B0911007Y-01/02 2021-2022学年春季学期

计算机组成原理（研讨课）

Git基础知识

2022年3月4日

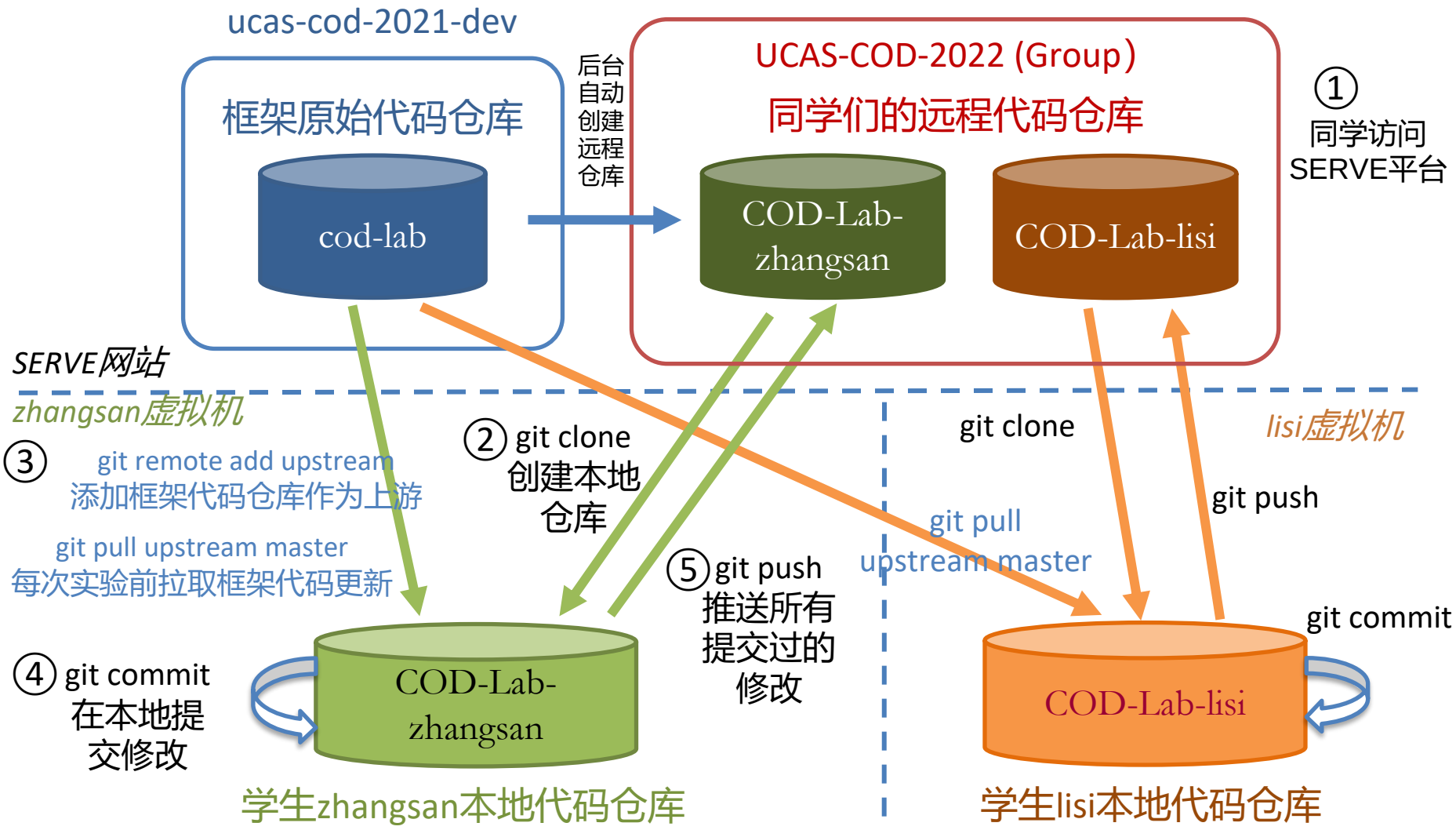
中国科学院大学
University of Chinese Academy of Sciences

中科院计算所
INSTITUTE OF COMPUTING TECHNOLOGY,CAS

为什么要使用Git?

