Oh my GIT!

Git ate my work and other stories

Vojtěch Vladyka

December 2, 2019

Table of contents

What is git?

- version control system made by Linus Torvalds (mostly) at 2005 for Linux Kernel development
- always capture state of working tree
- focused on distributed development
- supports rapid branching & merging
- it is pronounced GIT with G like GIF[1]

[1] Google. *Tech Talk: Linus Torvalds on git*. URL: `https://www.youtube.com/watch?v=4XpnKHJAok8`.

What is git?

Differences between Git and SVN

- Git is distributed by design
- Works with whole sourcetree (unlike SVN which works with individual files and their revisions)
- Strong support for non-linear development $\rightarrow$ rapid branching & merging
- Cryptographic authentication of history - every commit has SHA-1 hash of everything leading to that point

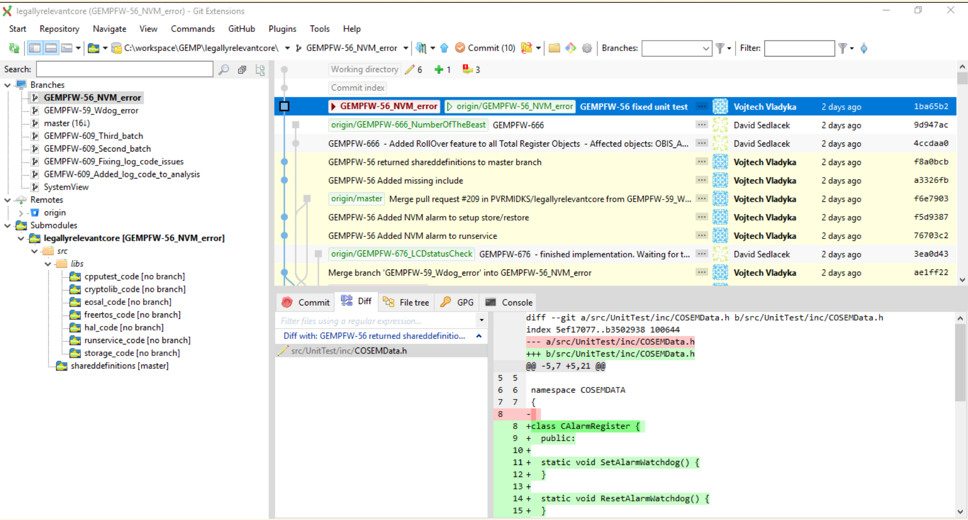
Setup git

Git clients

1. Git (all platforms)
2. Git Extensions (Windows, Mac)
3. Tortoise Git (Windows)
4. Sourcetree (Windows)
5. Fork (Windows, Mac)
6. Gitkraken (all platforms)
7. ...many others

Setup git

Git Extensions (Git client)


legallyrelevantcore (GEMPFW-56_NVM_error) - Git Extensions
Start Repository Navigate View Commands GitHub Plugins Tools Help
C:\workspace\GEMP\legallyrelevantcore\
GEMPFW-56_NVM_error
Commit (10)
Branches:
Filter:
Search:
Commit
Diff
File tree
GPG
Console
Filter files using a regular expression...

Branches
- GEMPFW-56_NVM_error
- GEMPFW-59_Wdog_error
- master (16↓)
- GEMPFW-609_Third_batch
- GEMPFW-609_Second_batch
- GEMPFW-609_Fixing_log_code_issues
- GEMFW-609_Added_log_code_to_analysis
- SystemView

Remotes
- origin

Submodules
- legallyrelevantcore [GEMPFW-56_NVM_error]
  - src
    - libs
      - cpputest_code [no branch]
      - cryptolib_code [no branch]
      - eosal_code [no branch]
      - freertos_code [no branch]
      - hal_code [no branch]
      - runservice_code [no branch]
      - storage_code [no branch]
  - shareddefinitions [master]

