текущего состояния репозитория ('git status');
5. "привязка" репозитория на GitHub к текущему репозиторию;
6. отправка изменений ветки "master" ('git push -u origin master').

```
cd ~/Documents/work
git init
git add .
git commit -m "Initial Commit"
git status
git remote add origin git@github.com:/work.git
git push -u origin master
```

Можно увидеть, что в репозитории на GitHub появились данные, которые были добавлены при помощи git (рис. @fig:007):



Репозиторий на GitHub

Работа с git-flow

В данной лабораторной работе взаимодействие с git-flow осуществляется посредством работы над отчетом и презентацией. Следовательно, добавить в отчет можно лишь часть процесса для более эффективной работы.

Первым делом необходимо скачать git-flow:

```
brew install git-flow
```

Далее необходимо инициализировать git-flow в директории work:

```
git flow init
```

После этого мы окажемся на ветке “develop”. Для того, чтобы процесс создания отчета был максимально независимым от работы над другими частями лабораторной работы, необходимо переключиться из ветки “develop” на ветку фич:

```
git flow feature start MYFEATURE
```

При необходимости можно сохранять и публиковать промежуточные версии при помощи команд:

```
git flow feature publish MYFEATURE
```

После окончания работы над отчетом производится слияние ветки фич с веткой разработки и удаление ветки фич:

```
git flow feature finish MYFEATURE
```

После окончания работы над фичами необходимо сделать релиз работы. Для этого необходимо создать ветку релиза, ответвля от ветки develop:

```
git flow release start RELEASE
```

Публикация релиза выполняется с помощью команды:

```
git flow release publish RELEASE
```

Завершить релиз (ветка релиз сливается с веткой master и в ветку develop и удаляется) можно с помощью команды

```
git flow release finish RELEASE
```

Выводы

В ходе выполнения лабораторной работы были изучены основные особенности системы контроля версий git, а также специфика выполнения и оформления лабораторных работ для данной дисциплины.