

**工控软件基础第二次实验报告**

项目名称：GitHub基础操作

实验时间：第2周，星期5第5~8节

专业班级：测控技术与仪器2101班

实验人姓名：叶木昱

实验人学号：202113650

|  |
| --- |
| **一、实验目的**  1.认识Git仓库三大区域；  2.进行一次完整的修改、提交、推送操作；  3.进行一次版本回退；  4.处理 commit 时间线分叉  5.进行本地仓库 commit 变化记录。 |
| **二、实验设备（或软件平台）**  1．Windows笔记本  2．Git软件  3．GitHub网站 |
| **三、实验内容及步骤**  1.认识Git仓库三大区域；  Git仓库中的三大区域包括工作区（Working Directory）、暂存区（Staging Area，也称为Index）和版本区（Repository）。  工作区（Working Directory）：  工作区是开发者在本地计算机上存放项目文件的地方，也就是我们日常编辑和修改代码的区域。当开发者对文件进行更改时，这些更改最初只反映在工作区中，而不会立即提交到版本库中。工作区中的文件可以是未被暂存的（即未跟踪的新文件或修改过的文件），也可以是已暂存的（准备提交的更改）。  暂存区（Staging Area/Index）：  暂存区是一个临时存放更改的区域，它位于工作目录和版本库之间。当开发者使用git add命令时，他们可以将工作目录中的更改添加到暂存区。一旦文件被添加到暂存区，它们的状态就会锁定，准备进行提交。使用git commit命令可以将暂存区中的更改提交到版本库中，从而创建一个新的提交（commit）。  版本区（Repository）：  版本区是存储项目历史记录的地方，它记录了项目的所有提交和分支。版本区中包含了项目的所有历史更改，以及每个更改的详细信息，如作者、日期和提交消息。版本区是Git系统中的核心部分，它确保了代码的历史和完整性。在本地计算机上，版本区通常位于工作目录下的.git隐藏文件夹中。对于远程仓库，版本区则托管在远程服务器上，如GitHub或其他代码托管平台。  这三大区域共同协作，使得Git能够有效地管理代码的变更，支持多人协作开发，同时保持项目历史的完整性和可追溯性。  2.进行一次完整的修改、提交、推送操作；  Git status查看仓库状态：    使用ls列出目录：    创建一个文件并查看仓库状态：    修改文件并查看状态：    结果显示已修改，说明git能够正常跟踪版本。  添加修改到暂存区以及撤销修改：      使用diff查看文件修改详情：  commit提交到版本区：    使用git log查看版本区提交记录：    配置个人信息：    查看当前所有分支信息：    commit提交此版本：    添加远程GitHub仓库url：    添加了一个叫做myhub的远程仓库链接。  使用push推送：    推送成功，在GitHub里查看：    所有资料都已推送完成。  3.进行一次版本回退：  使用git reset --soft HEAD^ 撤销最近的一次提交：    查看仓库状态：    上一次修改已经撤回暂存区。  4.处理 commit 时间线分叉：  查看仓库状态和分支状态：    发现二者出现冲突，强制推送本地版本：    推送成功，与GitHub版本一致。  5.进行本地仓库 commit 变化记录。  执行git log查看版本变化：    选择某一版本哈希值，使用git checkout指令直接回到此版本“    成功回退。 |
| **四、实验总结**  本次实验通过对Git版本控制系统和GitHub平台的实际操作，加深了对工作区、暂存区和版本区三大区域的认识，并理解了它们在版本控制过程中的作用。实验过程中，顺利实现了文件的修改、提交和推送，体验了版本控制的完整流程。此外，通过执行版本回退操作，认识到了版本历史追踪的重要性，这对于代码管理和问题调试具有显著意义。同时，通过查看本地仓库的commit记录，学习了如何有效地在本地管理代码变更历史。 |