- 学术界、工业界及开源社区主流的项目管理、代码版本维护方法
- 工作代码的云端备份
- 代码相关问题的Issue提交与互动解答，固化知识库
- 实验框架代码的管理、发布、更新与进度检查

实验代码仓库结构化视图



Git 的使用：基本概念

❑ **仓库 / Repository**
- .git 目录 + 工作目录

```
git_example  <- repository
├── .git     <- git database
├── build
├── src      <- working directory
└── target   (can be any file or directory)
```

❑ **提交 / Commit**
- 一次修改记录
- 主要包含：
  - Hash：基于修改内容和父节点，通过 SHA-1 算法生成的40位标识串
  - 父 commit：修改所对应的基础版本，1~2个
  - 时间：提交时间（执行 commit 时间，非文件系统的修改时间）
  - 作者信息
  - 修改内容：增删修改的文件列表，及对应文件的 diff 信息
  - 标题和说明

```
commit 3441368aae7d7e13d78166793c551dd14d0f89da (HEAD -> master)
Author: panxiaoquan <3298900670@qq.com>
Date:   Fri Mar 4 09:12:26 2022 +0800

    change test

diff --git a/test b/test
index e69de29..9daeafb 100644
--- a/test
+++ b/test
@@ -0,0 +1 @@
+test
```

❑ **Commit tree**
- 通过父子关系联系起来的修改历史记录

Git 的使用：基本概念

- **分支 / Branch**
  - 对应一个 commit tree，指向其最新 commit
  - 一般用于表示一系列新 commit 的目的/主题/场景
- **HEAD**
  - 当前所处分支最新 commit 的别名（指针）
- **工作区 / Working tree**
  - 平时存放项目代码的地方
  - 本地文件系统的内容
- **暂存区 / Staging area**
  - 将对工作区文件的增删修改暂存、收集
  - 从而实现以复数文件的修改作为一次 commit

Git 的使用：基本开发流程

- **`git add <file1> <file2> …` 添加修改和新建文件到暂存区**
- **`git commit` 将暂存区中的修改信息提交**
  - 不加参数 `-m` 会自动启动编辑器撰写提交信息
    - 指定编辑器（以Vim为例）：`git config --global core.editor vim`
  - 一站式快捷命令 ：`git commit –a –m ‘title’`
- **`git push <repo> <branch>` 推送最新修改到远程仓库**
  - 添加仓库 `git remote add <name> <url.git>`
  - 分支名要与本地仓库当前分支保持一致

Git 的使用：常用查询命令

- **git log**
  - 查看当前分支的提交记录（按q退出）
  - 选项 --graph显示commit tree
  - 选项 -p 显示每个文件的diff

```
* 3441368aae7d7e13d78166793c551dd14d0f89da (HEAD -> master, tag: v1.1, tag: v1.0) change test
* d8286747185f181ab557262b820a9a1c694dba39 (tag: v1.3) add test
```

- **git status**
  - 显示当前分支名、暂存区中的文件、已修改的文件、未跟踪的文件等当前环境信息

- **git diff**
  - 查看未添加到暂存区中的修改内容
  - 选项 --cached 则查看暂存区中文件的修改内容
  - git difftool 默认调用 vimdiff 进行并排对比

Git 的使用：分支

- **实验项目中分支的作用**
  - 为较大规模的尝试性修改建立新分支，对修改结果满意后再合并到主分支
  - 在比 commit 更大的粒度上管理代码的修改记录，方便撤销一系列针对同一功能的 commit 记录
- **查看分支：git branch**
- **创建分支：git branch <branch-name>**
  - 基于当前所处分支
- **跳转分支：git checkout <branch-name>**
  - git checkout -b <branch-name> 创建并跳转
- **（强制）删除分支：git branch -D <branch-name>**

