Лабораторная работа №4

Дисциплина: Операционные системы

Шония Ника Гигловна

Содержание

# Цель работы

Получение навыков правильной работы с репозиториями git.

# Задание

1. Установка git-flow
2. Установка Node.js
3. Настройка Node.js
4. Общепринятые коммиты

# Теоретическое введение

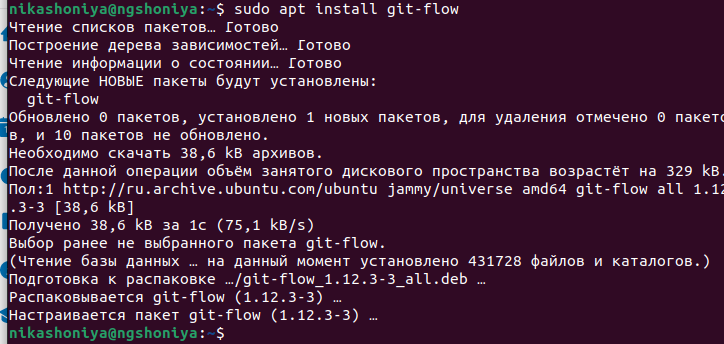
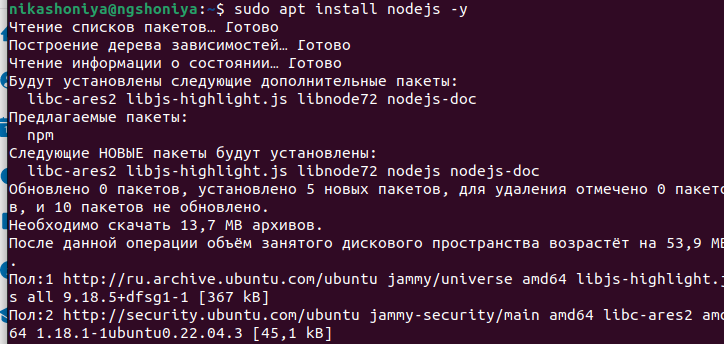
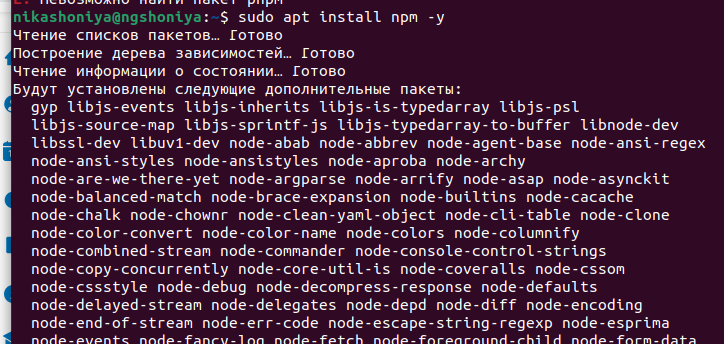
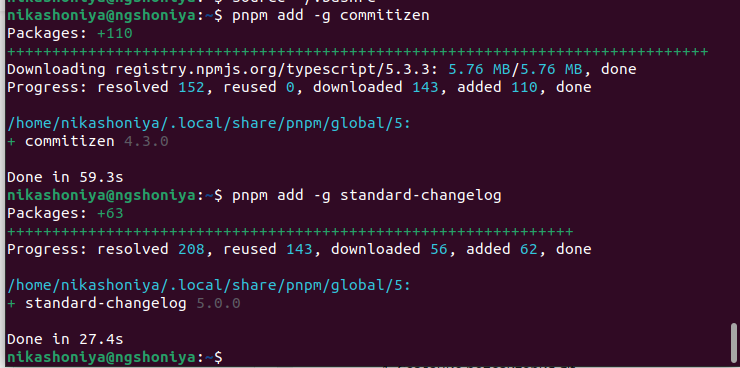
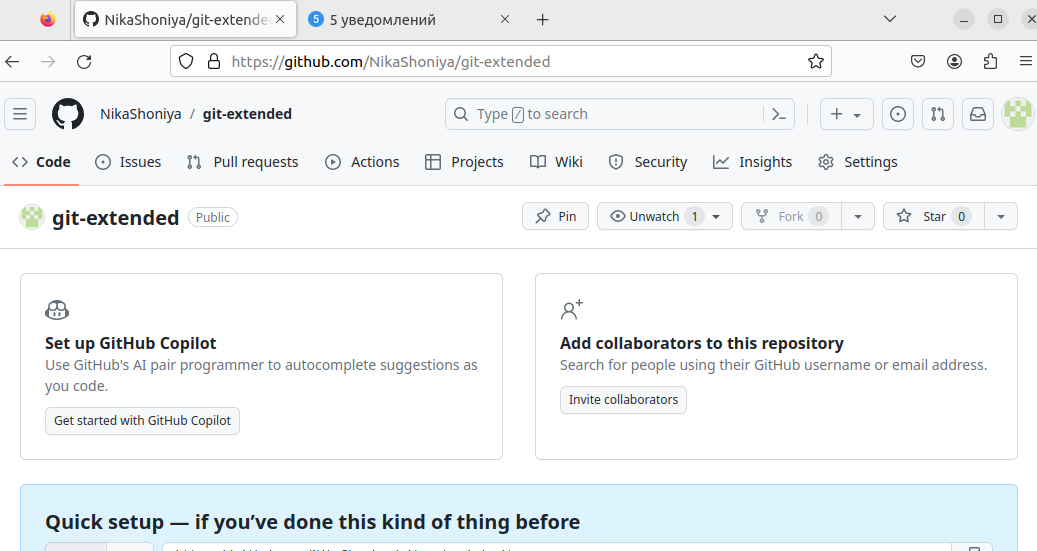
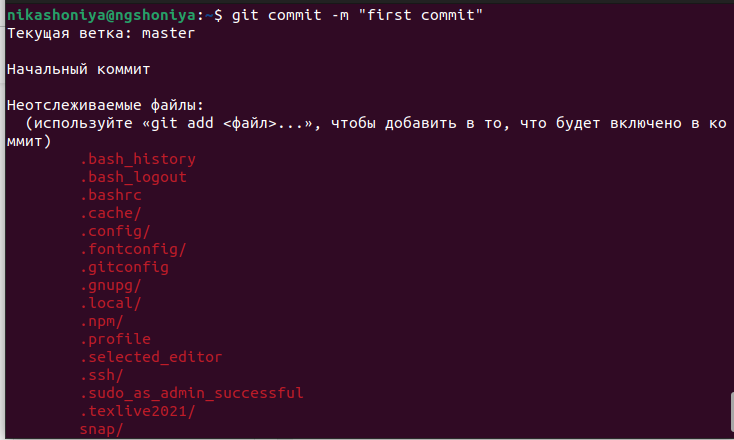
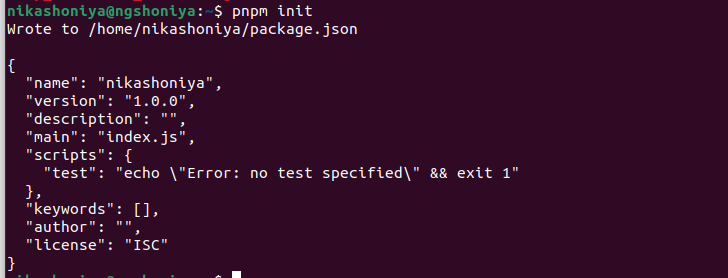
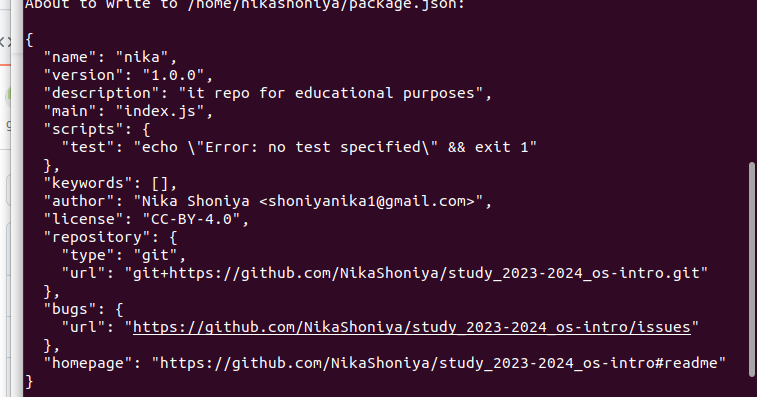
