

用Git高效协作

第一讲 – git版本控制与分支管理



JerryHan3



西科大学 计算机科学学院
SICO UNIVERSITY

本节目录

01. Git 基本概念
了解Git版本控制及其基础概念

Git 基本概念

了解Git版本控制及其基础概念

02. Git Commit
用Commit提交一次版本

Git Commit

用Commit提交一次版本

03. Git Branch
用分支让多项工作齐头并进

Git 分支

用分支让多项工作齐头并进

04. Git Merge
把多条分支合并在一起

Git Merge

把多条分支合并在一起

05. Git Rebase
多个分支的线性叠加

Git Rebase

多个分支的线性叠加

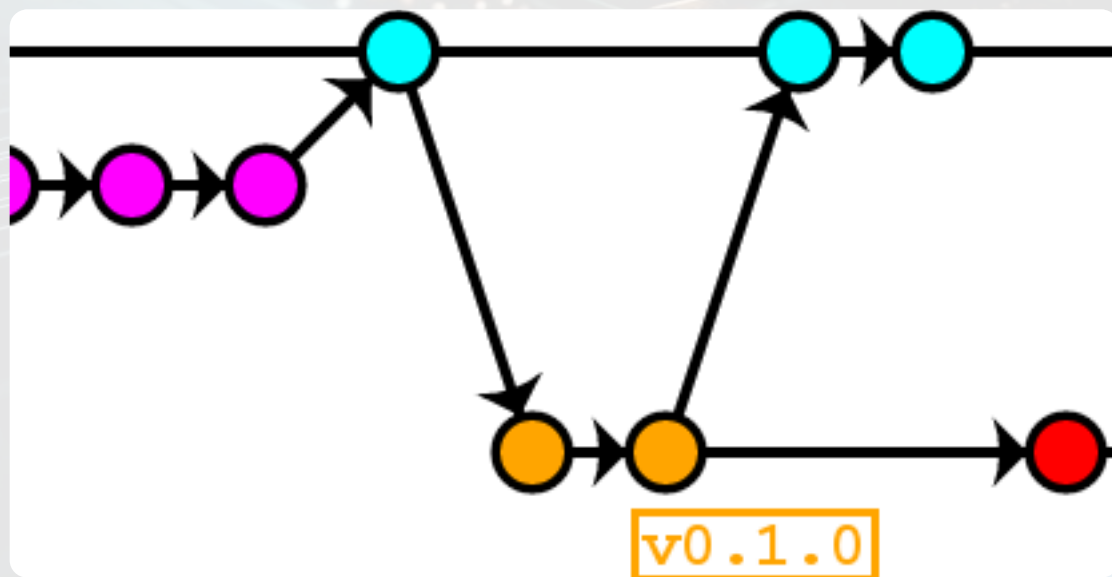
01.Git基本概念

了解Git版本控制及其基础概念



Git是什么?

Git是一个开源的分布式版本管理系统。
简单来讲，它能把你一个项目的各个版本管理得井井有条。



Git记录的是差异

每次提交Git不会把所有文件都复制一遍，它只会记录当前状态与上一版本之间的差异



Git管理的是仓库

经过初始化操作、由git管理版本的文件夹，被称为一个仓库（Repository，简称repo）

02.Git Commit

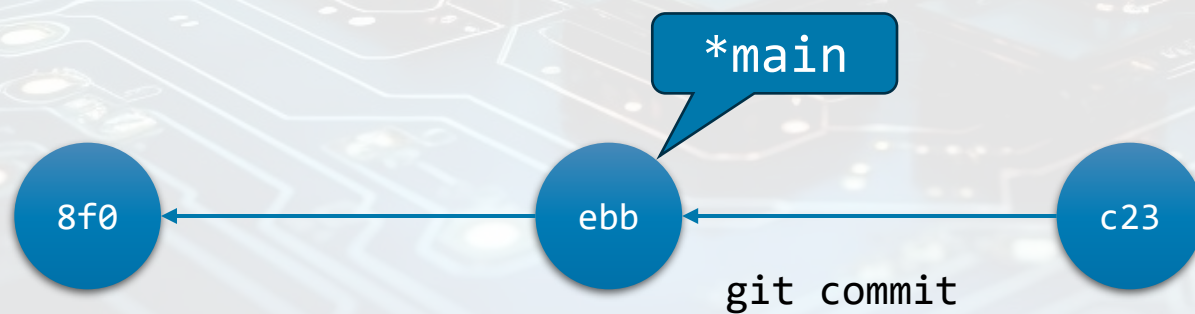
用Commit提交一次版本



Git Commit

Commit – 提交记录

每进行一次commit，就会让git保存一次提交记录。
前面提到git保存的是差异，实际上就是各个commit之间的差异。



提交记录的特性



理解提交记录，有三个关键词



快照

每个提交记录保存的都是整个目录下文件的快照。



差异

全量复制太过臃肿，Git会尝试对比文件并留存差异信息。



历史

提交记录之间父子的关系，记录了整个项目的历史记录

一般的提交步骤

当你每次更改一个Git控制的项目时.....

