深 圳 大 学 实 验 报 告

课程名称：软件工程

实验项目名称：实验五：配置管理

学院：计算机与软件学院

专业：软件工程

指导教师：杜文峰

报告人：__学号：___班级：_

实验时间：2025 年 11 月 12 日

教务部制

实验目的与要求：

（1）学习配置管理的基本原理

（2）了解主流的配置管理工具

（3）掌握 Git 版本管理工具的安装，配置

（4）掌握 Git 本地库的创建和配置

（5）掌握本地仓库中的文件管理

（6）了解分支和标签概念

（7）掌握如何创建与合并分支

（8）掌握如何解决分支

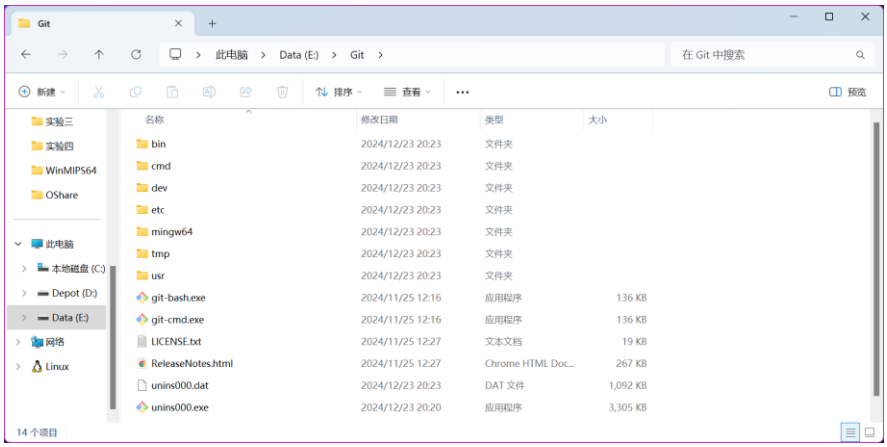
（9）掌握多人协作进行软件开发的方法

（10）掌握标签的创建和相关操作

实验过程及内容：

（1）下载安装 Git

从官网下载安装程序并安装到指定目录，安装程序会设定好环境变量：



在命令提示符查看 git 版本，验证安装成功：

```
C:\Users\OutIn>git -v
git version 2.47.1.windows.1
```

（2）Git 本地仓库的创建和配置

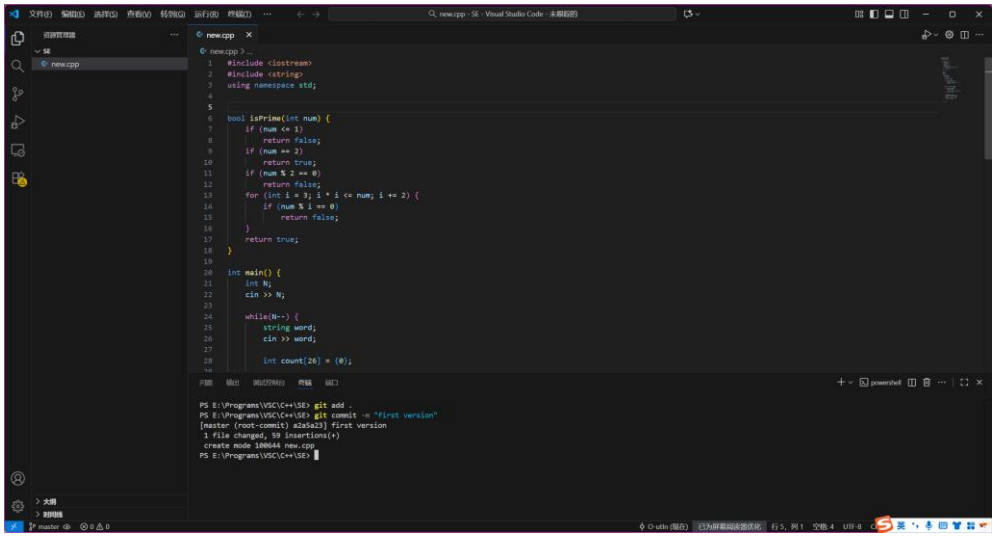
因为 Git 是分布式版本控制系统，所以，每个机器都必须自报家门：名字 Emai 地址。使用 git config --global user.name "Your Name" 和 git config --global user.email email@example.com 可以把当前全部默认设置为输入的名字和邮箱。由于我之前已经设置过了，下边截图展示当前的信息：

进入一个想要作为仓库的目录，然后输入 git init 就可以把这个目录变成 Git 可以管理的仓库：

```
PS E:\Programs\VSC\C++\SE> git init
Initialized empty Git repository in E:/Programs/VSC/C++/SE/.git/
PS E:\Programs\VSC\C++\SE>
```

