



Git in Reality

- A real merge

- Fixing up a real merge

Info:

A long time ago (about a year) I made some changes to EPOCH to check success after every memory allocation. These changes have sat un-merged since then, as they weren't suitable for main-line release. The main-line code has changed substantially, in particular:

a change from a hard-coded '3' to a constant 'ng'

new compiler defines added

tweaks to load balancing which change the size/shape of array needed

Try:

Pull the branches workshop/merge/main and workshop/merge/feature

Have a quick look at the diffs between the branches and understand the intention

Attempt to merge feature onto main

Fix conflicts

Fix any errors

Try not to cry

Bonus: give up, re-create changes from feature on top of main by hand

- Take Home Message

- If possible, avoid these sorts of merge situations
 - Try not to work on code somebody else is changing
 - Don't go too long without merging "devel" (or here, main) so you're not having to tackle all the changes at once, when you might have forgotten what your own changes do
 - Where possible and practical, make changes in small sections and get each *useful* part merged into the main code as you work
- If you have to tackle a giant merge make sure the result WORKS
 - Does it compile?
 - Does it run and pass tests?
 - If changes were to fix a bug, is it gone?
 - If changes added a feature, does that work?