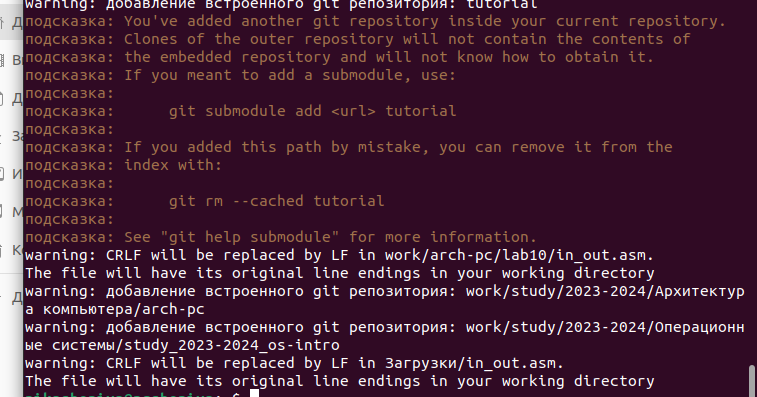
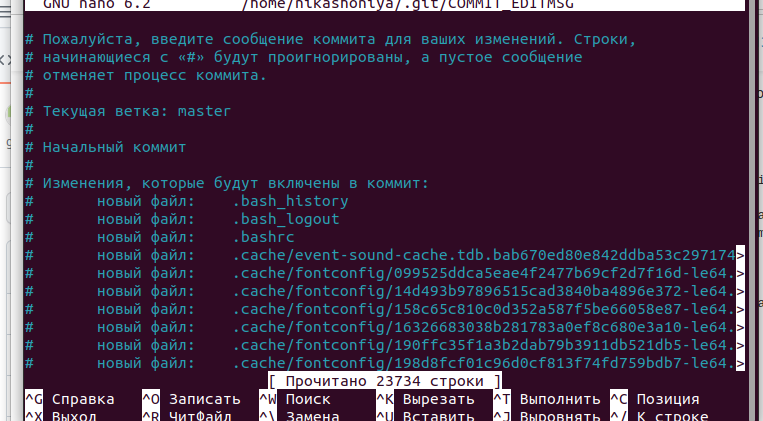
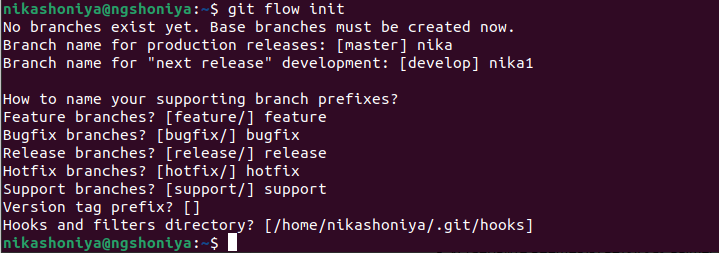
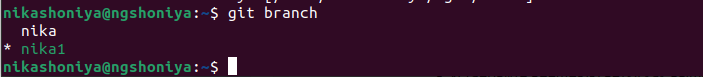
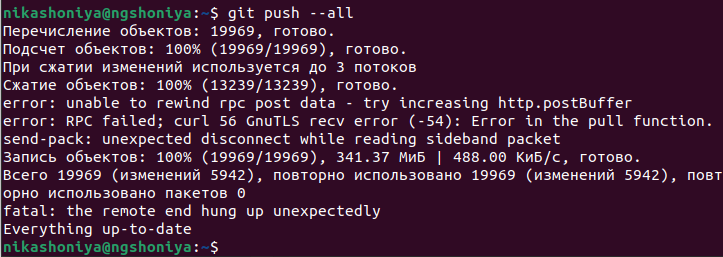
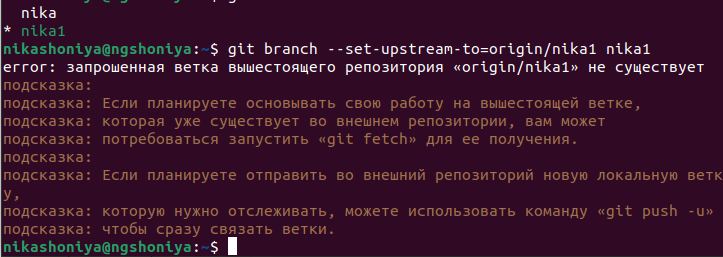
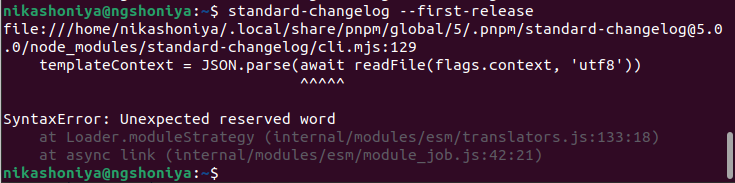