命令速记

git add . | 暂存所有更改

git commit | 提交一次记录



01

编辑

就和平常一样，
用别的什么工具修改你的项目。



02

暂存 (Stage)

告诉git，这些更改我要了！



03

提交 (Commit)

确认你的更改，
让git生成一个版本记录。

练习时间!

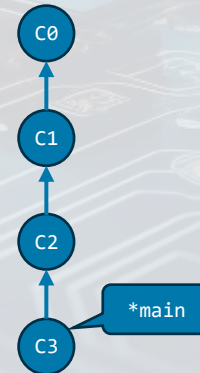
进行2次提交

打开learngitbranching.js.org, 完成关卡1-1。

目标

对项目提交2次记录。

最终效果图



03.Git Branch

用分支让多项工作齐头并进



Git Branch

Branch – 分支

如果两个人同时开发两个功能，直接提交到同一个存储库会发生冲突，怎么办？
如果我不想在开发一个功能的时候让别人的内容影响到我，怎么办？
将工作分到多个分支即可解决！你可以随时从随处创建分支，各个分支之间互不影响。



Git Branch

Branch – 分支

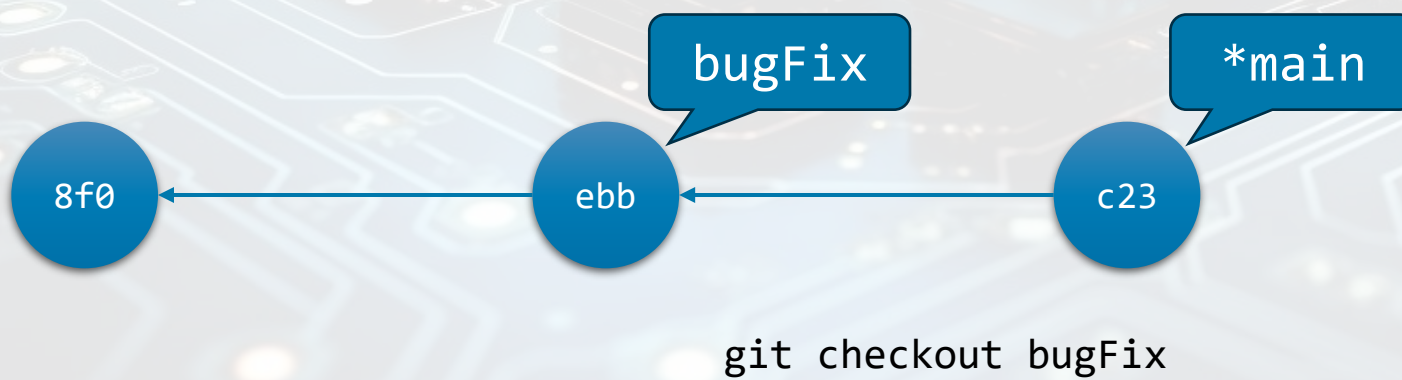
如果两个人同时开发两个功能，直接提交到同一个存储库会发生冲突，怎么办？
如果我不想在开发一个功能的时候让别人的内容影响到我，怎么办？
将工作分到多个分支即可解决！你可以随时从随处创建分支，各个分支之间互不影响。



