# 运维项目2实验报告

---

# 实验报告：版本控制与 GitHub 使用

## 一、实验目的

- 学习和掌握 Git 和 GitHub 的基本使用，包括版本控制、分支管理和协作开发。

- 通过将个人项目上传到 GitHub，了解如何在 GitHub 上创建、管理分支，并实现 Pull Request 流程。

- 掌握团队协作开发的基本流程，学会如何通过 Pull Request 进行代码审查和合并。

## 二、实验内容

1. 将项目上传到 GitHub：

- 在 GitHub 上创建一个新的仓库（命名为 `empty`）。

- 初始化本地 Git 仓库并将其连接到 GitHub 仓库。

- 提交初始代码并推送到远程仓库。

2. 分支管理：

- 在本地创建新分支进行开发，修改项目内容并提交更改。

- 将本地分支推送到 GitHub。

- 创建 Pull Request 来提交分支的更改。

- 进行代码审查并合并 Pull Request。

3. 迭代方向：实现 Git 的 Pull Request 流程：

- 通过实际操作掌握如何创建、管理分支，并通过 Pull Request 流程实现协作开发。

## 三、实验步骤

1. 创建 GitHub 仓库：

- 在 GitHub 上创建一个名为 `empty` 的新仓库，选择公开或私有仓库，点击 Create repository。

2. 初始化本地 Git 仓库并连接 GitHub：

- 在 Ubuntu 虚拟机的终端中，创建一个新的文件夹并初始化 Git 仓库：

```bash

mkdir empty

cd empty

git init

```

- 创建一个简单的 `index.html` 文件并提交到本地 Git 仓库：

```bash

echo "<!DOCTYPE html><html><head><title>Empty Project</title></head><body><h1>Welcome to the Empty Project</h1></body></html>" > index.html

git add .

git commit -m "Initial commit: Add index.html"

```

- 将本地仓库连接到 GitHub 仓库并推送初始代码：

```bash

git remote add origin https://github.com/Amy-by/empty.git

git push -u origin main

```

3. 分支管理：

- 创建一个新分支 `feature-update-title`，修改项目内容并提交：

```bash

git checkout -b feature-update-title

echo "<!DOCTYPE html><html><head><title>Updated Empty Project</title></head><body><h1>Updated Content</h1></body></html>" > index.html

git add .

git commit -m "Update title and content"

git push origin feature-update-title

```

4. 创建 Pull Request：

- 在 GitHub 上，点击 Pull requests 标签，选择 New Pull Request。

- 设置 base 分支为 `main`，compare 分支为 `feature-update-title`，并点击 Create pull request。

- 在 Pull Request 页面，填写描述并提交。

5. 合并 Pull Request：

- 点击 Merge pull request 按钮来合并更改。

- 如果需要，点击 Delete branch 来删除已合并的分支。

6. 删除本地和远程分支（可选）：

- 删除本地分支：

```bash

git branch -d feature-update-title

```

- 删除远程分支：

```bash

git push origin --delete feature-update-title

```

7. 同步主分支（main）：

- 保持本地的 `main` 分支与远程同步：

```bash

git checkout main

git pull origin main

```

## 四、实验出现问题及解决方法

1. 问题：无法推送到 GitHub 仓库

- 原因：本地与远程仓库的关联可能存在问题。

- 解决方法：检查远程仓库的 URL 是否正确，使用命令 `git remote -v` 检查远程仓库设置。如果发现问题，可以重新设置远程仓库：

```bash

git remote set-url origin https://github.com/Amy-by/empty.git

```

---

## 五、实验总结

通过这次实验，我掌握了 Git 和 GitHub 的基本操作流程，包括版本控制、分支管理以及使用 Pull Request 流程进行协作开发。通过创建分支、提交代码和进行代码审查，我更加了解了 GitHub 的协作开发功能，并学会了如何管理和合并分支。未来在团队开发中，我将能够更加高效地使用这些工具进行版本管理和代码协作。

---

如果你有任何问题或需要更进一步的帮助，随时告诉我！