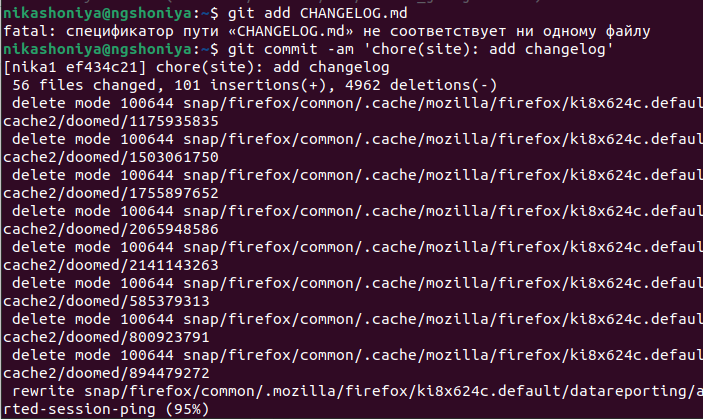
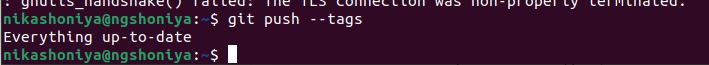
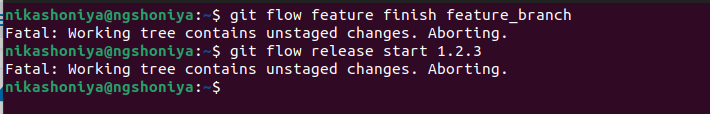
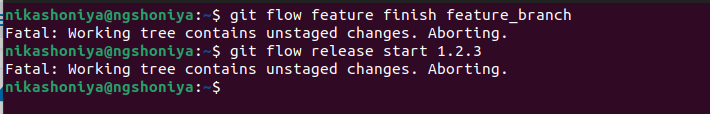
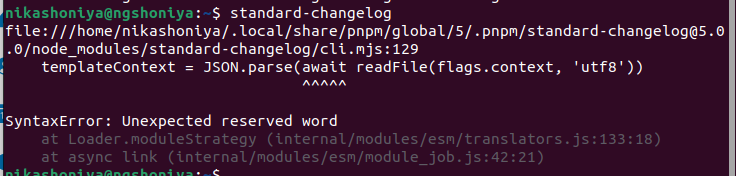
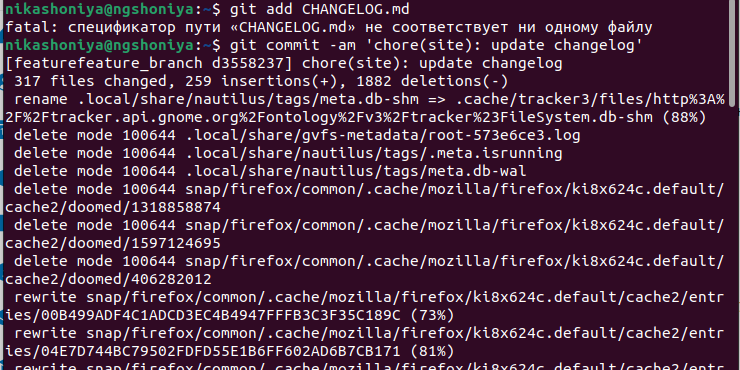
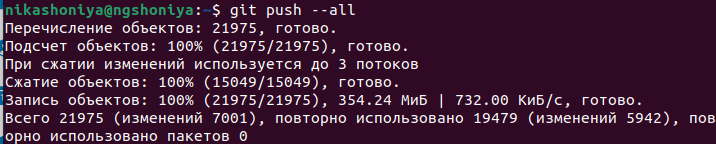
Рабочий процесс Gitflow