Git Branch

切换分支

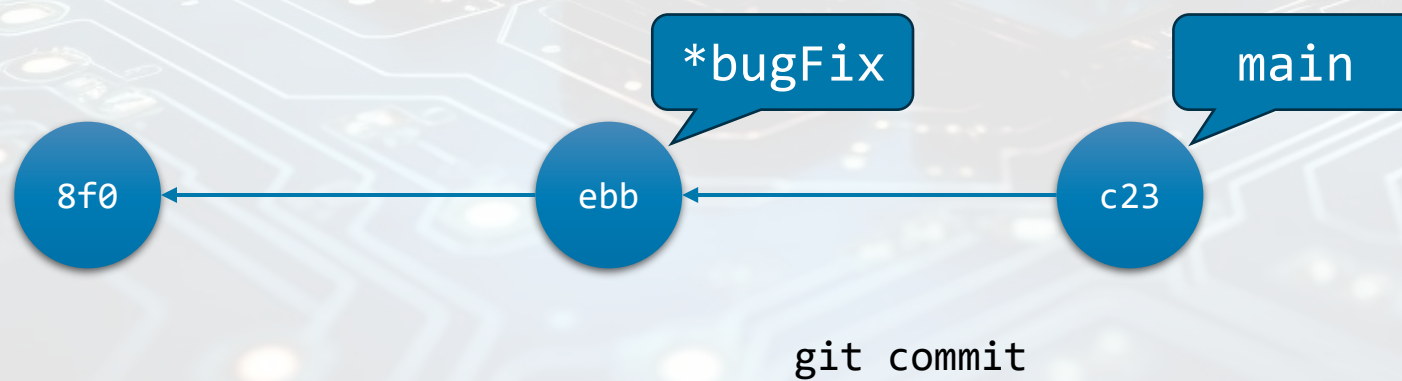
那么该如何切换分支呢?



Git Branch

切换分支

切换分支后，就可以在新的分支上提交更改了。



Git Branch

切换分支

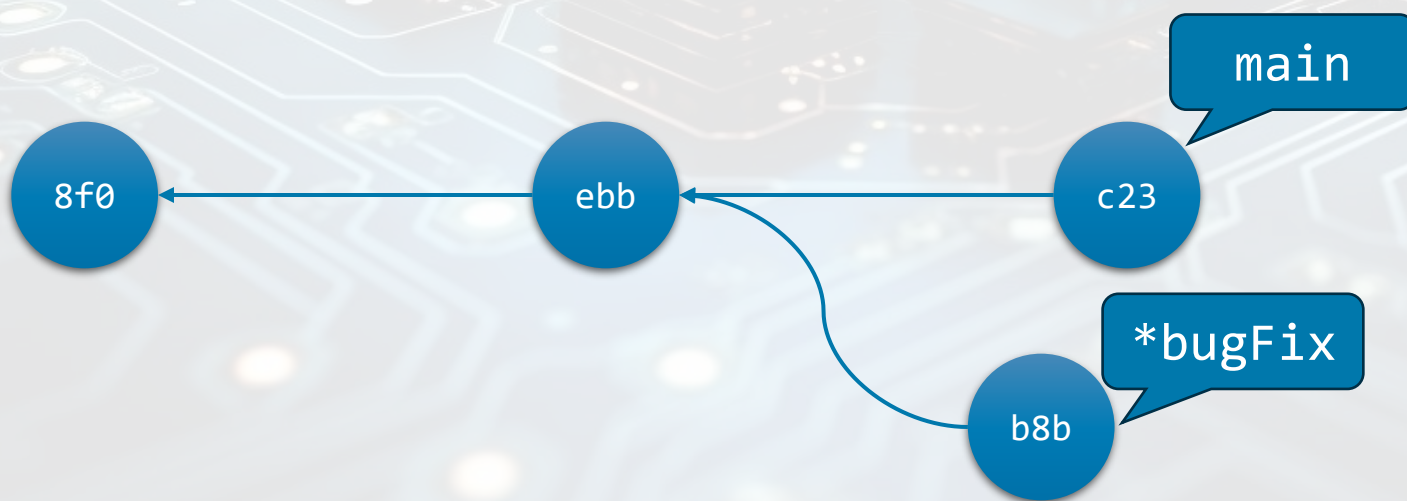
切换分支后，就可以在新的分支上提交更改了。

命令速记

`git branch <名字>` | 在当前位置创建新分支

`git checkout <名字>` | 切换分支

`git checkout -b <名字>` | 新建并切换到分支



练习时间!

