中国海洋大学
OCEAN UNIVERSITY OF CHINA

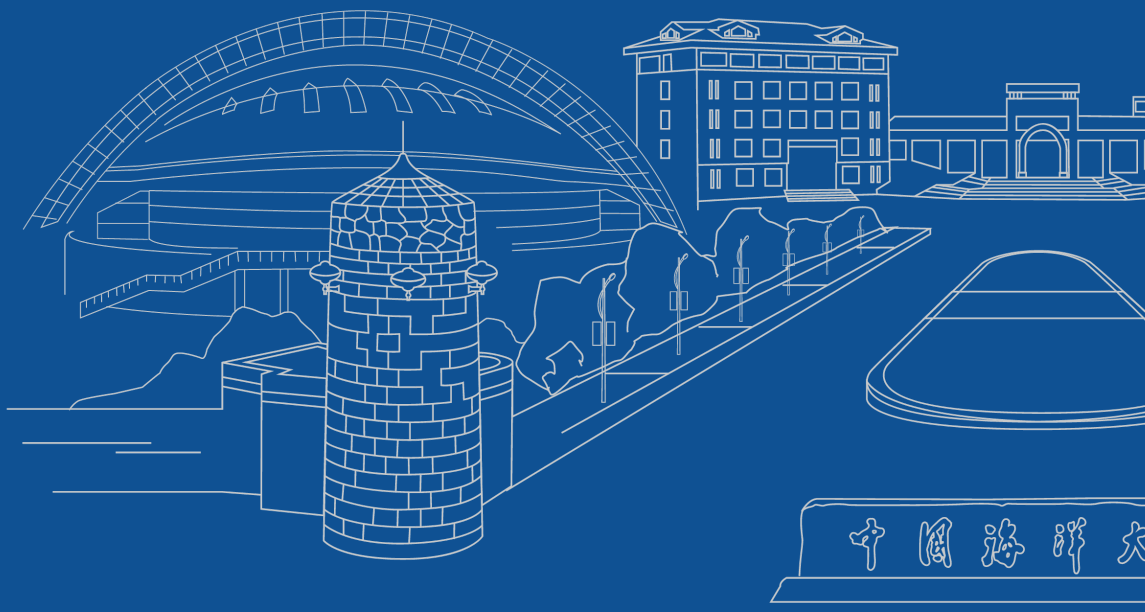
系统开发工具基础第一次实验报告

了解实践 Git 版本控制 学习使用 LaTeX 文档编辑

于景一

23090032047

信息科学与工程学部
计算机学院
计算机科学与技术专业



JStar, August 2024

中国海洋大学
OCEAN UNIVERSITY OF CHINA

系统开发工具基础第一次实验报告

了解实践 Git 版本控制 学习使用 LaTeX 文档编辑

于景一

23090032047

指导老师: 周小伟
系统开发工具基础, 计算机学院

信息科学与工程学部
计算机学院
计算机科学与技术专业

JStar, August 2024

概述

本实验报告使用由**于景一**制作的 LaTeX 模板完成。关于此模板的信息，您可以前往GitHub模板仓库具体了解。[1][2][3]

本实验报告是*系统开发工具基础课程*的第一次实验报告，主要关于**使用 Git 进行版本控制，以及掌握 LaTeX 模板的基本使用方法**，总结个人经验，记录心得体会。

关于 Git 的部分，我们主要学习了 Git 的基本概念，并通过实际操作，掌握了它的基本使用方法，包括但不限于初始化仓库、添加文件、提交更改、查看历史、分支管理等。特别的，由于我们在*程序设计基础实践*课程进行了小组协作（使用 GitHub 进行代码托管和版本管理），我们在实操中练习了使用 Git 进行远端操作、版本控制，以及如何解决冲突等问题。

至于 LaTeX 部分，我在制作个人模板、完成*计算机工程伦理*的课程报告[4]的过程中，对 LaTeX 有了更深入的了解。在实操中，我学习了 LaTeX 的基本语法，包括但不限于文档结构、文本格式、数学公式、插图表格、参考文献等。例如本模板，就是学习的成果，使用交叉编译，采用 *XeLaTeX*→ *BibTeX*→ *XeLaTeX*→ *XeLaTeX* 的编译顺序，支持中文、英文混排，支持引用文献，兼具较好的兼容性和美观性。

NOTE

本模板仍非特别完善，仍在活跃维护中，接下来将会进一步提升本地化水平，欢迎您提出宝贵意见。

[1] 或您可直接搜索 GitHub 账号 *@jstar0* 了解更多

[2] 您请注意，本模板基于 LPPL v1.3c 分发，本项目在原模板Polytechnic University of Leiria: LaTeX Thesis Template的基础上进行了合法地大量二改，包括但不限于自定义风格、中文化支持、样式重定义、功能增加等。