Working directory 6 +1 3			
Commit index			
GEMPFW-56_NVM_error origin/GEMPFW-56_NVM_error GEMPFW-56 fixed unit test	Vojtech Vladyka	2 days ago	1ba65b2
origin/GEMPFW-666_NumberOfTheBeast GEMPFW-666	David Sedlacek	2 days ago	9d947ac
GEMPFW-666 - Added RollOver feature to all Total Register Objects - Affected objects: OBIS_A...	David Sedlacek	2 days ago	4ccdaa0
GEMPFW-56 returned shareddefinitions to master branch	Vojtech Vladyka	2 days ago	f8a0bcb
GEMPFW-56 Added missing include	Vojtech Vladyka	2 days ago	a3326fb
origin/master Merge pull request #209 in PVRMIDKS/legallyrelevantcore from GEMPFW-59_W...	Vojtech Vladyka	2 days ago	f6e7903
GEMPFW-56 Added NVM alarm to setup store/restore	Vojtech Vladyka	2 days ago	f5d9387
GEMPFW-56 Added NVM alarm to runservice	Vojtech Vladyka	2 days ago	76703c2
origin/GEMPFW-676_LCDstatusCheck GEMPFW-676 - finished implementation. Waiting for t...	David Sedlacek	2 days ago	3ea0d43
Merge branch 'GEMPFW-59_Wdog_error' into GEMPFW-56_NVM_error	Vojtech Vladyka	2 days ago	ae1ff22

Diff with: GEMPFW-56 returned shareddefinitio...

src/UnitTest/inc/COSEMData.h

```
diff --git a/src/UnitTest/inc/COSEMData.h b/src/UnitTest/inc/COSEMData.h
index 5ef17077..b3502938 100644
--- a/src/UnitTest/inc/COSEMData.h
+++ b/src/UnitTest/inc/COSEMData.h
@@ -5,7 +5,21 @@

 namespace COSEMDATA
 {
-
+class CAlarmRegister {
+  public:
+
+  static void SetAlarmWatchdog() {
+  }
+
+  static void ResetAlarmWatchdog() {
+  }
```

Download it here → `https://gitextensions.github.io`

Git essentials

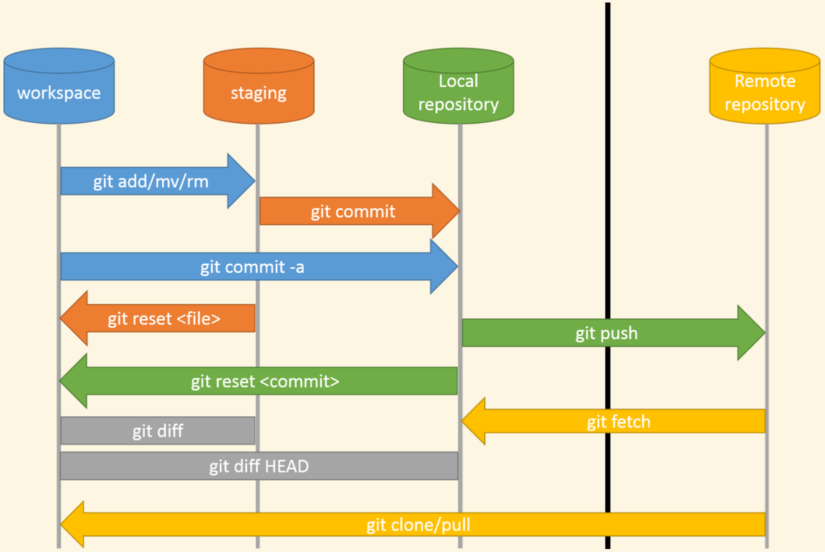
Short and incomplete command overview

```
git init
    clone

    fetch
    merge
    push
    pull

    branch
    checkout
    merge
    branch -d
```

```
git log
    log --follow [file]
    reflog

    add
    commit

    reset
    reset --hard
    stash

    rebase
    cherrypick
```