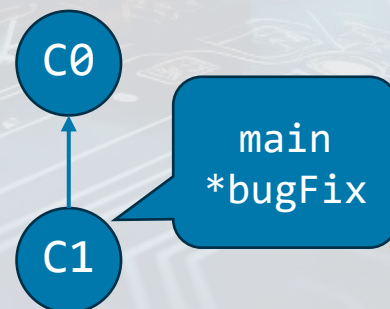
创建并切换到bugFix分支

打开learngitbranching.js.org, 完成关卡1-2。

目标

1. 创建名为bugFix的分支
2. 切换到刚创建的分支

最终效果图



04.Git Merge

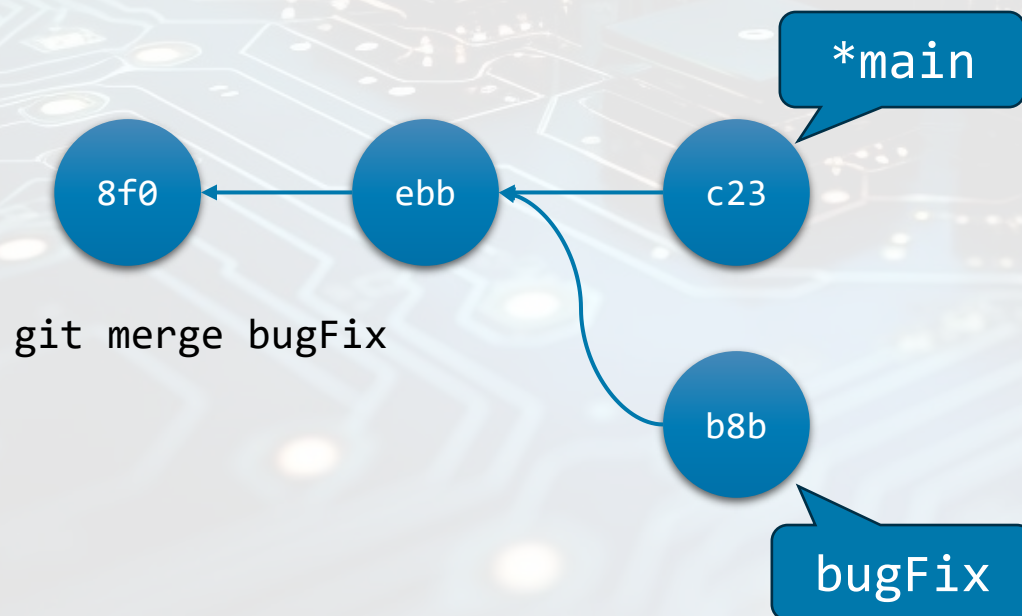
把多条分支合并在一起



Git Merge

如何汇总各个分支上的内容？

现在各个功能都开发得差不多了，是时候合并在一起了。
如果我们能创建一个提交记录，让它**继承两个分支的修改**的话.....



回想.....

提交记录的特性



理解提交记录，有三个关键词

快照

每个提交记录保存的都是整个目录下文件的快照。

差异

会记录两次提交间，Git仓库中文件的变化，并保存修改信息。

历史

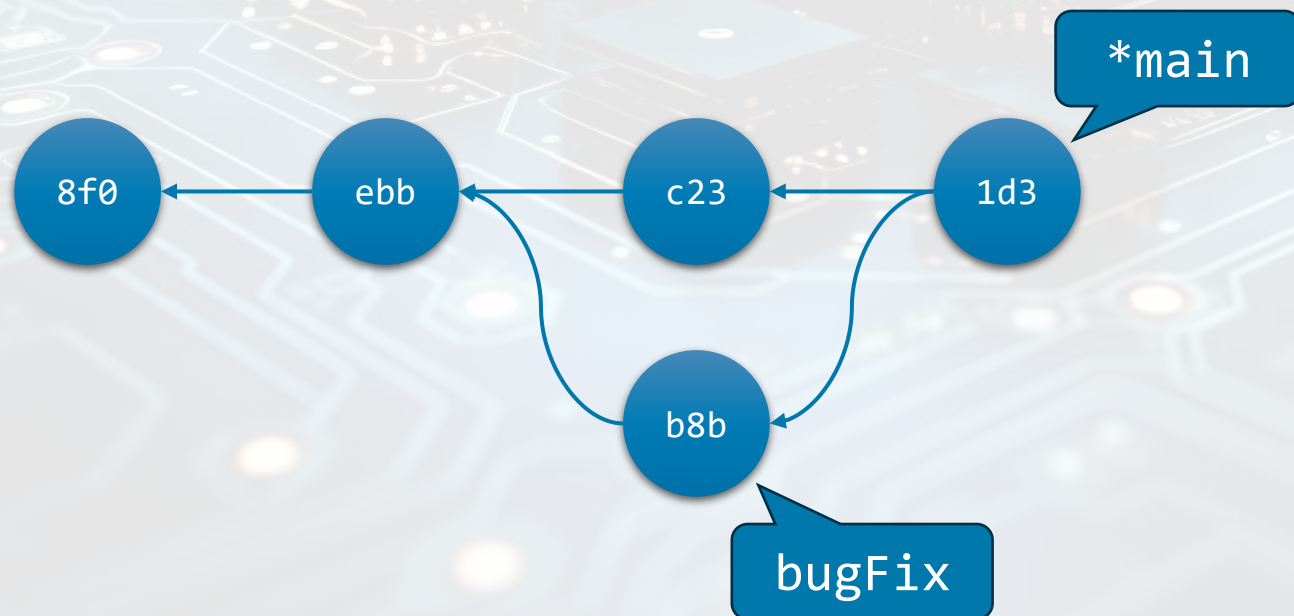
提交记录之间建立了关系，记录了每个提交的提交记录。

EDUCATION
IS
POWER!