[3] 模板提供两种样式，一种为学术论文样式，另一种为实验报告样式，具体区别请检查 GitHub 仓库上的两个分支。

[4] 如果您对此感兴趣，可以点击此链接进行预览。

目录

1 使用 Git 进行版本控制

NOTE

本人多年前就接触过 Git 的使用方法，相关记录可以查看本人的 GitHub 账号（@jstar0），并用它进行过代码托管、持续集成（自动化构建和部署）、项目管理、提 PR、SaaS 对接等操作。**因此在本报告中，对 Git 相关操作描述可能相对简略。**当然，经过系统学习，我对 Git 的掌握更加熟练和深入了。

1.1 理论基础

1.1.1 Git 的基本模型

Git 使用**对象和引用**的方式来管理版本控制。对象是 Git 中最基本的单位，包括三种类型：**blob**、**tree** 和 **commit**。其中，blob 是文件内容，tree 是目录结构，commit 是提交信息。而引用是指向对象的指针，包括分支、标签等。通过对象和引用的方式，实现了 Git 的基本模型，包含三个基本区域：**工作区、暂存区和版本库**。工作区是指用户进行编辑的区域，暂存区是指用户准备提交的地方，版本库用于存储历史记录。

此部分参考了 (计算机教育中缺失的一课, 2020) 的概念，仅作简述。

1.1.2 整理 Git 的基本操作

我们将远程操作和本地操作分开研究，这是基于我们实用性的考量。在远程操作中，我们主要关注**克隆、推送、拉取**等操作，而在本地操作中，我们主要关注**初始化库、添加、提交、查看、分支**等操作。我们给出两个表格Table 1.1和Table 1.2

远程操作：

表 1.1: *Git 远程操作一览表*

操作名	命令	简介
克隆	`git clone <url>`	从远程仓库克隆到本地
推送	`git push`	将本地提交推送到远程仓库

表 1.1: *Git* 远程操作一览表（续）

操作名	命令	简介
拉取	`git pull`	从远程仓库拉取最新版本
获取	`git fetch`	从远程仓库获取最新版本
指定远程	`git remote`	查看远程仓库
分支	`git branch`	查看分支

本地操作：

表 1.2: *Git* 本地操作一览表

操作名	命令	简介
初始化	`git init`	初始化一个 Git 仓库
添加	`git add <file>`	将文件添加到暂存区
提交	`git commit -m <message>`	将暂存区的文件提交到版本库
查看	`git status`	查看当前状态
分支	`git branch`	查看分支

1.2 实践操作

1.2.1 初始化一个仓库

要初始化一个仓库，我们可以选择直接从本地初始化或者从远程仓库克隆。从本地初始化的操作如下：

```
1       git init
```

代码 1.1: *INITialize*(初始化) 一个 *Git* 仓库

而若要从云端创建一个仓库并克隆到本地，以 GitHub 为例，我们先在云端创建一个仓库，如下图：

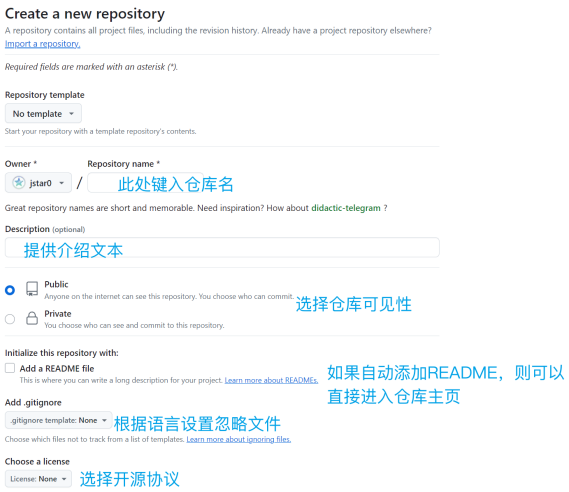


图 1.1: 在 *GitHub* 上创建一个仓库

然后，我们可以使用`git clone`命令将仓库克隆到本地：

```
1        git clone <url>
```

代码 1.2: *CLONE*(克隆) 一个 *Git* 仓库

1.2.2 添加、提交和查看

在仓库中，最基本的操作就是，添加文件、提交文件，以及查看当前状态。添加文件会将文件添加到**暂存区**，而提交文件会将暂存区的文件提交到**版本库**。添加文件的操作如下：

```
1        git add <file>
```