（3）在本地创建一个 C 语言或者 C++项目，将所有的项目源程序文件提交到本地库中。

创建项目：



（4）多次修改某一个源代码文件，进行多次提交。

进行多次修改，每次修改都提交：

```
 create mode 100644 new.cpp
PS E:\Programs\VSC\C++\SE> git add .
PS E:\Programs\VSC\C++\SE> git commit -m "second version"
[master aa364e0] second version
 1 file changed, 1 insertion(+), 33 deletions(-)
PS E:\Programs\VSC\C++\SE> git add .
PS E:\Programs\VSC\C++\SE> git commit -m "third version"
[master bd6c5f9] third version
 1 file changed, 1 insertion(+), 13 deletions(-)
```

（5）查看文件的历史记录，并且查看不同版本之间的差别。

（6）将文件恢复到前面的某个版本，给出详细的恢复过程。

上一步查到的哈希值就可以代表版本 ID，以此指定版本。我们现在将版本回退到第二个版本“second version”，其以 aa364e 开头：

```
    second version
PS E:\Programs\VSC\C++\SE> git reset --hard aa364e
HEAD is now at aa364e0 second version
PS E:\Programs\VSC\C++\SE>
```

再查看项目，发现已经恢复到第一次修改后的状态，即 second version 提交的内容，意味着恢复正确：

new.cpp

E:\Programs\VSC\C++\SE\new.cpp

```cpp
#include <iostream>
#include <string>
using namespace std;


bool isPrime(int num) {
    if (num <= 1)
        return false;
    if (num == 2)
        return true;
    if (num % 2 == 0)
        return false;
    for (int i = 3; i * i <= num; i += 2) {
        if (num % i == 0)
            return false;
    }
    return true;
}

int main() {
    int N;
    cin >> N;



    return 0;
}
```

（7）删除某一个提交文件，并且查看项目状态。

添加一个新文件 test.txt 到 Git 并且提交：

```
PS E:\Programs\VSC\C++\SE> git add .\test.txt
PS E:\Programs\VSC\C++\SE> git commit -m "new txt"
[master 77f5dc9] new txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test.txt
PS E:\Programs\VSC\C++\SE>
```

删除这个文件并查看项目状态：

```
PS E:\Programs\VSC\C++\SE> rm test.txt
PS E:\Programs\VSC\C++\SE> git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    test.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

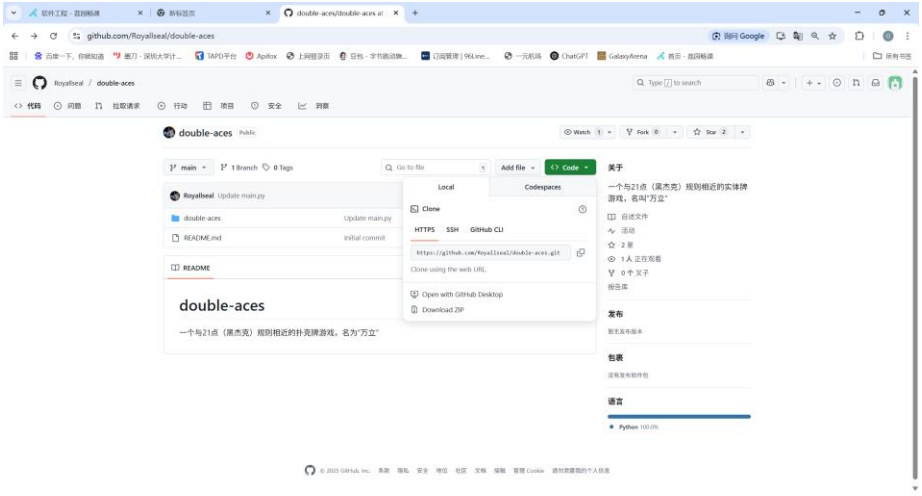
用 git rm 删除并提交：

```
PS E:\Programs\VSC\C++\SE> git rm test.txt
rm 'test.txt'
PS E:\Programs\VSC\C++\SE> git commit -m "delete txt"
[master c8a205b] delete txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 test.txt
```

现在，文件就从版本库中被删除了。

（8）进入 Github，尝试 clone 一个开源项目。

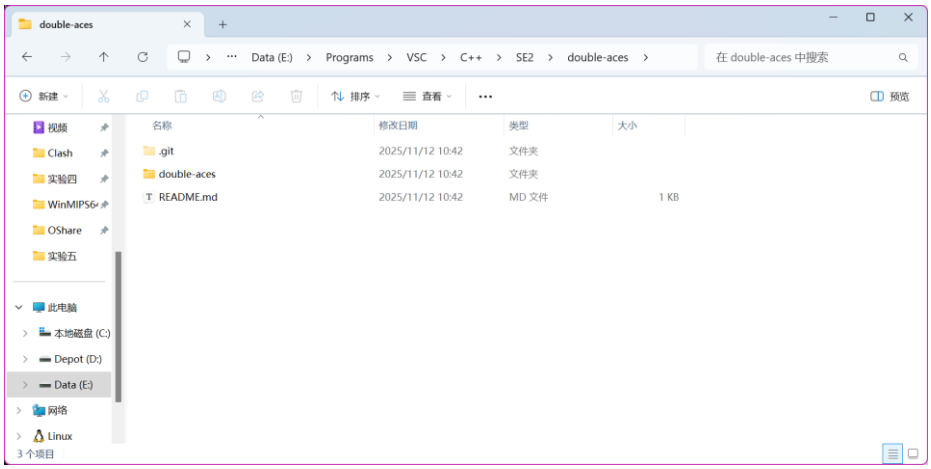
进入 GitHub，复制目标项目地址：



克隆到本地：