Git Merge

如何汇总各个分支上的内容？

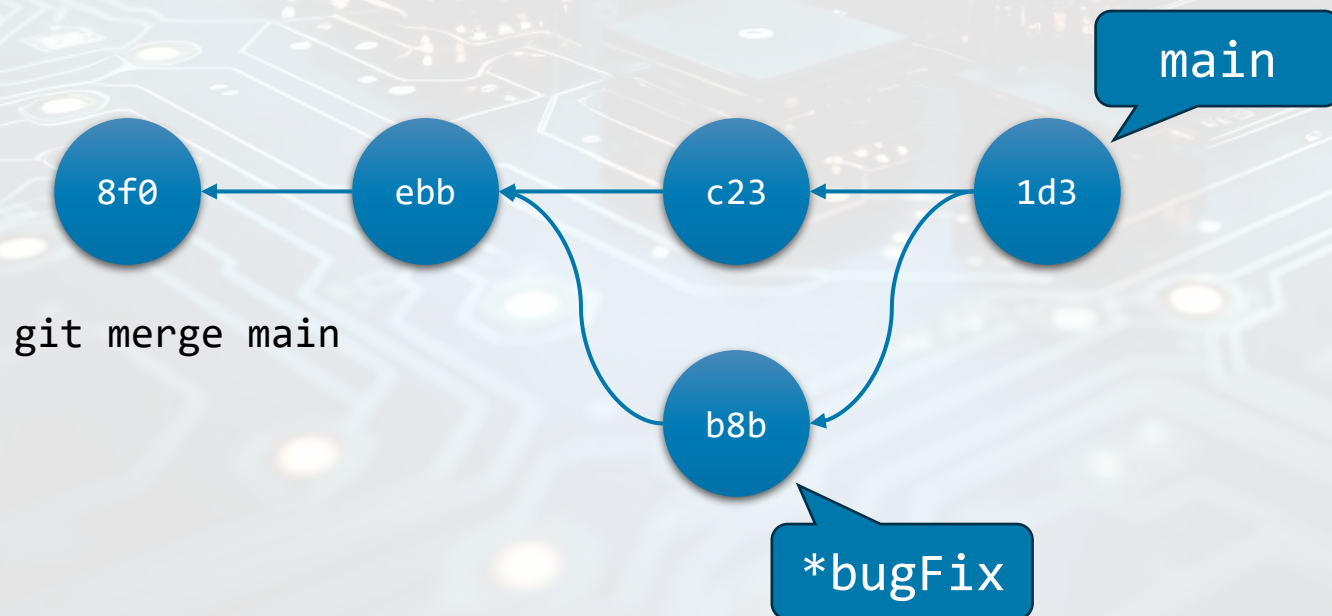
.....就能包括两个分支里面所有的变动了！



Git Merge

“快速前进 (Fast forward)”

