Лабораторная работа №2

Первоначальная настройка git

Емельянов Антон

Содержание

# 1. Цель работы

Изучить идеологию и применение средств контроля версий. Освоить умения по работе с git..

# 2. Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

# 3. Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в [табл. 1](#tbl-std-dir) приведено краткое описание стандартных каталогов Unix.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Таблица 1: Описание некоторых каталогов файловой системы GNU Linux   | Имя каталога | Описание каталога | | --- | --- | | / | Корневая директория, содержащая всю файловую | | /bin | Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям | | /etc | Общесистемные конфигурационные файлы и файлы конфигурации установленных программ | | /home | Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя | | /media | Точки монтирования для сменных носителей | | /root | Домашняя директория пользователя root | | /tmp | Временные файлы | | /usr | Вторичная иерархия для данных пользователя | |

Более подробно про Unix см. в [1–4].

# 4. Выполнение лабораторной работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown. Переместимся в каталог 2 лабы Откроем файл report.qmd и начнём исправлять его, превращая его в отчёт по лабе 2 используем make, чтобы создать pdf и docx Затем загружаем файлы на github В ходе работы с языком разметки markdown были получены базовые навыки редактуры и обращения с отчётами Лабораторная работа №2 Первоначальная настройка git Изучить идеологию и применение средств контроля версий. Освоить умения по работе с git. Начнём установку ПО с git ([рис. 1](#fig-001))

|  |
| --- |
| Рисунок 1: Установка git прошла успешно. |

Далее установим gh, а также внесём всю базовую информацию: имя глобального пользователя, адрес электронной почты, настроим utf-8 в выводе сообщений git. Зададим имя начальной ветки (master) и установим параметры autocrlf и safecrlf ([рис. 2](#fig-002))

|  |
| --- |
| Рисунок 2: Установка gh прошла успешно. |

([рис. 3](#fig-003))

|  |
| --- |
| Рисунок 3: Внесли все базовые параметры. |

Сгенерируем ключи размера 4096 по алгоритму RSA и по алгоритму ed25519, также сгенерируем pgp ключ, он понадобится для дальней привязки github. Заодно не забудем зарегистрироваться на github (у меня уже была регистрация, но в данном случае это почти ничего не меняет, так как делаю я с новой виртуальной машины). ([рис. 4](#fig-004))

|  |
| --- |
| Рисунок 4: SSH ключи сгенерированы по алгоритмам RSA и ed25519. |

([рис. 5](#fig-005))

|  |
| --- |
| Рисунок 5: Предложили защитить всё паролем. |

([рис. 6](#fig-006))

|  |
| --- |
| Рисунок 6: Сгенерирован pgp ключ. |

([рис. 7](#fig-007))

|  |
| --- |
| Рисунок 7: Github “Заведён” и готов к использованию. |

Теперь добавили ключ на Github, таким образом подключили систему к Github.

([рис. 8](#fig-008))

|  |
| --- |
| Рисунок 8: Ключик добавлен. |

Настроим автоматические подписи коммитов Git, заставим Git использовать мой email для подписи коммитов.

([рис. 9](#fig-009))

|  |
| --- |
| Рисунок 9: Теперь должен правильно подписывать commit-ы. |

Произведём авторизацию и создадим шаблон рабочего пространства, для этого создадим папку и в ней образуем репозиторий, скопируем репозиторий и, перейдя в каталог курса, удалим всё лишнее и создадим каталоги.

([рис. 10](#fig-010))

|  |
| --- |
| Рисунок 10: Подрубаем github. |

([рис. 11](#fig-011))

|  |
| --- |
| Рисунок 11: Создаём репозиторий курса. |

([рис. 12](#fig-012))

|  |
| --- |
| Рисунок 12: Копируем всё и кое-что удаляем. |

([рис. 13](#fig-013))

|  |
| --- |
| Рисунок 13: ереходим в каталог курса и готовимся к финальному рывку. |

Затем отправим файлы на сервер.

([рис. 14](#fig-014))

|  |
| --- |
| Рисунок 14: Подтверждаем и отправляем всё. Изменения отразились, УРА. |

# 5. Выводы

В этой лабораторной работе мы успешно создали и настроили систему контроля версий Git, её окружение и организовали её взаимодействие с Github.

Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются? VCS используется для контроля изменения файлов во времени, их основная задача - контроль версий, сравнение, возврат, совместная работа, ветвление и резервное копирование Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. Хранилище = все версии проекта, а также история их изменений Commit = снимок состояния проекта в момент времени История = последовательность Commit-ов

# Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.

2. Robbins A. Bash Pocket Reference. O’Reilly Media, 2016. 156 с.

3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.

4. Newham C. [Learning the bash Shell: Unix Shell Programming](http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658). O’Reilly Media, 2005. 354 с.