《软件过程与工具》实验2要求

实验2：配置管理工具Git实践

一、实验目的：

1. 了解配置管理工具Git及相应用环境；

2. 熟练掌握Git的基本指令和分支管理指令；

3. 掌握Git支持软件配置管理的核心机理；

4. 在实践项目中使用Gitee/GitLab/GitHub管理自己的项目源代码。

二、实验要求：

1. 本次实验由个人单独完成；

2. 按照第三条步骤进行实验；

3. 遵循实验报告模板撰写报告；

. 上交文件：

1. 实验报告文件命名：《软件过程与工具》实验2报告-学号-姓名.docx；
2. 上交时间：第9周周六（2020.11.07）24：00前；
3. 上交方式：发送到老师信箱：fgx@hit.edu.cn。

三、实验步骤指导：

**1.在本地机器上安装Git工具**

（1）Linux：

$ yum install git-core #在redhat等系统下用yum

$ apt-get install git-core #在debian, ubuntu等系统下用apt-get

（2）Windows：https://git-for-windows.github.io 下载安装包

TortoiseGit及语言包下载地址：https://tortoisegit.org/download。

（3）远程Git服务器：Gitee(<http://gitee.com>)  
 注：本课程实验考虑网络因素，搭建了实验室Git服务器

（4）说明：

* Linux和Windows下的Git安装操作二选一即可；
* 通过浏览器进入Gitee/GitLab/Github上申请账号；
* 使用命令行方式完成实验，避免图形界面下的操作。

**2.实验场景1：仓库创建与提交**

（1）R0：针对R1和R7，在进行每次Git操作之前，随时查看工作区、暂存区、Git仓库的状态，确认项目里的各文件当前处于什么状态；

（2）R1：本地初始化一个Git仓库，将自己在Lab1中所创建项目的全部源文件加入进去，纳入Git管理；

（3）R2：提交；

（4）手工对Lab1的某个文件进行修改；

（5）R3：查看上次提交之后都有哪些文件修改、具体修改内容是什么（查看修改后的文件和暂存区域中相应文件的差别）；

（6）R4：重新提交；

（7）再次对Lab1的某个文件进行修改；

（8）R5：重新提交；

（9）R6：把最后一次提交撤销；

（10）R7：查询提交记录。

**3.实验场景2：分支管理**

（1）在你的Gitee/GitLab/GitHub上，通过Web界面建立一个Project，将不少于10个文件（程序代码、文档等）加入进去，形成初始分支B1；在B1基础上建立两个并行的分支B2、B3，手工对B2和B3上的某些文件进行不同程度的修改并提交；

（2）R8：从Gitee/GitLab/GitHub上(URL)克隆一个已有的Git仓库到本地；

（3）R9：获得该仓库的全部分支；

（4）R10：在B2分支基础上创建一个新分支C4；

（5）R11：在C4上，对4个文件进行修改并提交；

（6）R12：在B3分支上对同样的4个文件做不同修改并提交；

（7）R13：将C4和B3分支合并，若有冲突，手工消解；

（8）R14：查看目前哪些分支已经合并、哪些分支尚未合并；

（9）R15：将C4和B3合并后的分支删除，将尚未合并的分支合并到一个新分支上，分支名字为你的学号。

**4.实验场景3：远程分支管理**

（1）R16：将本地以你的学号命名的分支推送到Gitee/GitLab/GitHub上；

（2）R17：将R1到R7各步骤得到的结果推送到Gitee/GitLab/GitHub上；

（3）R18：在Gitee/GitLab/GitHub上以Web页面的方式查看你的两个仓库的当前状态。