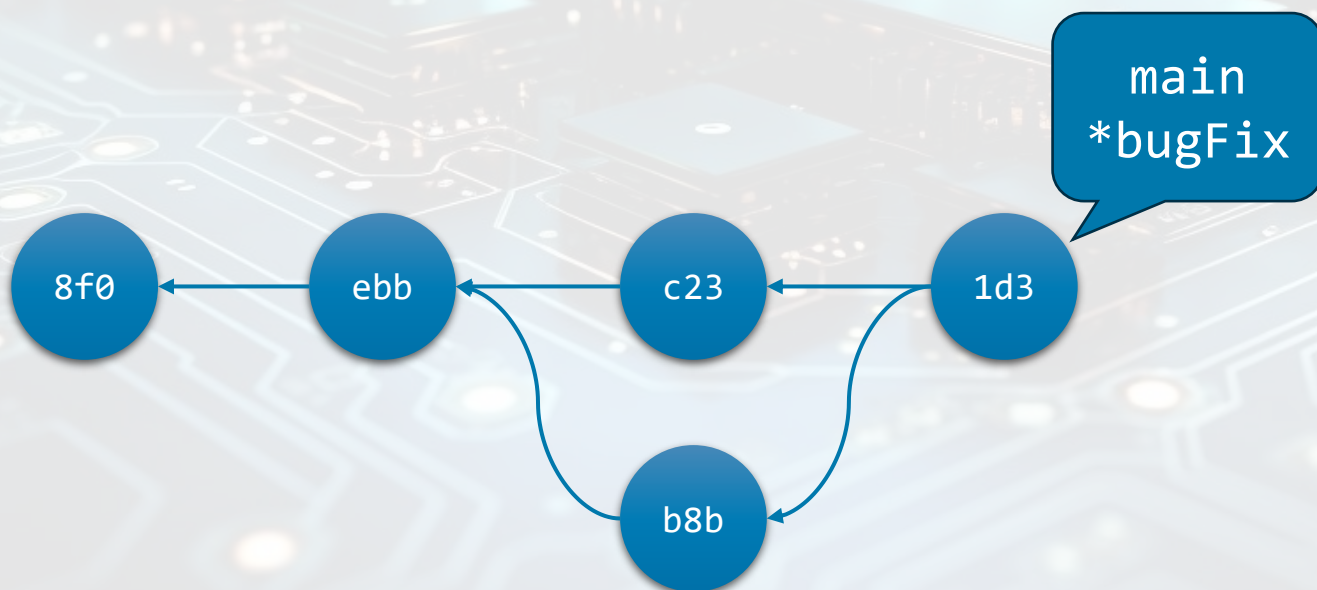
如果merge时git发现目标分支继承自当前分支，git就只会将分支位置向前移动一下。
这叫做“快速前进 (Fast forward)”。



Git Merge

“快速前进 (Fast forward)”

如果merge时git发现目标分支继承自当前分支，git就只会将分支位置向前移动一下。
这叫做“快速前进 (Fast forward)”。



命令速记

`git merge <名字>`

将指定的分支合并到当前分支上。



练习时间!

