

Информационные технологии (часть 2)

1. Требования к оформлению отчета

Отчет должен быть оформлен по СТП. Все рисунки должны быть читабельными. На каждом рисунке должна быть видна ФАМИЛИЯ СТУДЕНТА. **Обязательно выделяем команды на рисунках в терминале!**

2. Pull Request (часть 2)

Введение

Целью данной лабораторной работы является изучение GitHub.

Составить список используемых команд в терминале Linux

Команда	Значение	Расшифровать ключи

Теория

Форкнуть репозиторий означает создать копию существующего репозитория на **GitHub** под своим аккаунтом. Копия репозитория будет содержать все файлы, коммиты, ветки и историю изменений изначального репозитория. Форк позволяет вам свободно вносить изменения в копию репозитория, не затрагивая оригинальный репозиторий.

Задание к работе

Представим, что вы работаете на проекте. У вас имеется 2 репозитория на разных аккаунтах **GitHub**, первый (**main**) где находятся рабочие версии приложения. А второй репозиторий (**future**) для разработки новых фич. Также это можно проделать и на одном репозитории, но имея 2 разных ветки.

1) Создайте программу на языке C, которая складывает два числа. Ниже приведен пример кода на языке C:

```
#include <stdio.h>

int main() {
    int a = 5;
    int b = 3;
    int sum = a + b;
    printf("Сумма: %d\n", sum);
    return 0;
}
```

2) Создайте новый репозиторий на **GitHub** с названием «main» и загрузите программу в этот репозиторий. Добавьте коммит с пометкой о версии программы. Вы можете использовать следующие команды для выполнения этого:

Инициализация репозитория

git init

Добавление файлов программы

git add .

Создание коммита с пометкой о версии

git commit -m "Версия 1.0"

Связывание репозитория с удаленным репозиторием GitHub

git remote add origin <ссылка на ваш репозиторий GitHub>

Загрузка кода на GitHub

git push -u origin master

3) Зайдите на GitHub и форкните репозиторий «main» в свой аккаунт. Это создаст копию репозитория под названием «future» на вашем аккаунте. Вы можете клонировать форкнутый репозиторий на свой компьютер и работать с ним через консоль. Если вы используете один аккаунт GitHub, вы можете создать новую ветку в репозитории «main» вместо форка. Вот команды для форка репозитория:

Форк репозитория на GitHub

Зайдите на репозиторий "main" и нажмите кнопку "Fork"

Клонирование форкнутого репозитория на локальную машину

git clone <ссылка на ваш форкнутый репозиторий>

4) В форкнутом репозитории «future» добавьте новую фичу для программы, которая позволяет вычитать два числа. Отредактируйте программу следующим образом:

```
#include <stdio.h>

int main() {
    int a = 5;
    int b = 3;
    int sum = a + b;
    int difference = a - b;
    printf("Сумма: %d\n", sum);
    printf("Разность: %d\n", difference);
    return 0;
}
```

После внесения изменений, загрузите код с новой фичей на репозиторий «future» и сделайте коммит с описанием вашей фичи:

Добавление изменений

`git add .`

Создание коммита с описанием фичи

`git commit -m "Добавлена возможность вычитания чисел"`

Загрузка изменений на GitHub

`git push origin master`

5) Затем на **GitHub** перейдите на репозиторий «**main**» и нажмите кнопку «**Pull Request**». При создании **Pull Request** выберите репозиторий «**main**» и ветку, на которой находится конечная версия программы. В качестве базовой ветки выберите ветку «**master**» или главную ветку «**main**». В поле «Сравниваемая ветка» выберите репозиторий «**future**» и ветку, на которой находится ваша новая фича. Добавьте комментарий о фиче и нажмите «**Создать Pull Request**».

6) Затем зайдите на репозиторий «**main**» и примите созданный **Pull Request**. Перед принятием изменений вы можете просмотреть, что будет добавлено, а также оставить комментарий создателю **Pull Request**. Если вносятся изменения в одном репозитории с двумя ветками, вы можете просто выполнить слияние веток вместо создания **Pull Request**.

Вот примеры команд для принятия **Pull Request** на репозитории «**main**»:

Переключение на ветку main

`git checkout main`

Получение изменений с удаленного репозитория

`git pull origin main`

Просмотр списка Pull Request-ов

`git pull-request`

Принятие, Pull Request-а с идентификатором <идентификатор Pull Request-а>

`git pull-request accept <идентификатор Pull Request-а>`