Рабочий процесс Gitflow Workflow. Будем описывать его с использованием пакета git-flow.

Общая информация

Gitflow Workflow опубликована и популяризована Винсентом Дриссеном.  
Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом выпуска проекта.  
Данная модель отлично подходит для организации рабочего процесса на основе релизов.  
Работа по модели Gitflow включает создание отдельной ветки для исправлений ошибок в рабочей среде.  
Последовательность действий при работе по модели Gitflow:  
 Из ветки master создаётся ветка develop.  
 Из ветки develop создаётся ветка release.  
 Из ветки develop создаются ветки feature.  
 Когда работа над веткой feature завершена, она сливается с веткой develop.  
 Когда работа над веткой релиза release завершена, она сливается в ветки develop и master.  
 Если в master обнаружена проблема, из master создаётся ветка hotfix.  
 Когда работа над веткой исправления hotfix завершена, она сливается в ветки develop и master.

# Выполнение лабораторной работы

1. Установка git-flow Устанавливаем git-flow 
2. Установка Node.js Установливаем Node.js 
3. Настройка Node.js Запускаю pnpm setup  выполняю source ~/.bashrc выполняю source ~/.bashrc
4. Общепринятые коммиты Данная программа используется для помощи в форматировании коммитов.  Создаю репозиторий  Делаем первый коммит и выкладываем на github (image/07.png){#fig:001 width=70%} Конфигурация для пакетов Node.js  Необходимо заполнить несколько параметров пакета.  Добавим новые файлы  Выполним коммит  Отправим на github Отправим на github Инициализируем git-flow  Проверьте, что Вы на ветке develop  Загрузите весь репозиторий в хранилище  Установите внешнюю ветку как вышестоящую для этой ветки:  Создадим релиз с версией 1.0.0 Создадим релиз с версией 1.0.0 Создадим журнал изменений  Добавим журнал изменений в индекс  Отправим данные на github  (image/20.png){#fig:001 width=70%} Создадим релиз на github. Для этого будем использовать утилиты работы с github Создадим релиз на github. Для этого будем использовать утилиты работы с github Создадим ветку для новой функциональности Создадим ветку для новой функциональности По окончании разработки новой функциональности следующим шагом следует объединить ветку feature\_branch c develop  Создадим релиз с версией 1.2.3  Создадим журнал изменений  Добавим журнал изменений в индекс  Зальём релизную ветку в основную ветку Зальём релизную ветку в основную ветку Отправим данные на github (image/30.png){#fig:001 width=70%} Создадим релиз на github с комментарием из журнала изменений Создадим релиз на github с комментарием из журнала изменений

# Выводы

Я получила навыки правильной работы с репозиториями git.

# Список литературы