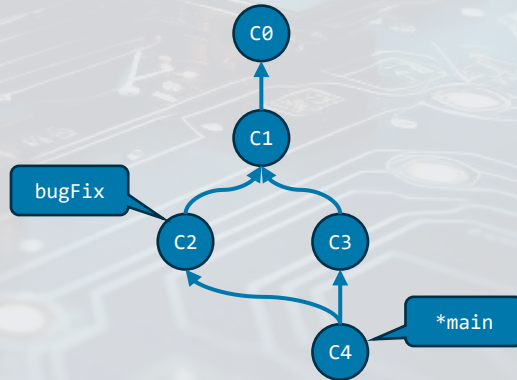
在两个分支上提交并合并

打开learngitbranching.js.org，完成关卡1-3。

目标

1. 创建名为bugFix的分支
2. 依次在bugFix分支和main分支上各提交一次
3. 将bugFix合并到main分支中

最终效果图



05.Git Rebase

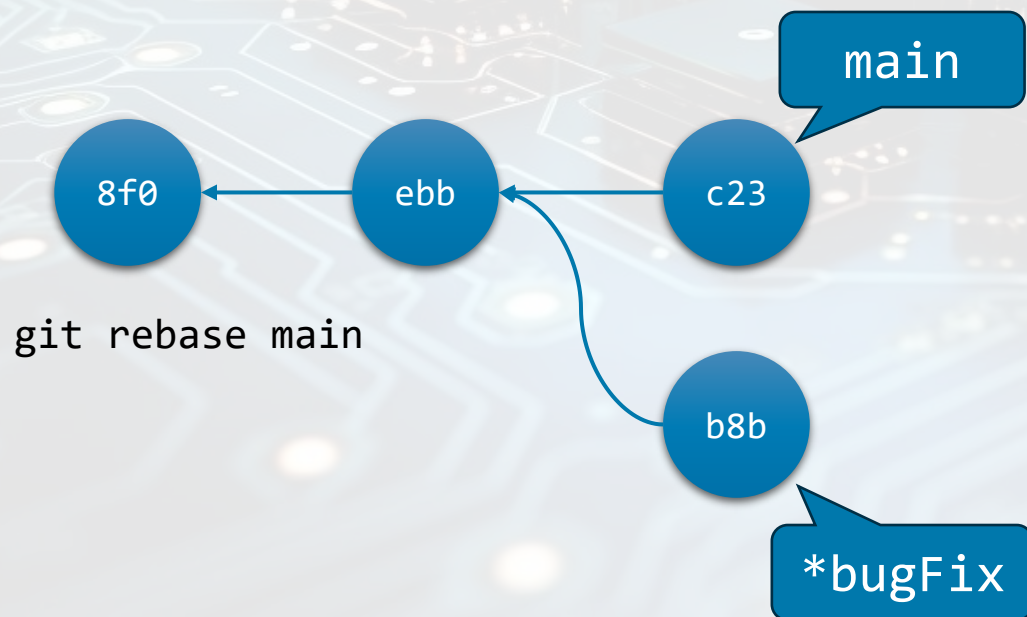
多个分支的线性叠加



Git Rebase

另一种汇总思路

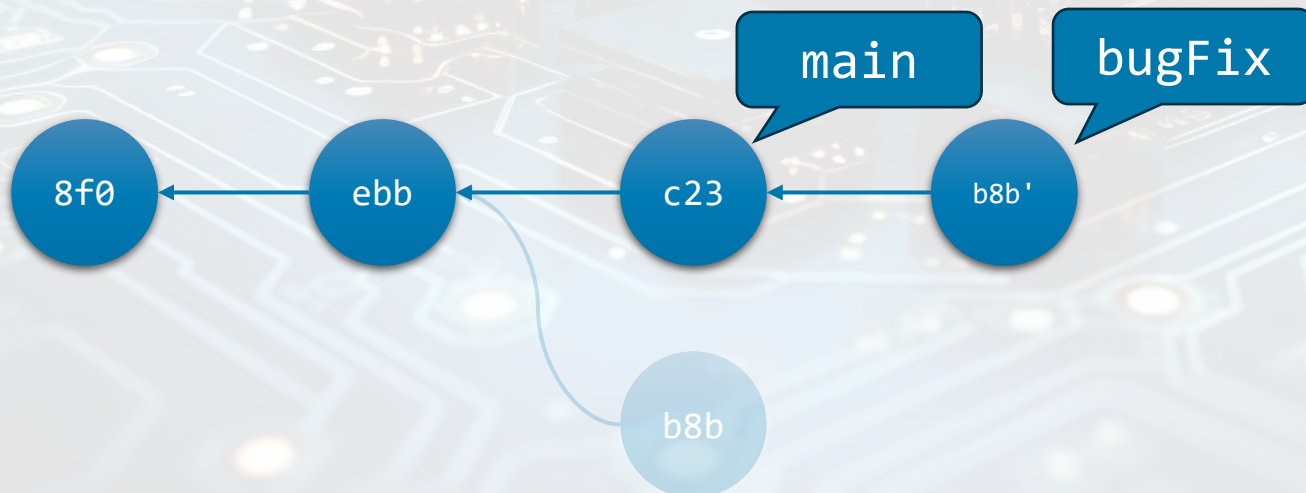
Rebase的目的和Merge是一样的，不同的是它是通过移动当前分支所做的更改实现这一目的。



Git Rebase

另一种汇总思路

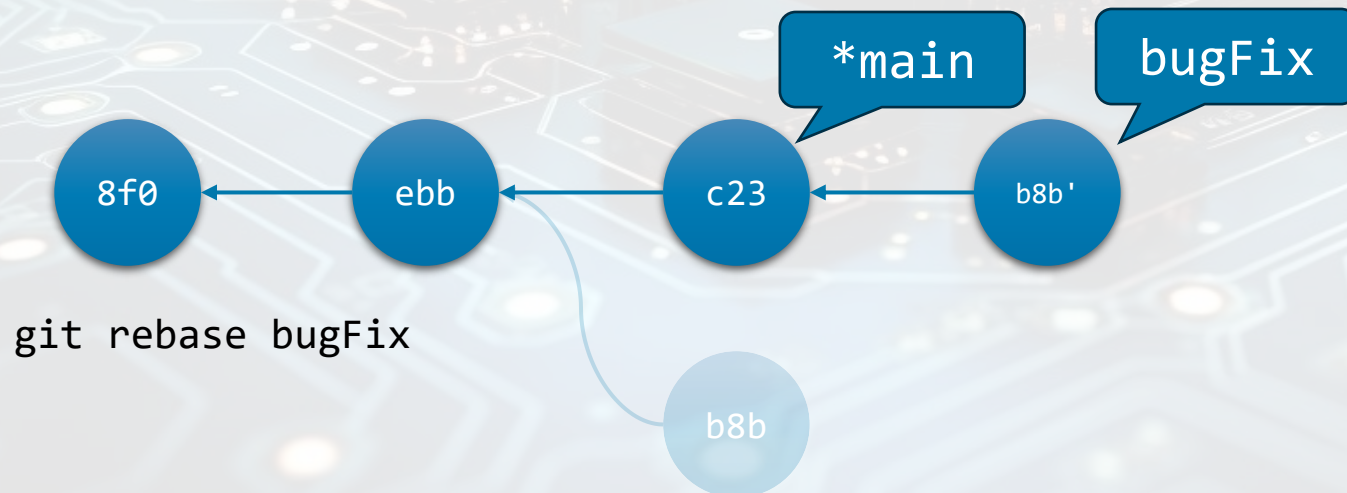
Rebase的目的和Merge是一样的，不同的是它是通过移动当前分支所做的更改实现这一目的。
注意rebase实际上是复制了bugFix记录中的副本放在main上面。



