Subject: Re: GIT essentials
Posted by andrei_natanael on Mon, 11 Jan 2010 16:59:32 GMT
View Forum Message <> Reply to Message

mr_ped wrote Well, the problem is that cloning(branching) with GIT is cheap, while with SVN it will take those 5+min. That's not good.

Actually branching in git takes almost not time.
real 0m0.044s
user 0m0.000s
sys 0m0.010s

Cloning the main repository takes 1 minute and equivalent of clone is checkout in svn I guess(in git checkout is a different operation).
mr_ped wrote
Also I think if you are proposing this, you should maybe try also design the work-flow model, i.e. how Mirek will remain the master of U++, yet other contributors will submit patches with GIT to him. So once he will want to try it out, he can read some ideas how it should work in U++ community. (because GIT allows many ways of cooperation)

Mirek will remain the master of U++. We may follow Linus work-flow, Mirek as dictator and perhaps if U++ will grow more other persons as lieutenants. Right now Mirek being single person who will decide what goes in repo is the right choice.
mr_ped wrote
What's puzzling *me* as non-user of git is the "pull". I understand anyone can clone repo, do his changes, prepare public commit (patch) and tell maintainer of project it's really great improvement and he should adopt it. Then comes the "pull" by maintainer from the contributor's repo? So everyone's personal repository has to be on public IP? (I find this unlikely with current U++ contributors)
I know this can be worked around by submitting patches for example trough e-mail, or by letting contributor to instead push into central repo, I'm just asking if I understand this part correctly. [...] Maybe I misunderstood something important about DVCS/GIT?
You shouldn't have a public IP, for that exists sites like gitorious, googlecode, sourceforge. If someone wants to let Mirek see his changes he will push changes to his own repo from one of those sites and send to Mirek a link to it.
Perhaps if we take a realistic situation it will be more easier to understand that. I have a clone of U++ at gitorious, and also a local clone of if. If i will make changes to local clone and merge changes with the master branch from gitorious my repo will diverge from current U++ repo (supposing it's using git). Now i may send to Mirek a link to my repo, he will clone it (or only clone last revision) and if he like it he will merge mine repo with U++ main repo. I don't have to tell to anyone my IP (my IP is dynamic anyway and telling one will have no effect next time because it may get changed). No need to use a central repo, no emailed patches.

I don't understand yet very well git but as i understand now it makes more sense to me to use it than using svn. If I'll make changes to Chameleon with git i have to do simple operations:
git branch cham-changes
git checkout cham-changes
// change the code

git commit -a
git checkout master
// if i'm glad with changes

git merge cham-changes
git commit -a
git branch -d cham-changes

// if i'm not pleased with changes

git branch -D cham-changes

With svn i'll have to change the files... if i dislike the changes... revert them... in the meantime i cannot work on something else because the repo is blocked by my changes to Chameleon. To solve this i have to use branches in svn too, it take time and i don't know how hard is to merge them... perhaps someone with merging experience will tell me. In git merging is so simple .