PH502: Scientific Programming Concepts

Irish Centre for High End Computing (ICHEC)

September 23, 2020

Overview

- The second useful utility is version control.
- If many people are working on a project then it is important that everyone has access to the latest version.
- Also if mistakes are made or files deleted, files can be rolled back or replaced.
- Many online software teams use version control.

Version Control

Git

- Version control is especially important when software is being maintained by a group.
- Each author has a local set of files and a snap shot of the files contained in a repository. If changes are made to the files then they differ from the repository snap shot. The repository can be updated giving a new snap shot. A record of modifications are maintained within the repository enabling a return to previous versions.
- There are a number of version control packages: cvs, svn, git, mercurial, bazaar, monotone and libresource. At ICHEC we use git.
- Here we will focus on the command line interface to "git". A git repository can be created within a directory. All files and subdirectories that are part of the repository will be version controlled.

Git

- Below is a table of some git commands. These can be used to maintain a local git repository.

Command	Function
git init	create repository
git add/rm	add remove files in repo
git commit	commit the modifications to the repo
git tag	tag the current committed version
git checkout	take file from repo
git log	check the committed changes to the repo
git diff	see changes from the repo version
git status	shows file changes from repo

Remote Repository

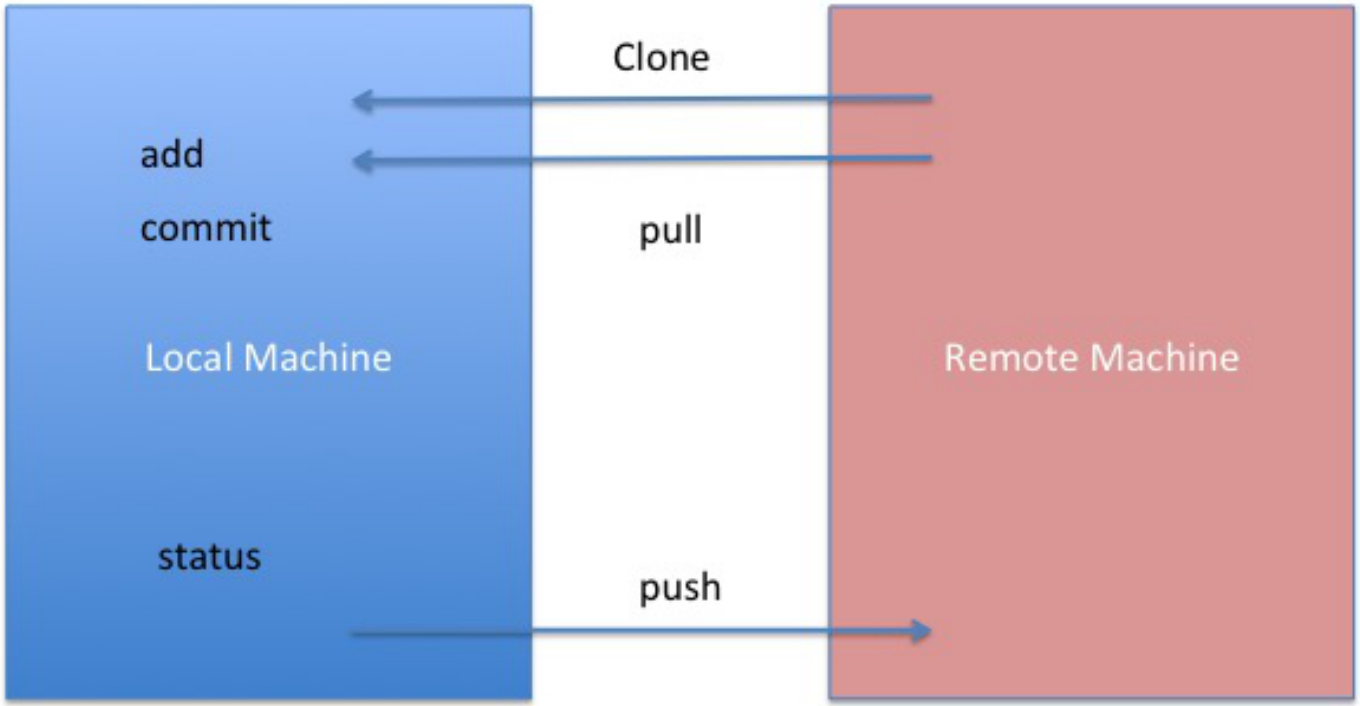
- It is possible to have the repository in a different location. In this case both the local repository and remote one need to be aligned.
  Additional commands are:

Command	Function
git clone	clone central repo locally
git pull	get latest version from central repo
git push	synchronize local repo with centre

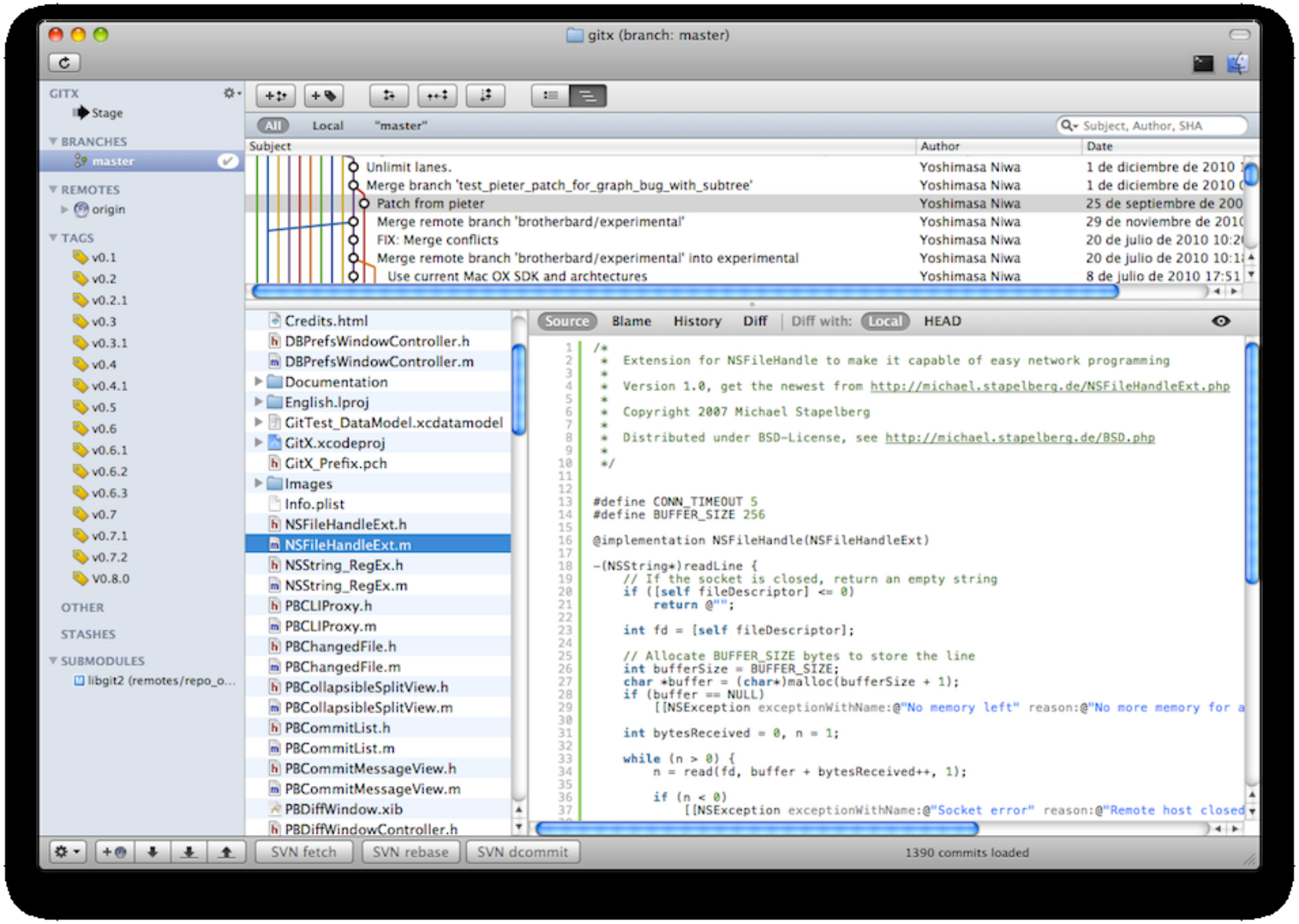
- An example of using git

```
# Make sure up to date with remote repo
git pull
# Edit prog.c and make but changes cause error
# Return to the original
git checkout prog.c
# Edit and make, this time successful
# Commit changes and push to remote repo
git commit -m "Fixed bug" prog.c
git push origin master
```