```
PS E:\Programs\VSC\C++\SE2> git clone https://github.com/Royallseal/double-aces.git
Cloning into 'double-aces'...
remote: Enumerating objects: 17, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 17 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (17/17), 7.93 KiB | 3.97 MiB/s, done.
Resolving deltas: 100% (2/2), done.
PS E:\Programs\VSC\C++\SE2>
```

如图所示，项目已经到本地：



（9）在本地创建一个 C 语言或者 C++项目，将所有的项目源程序文件提交到本地库中。操作与（3）重复

（10）为某个文件创建分支，在分支中提交和对比文件

创建新的分支并选择：

```
PS E:\Programs\VSC\C++\SE> git checkout -b new_branch
Switched to a new branch 'new_branch'
```

修改文件后提交：

```
PS E:\Programs\VSC\C++\SE> git checkout -b new_branch
Switched to a new branch 'new_branch'
PS E:\Programs\VSC\C++\SE> git add .\new.cpp
PS E:\Programs\VSC\C++\SE> git commit -m "new branch1"
[new_branch 49d98b5] new branch1
 1 file changed, 13 deletions(-)
PS E:\Programs\VSC\C++\SE>
```

查看分支间的差异：

```
PS E:\Programs\VSC\C++\SE> git branch
  master
* new_branch
PS E:\Programs\VSC\C++\SE> git diff master new_branch
diff --git a/new.cpp b/new.cpp
index 0ce8bfa..c97861d 100644
--- a/new.cpp
+++ b/new.cpp
@@ -3,19 +3,6 @@
 using namespace std;


-bool isPrime(int num) {
-    if (num <= 1)
-                return false;
diff --git a/new.cpp b/new.cpp
index 0ce8bfa..c97861d 100644
--- a/new.cpp
+++ b/new.cpp
```

（11）将修改的内容合并到主分支

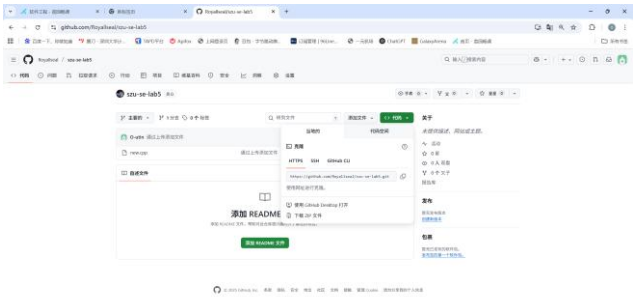
切换到主分支（master），将分支合并到主分支：

```
PS E:\Programs\VSC\C++\SE> git checkout master
Switched to branch 'master'
PS E:\Programs\VSC\C++\SE> git merge new_branch
Updating c8a205b..49d98b5
Fast-forward
 new.cpp | 13 -------------
 1 file changed, 13 deletions(-)
```

没有冲突，合并成功。

（12）与项目组同学合作，各自将本地代码提到同一个远程版本库

新建一个远程代码仓库：

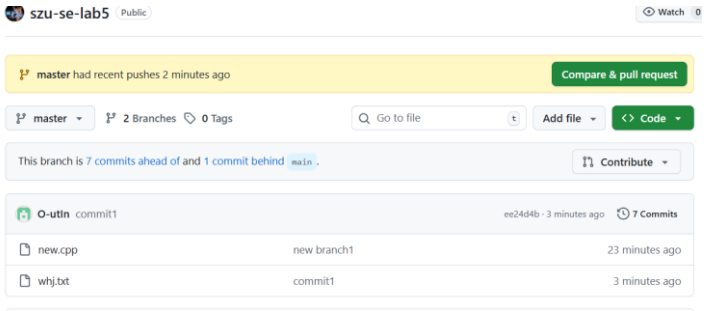


```
PS E:\Programs\VSC\C++\SE> git remote add origin https://github.com/Royallseal/szu-se-lab5.git
PS E:\Programs\VSC\C++\SE>
```

上传代码到远程代码仓库：

