
Front matter

lang: ru-RU title: "ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2" subtitle: "Операционные системы" author: "Дмитриев Александр Дмитриевич"

Formatting

toc-title: "Содержание" toc: true # Table of contents toc_depth: 2 lof: true # List of figures lot: true # List of tables fontsize: 12pt linestretch: 1.5 papersize: a4paper documentclass: scrreprt polyglossia-lang: russian polyglossia-otherlangs: english mainfont: PT Serif romanfont: PT Serif sansfont: PT Sans monofont: PT Mono mainfontoptions: Ligatures=TeX romanfontoptions: Ligatures=TeX sansfontoptions: Ligatures=TeX,Scale=MatchLowercase monofontoptions: Scale=MatchLowercase indent: true pdf-engine: lualatex header-includes:

- `\linepenalty=10` # the penalty added to the badness of each line within a paragraph (no associated penalty node) Increasing the value makes tex try to have fewer lines in the paragraph.
 - `\interlinepenalty=0` # value of the penalty (node) added after each line of a paragraph.
 - `\hyphenpenalty=50` # the penalty for line breaking at an automatically inserted hyphen
 - `\exhyphenpenalty=50` # the penalty for line breaking at an explicit hyphen
 - `\binoppenalty=700` # the penalty for breaking a line at a binary operator
 - `\relpenalty=500` # the penalty for breaking a line at a relation
 - `\clubpenalty=150` # extra penalty for breaking after first line of a paragraph
 - `\widowpenalty=150` # extra penalty for breaking before last line of a paragraph
 - `\displaywidowpenalty=50` # extra penalty for breaking before last line before a display math
 - `\brokenpenalty=100` # extra penalty for page breaking after a hyphenated line
 - `\predisdisplaypenalty=10000` # penalty for breaking before a display
 - `\postdisplaypenalty=0` # penalty for breaking after a display
 - `\floatingpenalty = 20000` # penalty for splitting an insertion (can only be split footnote in standard LaTeX)
 - `\raggedbottom` # or `\flushbottom`
 - `\usepackage{float}` # keep figures where there are in the text
 - `\floatplacement{figure}{H}` # keep figures where there are in the text
-

Цель работы

Целью данной работы является изучение идеологии и применение средств контроля версий.

Выполнение лабораторной работы

1. В первую очередь, создадим учетную запись на <https://github.com>.
(<https://github.com/addmitriev66/lab03/blob/main/snapshots.lab02/Снимок%20экрана%202021-05-04%20в%2015.20.37.png>)
2. Настроим систему контроля версий git с использованием сервера репозитория: Создадим локальный репозиторий. Сначала сделаем предварительную конфигурацию, указав имя и email владельца репозитория

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей, используя команду: `ssh-keygen -C "Имя Фамилия work@mail"`

(<https://github.com/addmitriev66/lab03/blob/main/snapshots.lab02/Снимок%20экрана%202021-05-04%20в%2015.25.00.png>)

С помощью команды `cat ~/.ssh/id_rsa.pub | xclip -sel clip` скопируем из локальной консоли ключ в буфер обмена и вставляем ключ в появившееся на сайте поле. Для этого заходим в настройки, выбираем пункт «SSH and GPG keys», далее нажимаем «New SSH key» и вставляем ключ в раздел «key»

(<https://github.com/addmitriev66/lab03/blob/main/snapshots.lab02/Снимок%20экрана%202021-05-04%20в%2015.25.07.png>)

В результате создается ключ

3. Создаем репозиторий
(<https://github.com/addmitriev66/lab03/blob/main/snapshots.lab02/Снимок%20экрана%202021-05-04%20в%2015.27.57.png>)

Создаем каталог work и переходим в него. Создаем каталог 2020-2021 и переходим в него, далее создаем каталог «laboratory» и тоже переходим в него
(<https://github.com/addmitriev66/lab03/blob/main/snapshots.lab02/Снимок%20экрана%202021-05-04%20в%2015.26.59.png>)

Делаем первый коммит и выкладываем его на гитхаб
(<https://github.com/addmitriev66/lab03/blob/main/snapshots.lab02/Снимок%20экрана%202021-05-04%20в%2015.29.02.png>)

Затем повторяем команду `git commit -m "first commit"` Используя команду `git push`, отправим все произведенные изменения локального дерева в центральный

репозиторий.

(<https://github.com/addmitriev66/lab03/blob/main/snapshots.lab02/Снимок%20экрана%202021-05-04%20в%2016.33.54.png>)

4. Первичная конфигурация. Добавим файл. Добавляем шаблон игнорируемых файлов. Сначала посмотрим список имеющихся шаблонов
(<https://github.com/addmitriev66/lab03/blob/main/snapshots.lab02/Снимок%20экрана%202021-05-04%20в%2015.57.03.png>)
(<https://github.com/addmitriev66/lab03/blob/main/snapshots.lab02/Снимок%20экрана%202021-05-04%20в%2015.57.09.png>)

Затем добавляем новые файлы, используя команду `git add`. Выполняем коммит `git commit -m "first commit"` и отправляем на гитхаб `git push`.

(<https://github.com/addmitriev66/lab03/blob/main/snapshots.lab02/Снимок%20экрана%202021-05-04%20в%2015.58.29.png>)

5. Конфигурация `git-flow` Для начала инициализируем `git-flow` `Git flow init -f`
Префикс для ярлыков устанавливаем в `v`.
(<https://github.com/addmitriev66/lab03/blob/main/snapshots.lab02/Снимок%20экрана%202021-05-04%20в%2015.59.42.png>)

Проверяем, что мы находимся на ветке `develop`, командой `git branch` Создаем релиз с версией 1.0.0 `Git flow release start 1.0.0`

(<https://github.com/addmitriev66/lab03/blob/main/snapshots.lab02/Снимок%20экрана%202021-05-04%20в%2016.00.50.png>)

(<https://github.com/addmitriev66/lab03/blob/main/snapshots.lab02/Снимок%20экрана%202021-05-04%20в%2016.16.06.png>)

Заливаем релизную ветку в основную ветку, используя команду `git flow release finish 1.0.0`

(<https://github.com/addmitriev66/lab03/blob/main/snapshots.lab02/Снимок%20экрана%202021-05-04%20в%2016.16.19.png>)

Отправим данные на github `git push --all git push --tags`

(<https://github.com/addmitriev66/lab03/blob/main/snapshots.lab02/Снимок%20экрана%202021-05-04%20в%2016.17.09.png>)

Создаем релиз на гитхаб.

Выводы

Мы изучили идеологии и применение средств контроля версий.