Git笔记

# Git基础要点

## Git与SVN区别：

* SVN是集中式的，必须要一个服务器，通过网络协议连接到服务器，而Git可以不需要服务器。
* SVN可以看成只有2个层次，一个服务器仓库，一个本地工作区，每次提交、分支、合并等操作都同步到服务器。而Git可以看成4个层次，一个服务器仓库（如果有），一个本地仓库，一个暂存区，一个工作区，每次提交、分支、合并等操作都可以只在本地进行。
* Git使用reset回退版本、rebase重整版本后，可以消除一些不需要的版本记录，结合第二点，可以尽量缩减提交数量，达到减小仓库体积的目的【未实践过】。SVN对此支持未知

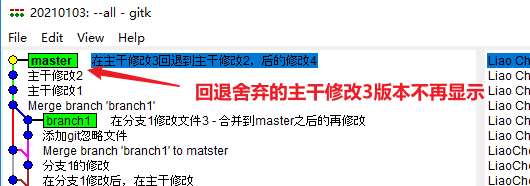
## Gti远程更新后，如果不是跟踪分支，需要进行合并才能更新

## Detached branch是指当前HEAD不在本地分支，可能在远程分支

## git revert 和 git reset的区别

- git revert是用一次新的commit来回滚之前的commit，此次提交之前的commit都会被保留；

- git reset是回到某次提交，提交及之前的commit都会被保留，但是此commit id之后的修改都会被删除。如下图



## 跟踪分支

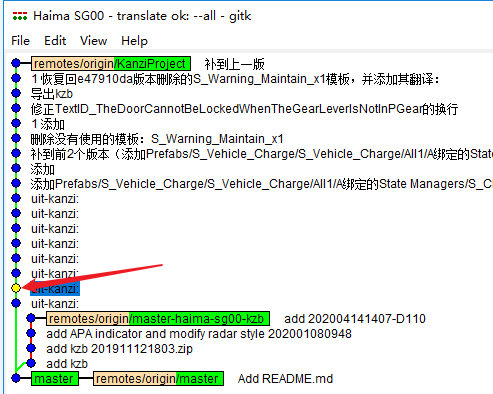
【来自git文档：

<https://www.git-scm.com/book/zh/v2/Git-%E5%88%86%E6%94%AF-%E8%BF%9C%E7%A8%8B%E5%88%86%E6%94%AF>】

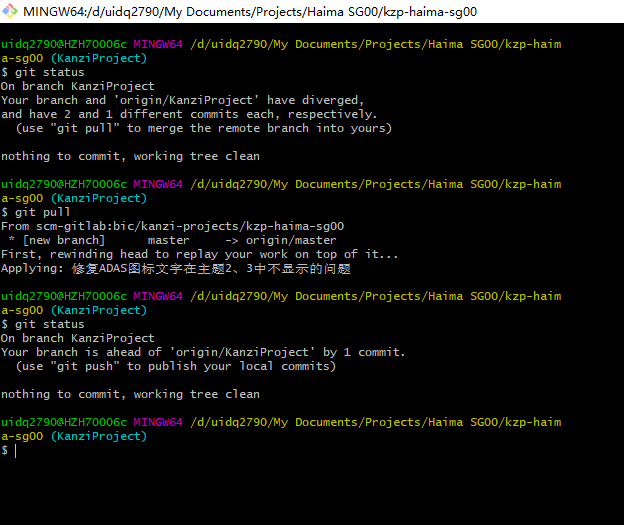
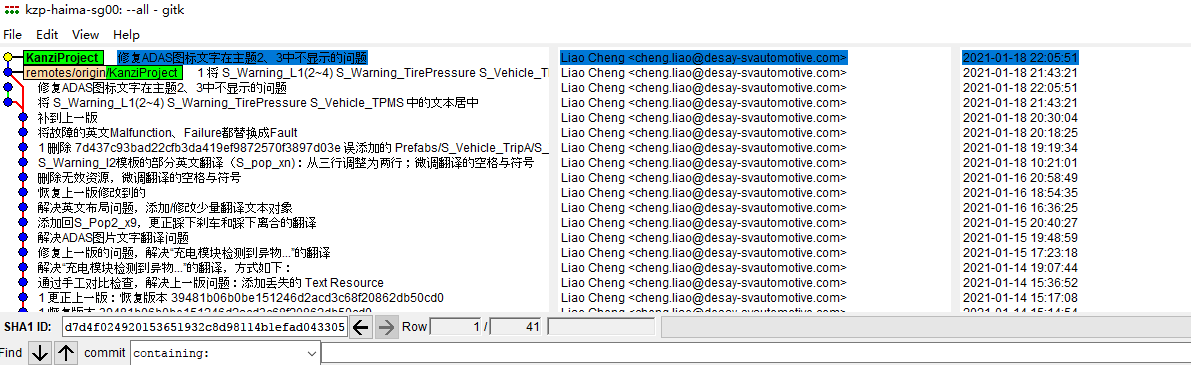
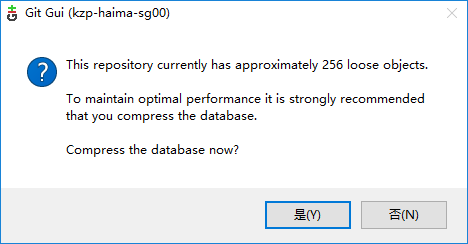
从一个远程跟踪分支检出一个本地分支会自动创建所谓的“跟踪分支”（它跟踪的分支叫做“上游分支”）。 跟踪分支是与远程分支有直接关系的本地分支。 如果在一个跟踪分支上输入 git pull，Git 能自动地识别去哪个服务器上抓取、合并到哪个分支。【本地分支没有设置跟踪的远程分支时，不能只使用git pull，需要使用完整的git pull <remote> <localBranch>】

# Git GUI

## Git GUI的版本图中，黄色点代表当前缓存版本



# 疑问



上图文字：

uidq2790@HZH70006c MINGW64 /d/uidq2790/My Documents/Projects/Haima SG00/kzp-haima-sg00 (KanziProject)

$ git status

On branch KanziProject

Your branch and 'origin/KanziProject' have diverged,

and have 2 and 1 different commits each, respectively.

(use "git pull" to merge the remote branch into yours)

nothing to commit, working tree clean

uidq2790@HZH70006c MINGW64 /d/uidq2790/My Documents/Projects/Haima SG00/kzp-haima-sg00 (KanziProject)

$ git pull

From scm-gitlab:bic/kanzi-projects/kzp-haima-sg00

\* [new branch] master -> origin/master

First, rewinding head to replay your work on top of it...

Applying: 修复ADAS图标文字在主题2、3中不显示的问题

uidq2790@HZH70006c MINGW64 /d/uidq2790/My Documents/Projects/Haima SG00/kzp-haima-sg00 (KanziProject)

$ git status

On branch KanziProject

Your branch is ahead of 'origin/KanziProject' by 1 commit.

(use "git push" to publish your local commits)

nothing to commit, working tree clean

uidq2790@HZH70006c MINGW64 /d/uidq2790/My Documents/Projects/Haima SG00/kzp-haima-sg00 (KanziProject)

$