```
PS E:\Programs\VSC\C++\SE> git add .
PS E:\Programs\VSC\C++\SE> git commit -m "commit1"
[master ee24d4b] commit1
 1 file changed, 1 insertion(+)
PS E:\Programs\VSC\C++\SE> git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 274 bytes | 274.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Royallseal/szu-se-lab5.git
   feb678e..ee24d4b  master -> master
```

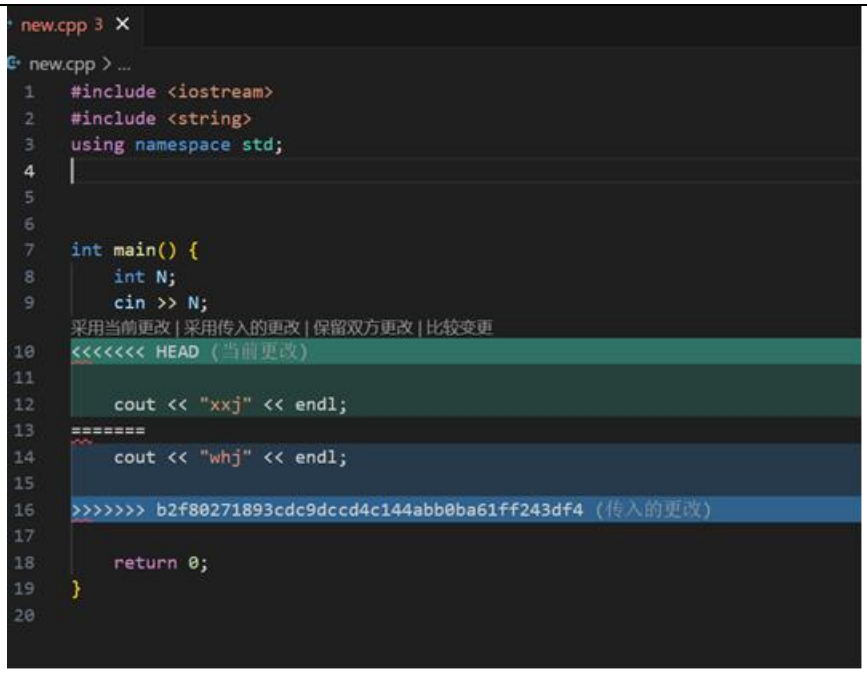
可以看到，上传成功：



（13）与项目组同学共同编写同一个代码文件，体验冲突解决。

当我们同时操作了一个文件并动了同一个函数，就会发送冲突。后提交的同学先将仓库的代码拉到本地，解决冲突后再重新上传：

```
PS E:\Programs\VSC\C++\SE> git add .
PS E:\Programs\VSC\C++\SE> git commit -m "de"
[master 6103087] de
 2 files changed, 6 insertions(+), 1 deletion(-)
 create mode 100644 .vscode/settings.json
PS E:\Programs\VSC\C++\SE> git push origin master
To https://github.com/Royallseal/szu-se-lab5.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/Royallseal/szu-se-lab5.
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
PS E:\Programs\VSC\C++\SE> git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 330 bytes | 16.00 KiB/s, done.
```



（14）为项目增加本地和远程标签，并查看、删除标签

增加一个轻量标签：

```
PS E:\Programs\VSC\C++\SE> git tag v1.0
```

将标签推送到远程仓库：

```
PS E:\Programs\VSC\C++\SE> git push origin master v1.0
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 16 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 882 bytes | 441.00 KiB/s, done.
Total 8 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Royallseal/szu-se-lab5.git
   9954eb4..9e4e930  master -> master
 * [new tag]         v1.0 -> v1.0
```

查看标签：

```
PS E:\Programs\VSC\C++\SE> git tag
v1.0
```

删除标签：

```
PS E:\Programs\VSC\C++\SE> git tag -d v1.0
Deleted tag 'v1.0' (was 9e4e930)
PS E:\Programs\VSC\C++\SE> git tag
```

图文并茂地撰写所有实习过程，并配有步骤的截图。

实验结论： 通过本次实验，我深入学习了 Git 版本控制工具的基本操作与配置，掌握了本地库的创建、文件管理、分支与标签的操作流程。同时，我了解了如何使用分支进行开发、合并分支以及解决冲突，特别是在与项目组同学协作时，体会到 Git 在多人开发中的重要性。此外，通过克隆远程仓库和添加标签的实践，我进一步理解了 Git 在管理项目历史版本和标记关键节点方面的优势。整个实验让我熟练掌握了 Git 的常用命令和工作流程，强化了版本控制的实际操作技能，为今后的团队协作和项目开发打下了坚实的基础。
指导教师批阅意见：
成绩评定： 指导教师签字： 年　　月　　日
备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。