# git冲突检测：

1. Git 中操作有三种：添加文件，删除文件，修改文件。
2. 关于git cherry-pick 时的冲突检测条件：
   1. 如果在branch A 中pick的commit B操作是添加文件，而且这个文件还不存在于现在的branch A，则不会有冲突。
   2. 如果在branch A 中pick的commit B操作是修改文件，则要参考这个commit B值之前镜像中的该文件是否与branch A中的一样。如果一样，则没有冲突，如果不一样，则有冲突。就是在replay一个commit是要看看replay的起点是否一样，一样才行，否则失败。
   3. 如果在branch A中pick的commit B操作是删除文件， 则和修改文件差不多。要参考这个commit B值之前镜像中的该文件是否与branch A中的一样。如果一样，则没有冲突，如果不一样，则有冲突。就是在replay一个commit是要看看replay的起点是否一样，一样才行，否则以冲突而失败。
3. git merge 命令冲突检测条件和git cherry-pick应该是一致的。
4. 新发现，对于ZigBee工程配置文件（感觉是xml格式的文件）中，即使要cherry pick的父节点commit与现在的commit中的文件即使完全不一样，同样无冲突的cherry pick了。值得思考。对于普通文本却不行！
   1. 猜测是因为xml文件格式规范，git可以理解文件中变量的变化，当cherry pick需要对比文件时，不再是基于文件（line的）添加删除和修改，而是基于xml里面的元素或变量如何变化的。--待分析
   2. 刚刚又深一步实验测试，没有发现git 对待ZigBee工程文件和一般文件有差异，git 并不能理解文件中变量的变化。99.9%的肯定，当cherry pick需要对比文件时，仍然和一般文件一样，基于文件级别的添加删除和更改。之所以在工程文件中出现了”cherry pick的父节点commit与现在的commit中的文件不完全一样，却直接replay 这个commit“，可能因为git的bug！！！
5. 总结可得：
   1. Cherry pick 涉及3个commit；（类似[three-way merge](http://en.wikipedia.org/wiki/Three-way_merge), 但不全一样）
   2. 三个commit分别是：
      1. 被picked 的parent commit：commit1
      2. 被picked 的commit： commit2
      3. 发起cherry pick 的commit：commit3
   3. 则对于一个file1被标记为冲突的条件是：file1在三个commit中**内容**互不相同 (若不存在file1，视为存在一个内容为空的file1处理) ！！！！