Git 的使用：分支

- **合并分支（1）：`git merge <branch-name>`**
  - 将指定名字的分支合并到当前分支上
  - 即将两个分支各自独有的 commit 作用在同一套项目文件上
- **处理冲突**
  - git merge 可能因共同修改的代码片段而中断，此时用 git status 查看冲突文件

```
13
14 <<<<<<< HEAD
15     always @(posedge clk) begin
16         counter <= 32'b1;
17     end
18 =======
19     assign counter = 32'b0;
20 >>>>>>> branch-to-be-merged
21
```

  - 冲突片段具有如图所示的格式，
    上半部分是当前分支的修改，
    下半部分是要合并分支的修改，
    整理完冲突代码后要删除标记用的特殊符号
  - git add 添加处理完冲突的文件，最后通过 git commit 提交
  - git merge --abort 可以放弃此次合并

Git 的使用：标签

- **实验项目中标签的作用**
  - 给仓库历史中的某一个 commit 打上标签，以示重要
  - 人们会使用这个功能来标记发布结点（v1.0、v2.0等等）
- **列出标签：git tag**

```
pxq@PXQ:~/gittest/.git$ git tag
v1.0
v1.1
```

- **创建标签：git tag -a <tag-name> [commit hash] -m <message>**
  - 基于当前最新commit或某一确定commit创建标签
- **查看标签详细信息：git show <tag-name>**
  - git show v1.0 显示v1.0及其所基于的commit的信息

```
tag v1.0
Tagger: panxiaoquan <3298900670@qq.com>
Date:   Fri Mar 4 09:29:23 2022 +0800

v1.0

commit 3441368aae7d7e13d78166793c551dd14d0f89da (HEAD -> master, tag: v1.1, tag: v1.0)
Author: panxiaoquan <3298900670@qq.com>
Date:   Fri Mar 4 09:12:26 2022 +0800

    change test

diff --git a/test b/test
index e69de29..9daeafb 100644
--- a/test
+++ b/test
```

Git 的使用：撤销操作

- 如果当前代码的修改不尽如人意，git 所提供的撤销和回退操作可以快速丢弃修改，返回到指定的代码版本。这也是版本管理的重要意义之一
- git log 可查看 commit 的 hash 用于指定要处理的 commit

```
commit ef27704576d50a2fba44c0e2ae11ff28575f1b0f
Merge: 23a06be d8cfc09
Author: Ran Zhao <zhaoran@ict.ac.cn>
Date:   Fri Mar 9 14:53:32 2018 +0800

    Merge branch 'master' of https://github.com/ucas-cod/prj0-student
```

- 撤销未进入暂存区的修改
  - `git checkout -- <file1> <dir1> …`

Git 的使用：撤销操作

❑ **将文件移出暂存区**

- `git reset <file1> …`
- 修改保留

❑ **`git reset <commit-hash>` 回退到指定的 commit 版本**

- 回退路径上所有的修改移入暂存区
- 选项 --hard 直接抛弃这些修改

```
commit a3e86b61c206c1158506b0d122324b8725615a72
Merge: fab05c0 b99234d
Author: a <a.com>
Date:   Thu Mar 15 16:58:11 2018 +0800

    Merge branch 'branch-to-be-merged'
```

❑ **撤销合并**

- 方法一：git revert -m <n> <merge-commit>
  - n=1为第一父分支，n=2为第二父分支
  - 合并后的 commit 保留，并需要解决可能存在的冲突
- 方法二：git reset --hard <merge-parent-commit>
  - git log 中的 merge commit 会显示其两个分支的 commit
  - 从最新 commit 到父分支之前的 commit 全部舍弃

演示

- 拉取远程代码库：git clone https://gitlab.agileserve.org.cn:8001/ucas-cod-2022/<GitLab用户名>/COD-Lab
- 添加上游框架代码库：cd cod-lab && git remote add upstream https://gitlab.agileserve.org.cn:8001/ucas-cod-2021-dev/cod-lab
- 拉取最新框架代码：git pull upstream master
- 修改代码：echo "Hello COD" > hello.md
- 将修改添加至暂存区：git add hello.md
- 提交修改：git commit -m 'Add hello.md'
- 将修改推送到远程仓库：git push

Q & A ?

中国科学院大学
University of Chinese Academy of Sciences

中科院计算所
INSTITUTE OF COMPUTING TECHNOLOGY,CAS