Visualization



Git Gui


gitx (branch: master)
GITX
Stage
BRANCHES
master
REMOTES
origin
TAGS
v0.1
v0.2
v0.2.1
v0.3
v0.3.1
v0.4
v0.4.1
v0.5
v0.6
v0.6.1
v0.6.2
v0.6.3
v0.7
v0.7.1
v0.7.2
V0.8.0
OTHER
STASHES
SUBMODULES
libgit2 (remotes/repo_o...
All
Local
"master"
Subject, Author, SHA
Subject
Author
Date
Unlimit lanes.
Merge branch 'test_pieter_patch_for_graph_bug_with_subtree'
Patch from pieter
Merge remote branch 'brotherbard/experimental'
FIX: Merge conflicts
Merge remote branch 'brotherbard/experimental' into experimental
Use current Mac OX SDK and archtectures
Yoshimasa Niwa
1 de diciembre de 2010
25 de septiembre de 200
29 de noviembre de 201
20 de julio de 2010
8 de julio de 2010 17:51
Credits.html
DBPrefsWindowController.h
DBPrefsWindowController.m
Documentation
English.lproj
GitTest_DataModel.xcdatamodel
GitX.xcodeproj
GitX_Prefix.pch
Images
Info.plist
NSFileHandleExt.h
NSFileHandleExt.m
NSString_RegEx.h
NSString_RegEx.m
PBCLIProxy.h
PBCLIProxy.m
PBChangedFile.h
PBChangedFile.m
PBCollapsibleSplitView.h
PBCollapsibleSplitView.m
PBCommitList.h
PBCommitList.m
PBCommitMessageView.h
PBCommitMessageView.m
PBDiffWindow.xib
PBDiffWindowController.h
Source
Blame
History
Diff
Diff with:
Local
HEAD
/*
* Extension for NSFileHandle to make it capable of easy network programming
* Version 1.0, get the newest from http://michael.stapelberg.de/NSFileHandleExt.php
* Copyright 2007 Michael Stapelberg
* Distributed under BSD-License, see http://michael.stapelberg.de/BSD.php
*/
#define CONN_TIMEOUT 5
#define BUFFER_SIZE 256
@implementation NSFileHandle(NSFileHandleExt)
-(NSString*)readLine {
// If the socket is closed, return an empty string
if ([self fileDescriptor] <= 0)
return @"";
int fd = [self fileDescriptor];
// Allocate BUFFER_SIZE bytes to store the line
int bufferSize = BUFFER_SIZE;
char *buffer = (char*)malloc(bufferSize + 1);
if (buffer == NULL)
[[NSException exceptionWithName:@"No memory left" reason:@"No more memory for a
int bytesReceived = 0, n = 1;
while (n > 0) {
n = read(fd, buffer + bytesReceived++, 1);
if (n < 0)
[[NSException exceptionWithName:@"Socket error" reason:@"Remote host closed
SVN fetch
SVN rebase
SVN dcommit
1390 commits loaded

Git Resources

ICHEC
Irish Centre for High-End Computing

- on the command line: 'git –help' or 'man git'
- Git documentation: git-scm.com/documentation
- Online courses at Udacity, Coursera
- GitLab: free private git repository host