代码 1.3: *ADD*(添加) 文件到暂存区

如需提交所有文件，可以使用 `*`（通配符），或者在不使用此指令而是使用 `git commit -a` 命令。但注意，使用此指令或许会造成不希望的结果。提交文件的操作如下：

```
1        git commit -m <message>
```

代码 1.4: *COMMIT*(提交) 文件到版本库

注意，上述指令中`-m`参数可以留空，不过后期会自动打开一个文件要求输入 Commit 信息。查看当前状态的操作如下：

```
1        git status
```

代码 1.5: *STATUS*(状态) 查看当前状态

上述操作是 Git 中最基本的操作，我们在本地创建一个仓库作为示例，如下图：

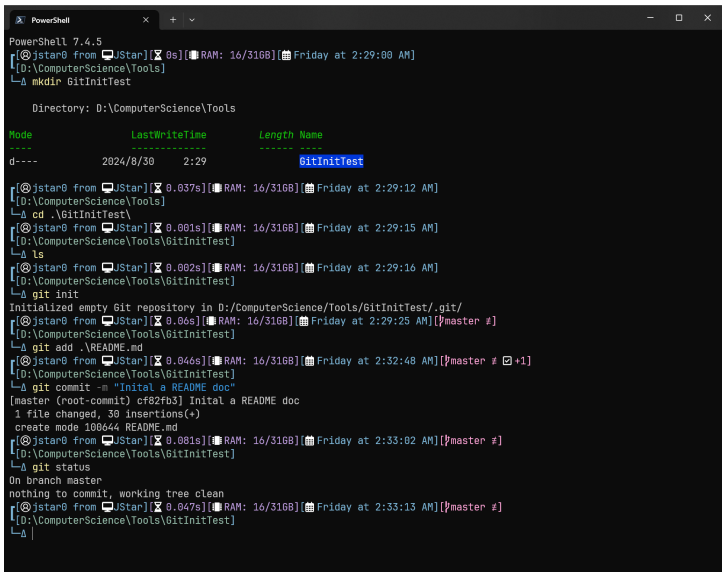


图 1.2: 本地操作创建、提交示例

1.2.3 推送到远端、拉取最新版本

在本地操作完成后，我们可以将本地的提交推送到远程仓库，或者从远程仓库拉取最新版本。推送的操作如下：

2

Document Variables

In this document, it is possible to change the value of various variables, such as the author's name, the title of the work, and others, so that the author has greater flexibility in their work. The file that operates these variables is `Variables/Variables.tex`. Below, in Table 2.1, you can see all the variables that can be changed, as well as whether or not they must be filled in for the theme to work correctly. Please note that in order to omit a variable, it is only necessary to comment it out. Additionally, if you wish to suggest any other variable, please contact me via the contacts available on GitHub or in the class file of the template.

表 **2.1:** *Document variables that can be changed.*

Variable	**Mandatory**	**Optional**
Title	✓	-
Subtitle	✓	-
University	✓	-
School	✓	-
Department	✓	-
Degree	✓	-
Course	-	✓
Local and date	✓	-
First author name	✓	-
First author identification	✓	-
Second author name	-	✓
Second author identification	-	✓
Third author name	-	✓
Third author identification	-	✓
Supervisor name	✓	-
Supervisor e-mail	✓	-
Supervisor title and affiliation	✓	-
Co-supervisor name	-	✓
Co-supervisor e-mail	-	✓
Co-supervisor title and affiliation	-	✓
Second co-supervisor name	-	✓
Second co-supervisor e-mail	-	✓
Second co-supervisor title and affiliation	-	✓

3 Custom Notes

In the course of writing, it is sometimes necessary to insert notes in the middle of the document so that in the future we can return to them and read what we have written. Sometimes we need a space to jot down a quick `TODO`. With this in mind, two commands have been developed: `\todo{TEXT}` and `\note{TEXT}`. These commands accept normal text and produce two different types of boxes, respectively, for each type of use. These boxes are "breakable", which means that they can be placed anywhere in the document and will behave as if they were text. Below is a simple usage of both environments. Please note that both of these environments were created with the intention of helping you develop your work and **should not be used in the final version of your document!**

TODO

Write about X! Later, we should remove Y.

NOTE

Come to me, all you who are weary and burdened, and I will give you rest. (一个人类). Take my yoke upon you and learn from me, for I am gentle and humble in heart, and you will find rest for your souls.

参考文献

计算机教育中缺失的一课 (2020). *MIT Missing Semester - Lecture 6: Version Control (git)*. Tech. rep. MIT.

4

Annex A

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.