Git Rebase

“快速前进 (Fast forward)”

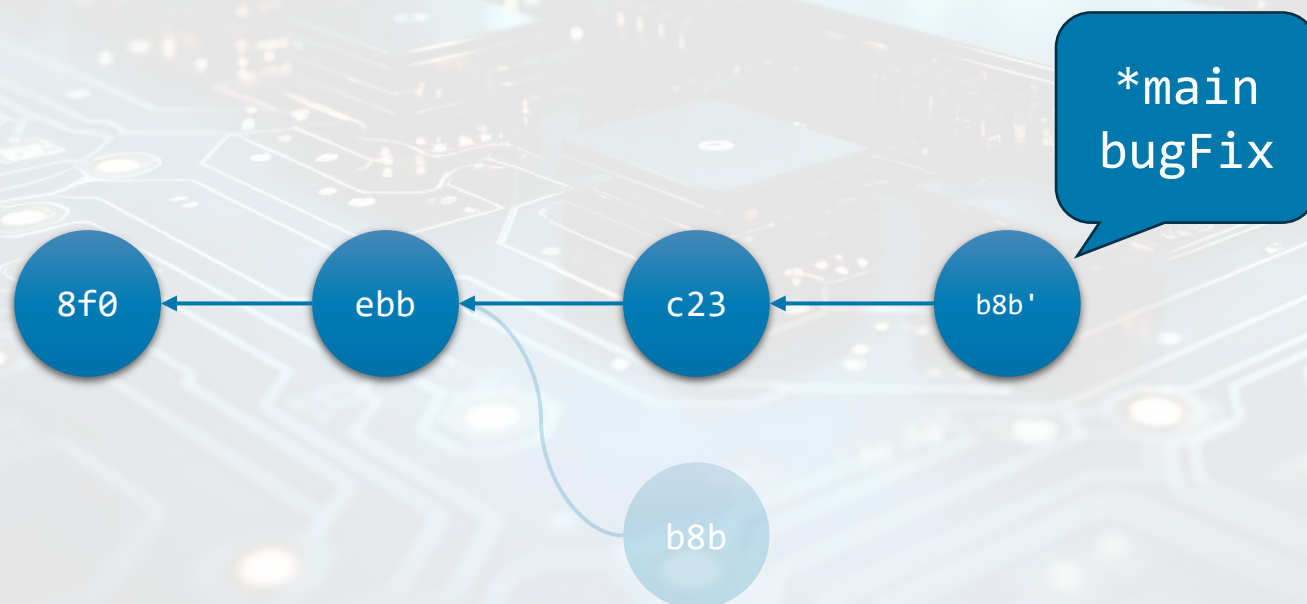
此外，如果可能的话，Rebase也会使用快速前进的技巧。



Git Rebase

“快速前进 (Fast forward)”

此外，如果可能的话，Rebase也会使用快速前进的技巧。



命令速记

`git rebase <名字>`

将当前分支的记录复制到目标分支上。



练习时间!

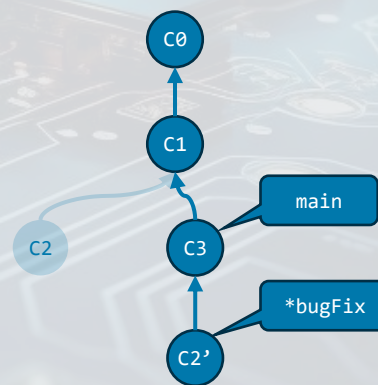
用rebase合并两个分支

打开learngitbranching.js.org, 完成关卡1-4。

目标

1. 创建名为bugFix的分支
2. 依次在bugFix分支和main分支上各提交一次
3. 将bugFix用rebase放到main分支上

最终效果图



本课总结

Git基本用法

命令	用途
git add .	暂存所有文件
git commit	提交一次更改
git branch <名字>	在当前位置创建新分支
git checkout <名字>	切换到目标分支 带-b参数则为同时创建并切换
git merge <名字>	合并目标分支到当前分支
git rebase <名字>	将当前分支的记录复制到目标分支上



Thank You



JerryHan3



西科大学 计算机科学学院
SICO UNIVERSITY