Git essentials

Workflow overview



Git essentials
Init repo

```
git init
```

→ initialize new repository in current directory with all its contents

```
git clone https://somerepo.com/repo.git
```

→ downloads repository with all current branches & commits

Git essentials

Fetching changes and returning them back

```
git pull
```

→ Download commits from server and apply them to your working tree

```
git push
```

→ Upload your commits to server

```
git fetch --all
```

→ Download commits from server (from all branches eventually)

```
git commit [files] -m "Commit message"
```

→ Create a commit with commit message and specified files
→ You can use -a to commit all changes instead list of files

```
git add [file]
```

→ Move file to staging area - add it to commit list

Git essentials
Undoing changes

```
git reset
```

→ Soft reset - remove files from staging area and move them back to working area. Nothing is actually reverted, this resets it only for git.

```
git reset --hard
```

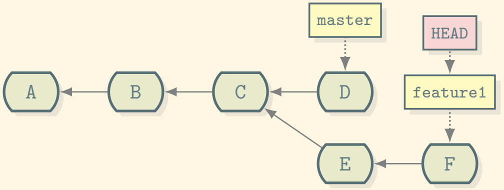
→ Hard reset - this do all what soft reset but also REMOVES your uncommitted changes.

→ You can do this only locally. Once you push your changes, you cannot get it back. (not exactly true but you will break a lot of thing for all colleagues). Only way how to propperly revert things it using git revert.

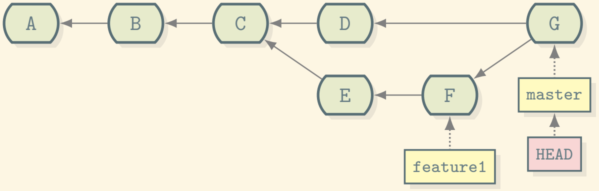
Git essentials

Branching

```
git checkout -b feature1
(git branch feature1 ; git checkout feature1)
```



```
git checkout master
git merge feature1
```



Git essentials

Logging and searching lost stuff

```
git log --graph
```

→ Shows log for whole repository tree

```
git log --follow [file]
```

→ Shows log for selected file even across renames

```
git reflog
```

→ Shows log of ALL changes in repository. Stashes, detached branches, all...

Git essentials

Stashing - let me just put this here for later (and never use it again)

```
git stash
git stash push
```

→ Stashes current changes to new stash stack

```
git stash pop
```

→ Pops newest stash and apply it to current working copy

```
git stash list
```

→ Show all stashes in list

```
git stash apply stash@{2}
```

→ Get stash #2 and apply it to current working copy

```
git checkout stash@{2} -- somefile
```

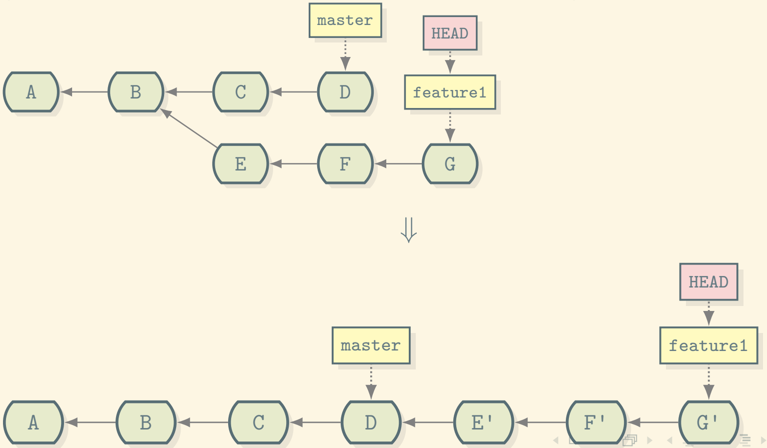
→ Get stash #2 and apply it to specified files in current working copy

Git essentials

Rebase

```
git rebase target_branch source_branch
```

→ Take source branch and reattach it on end of target branch.
This creates series of new commits and remove old ones from tree (althrough they will still be accessible by reflog)

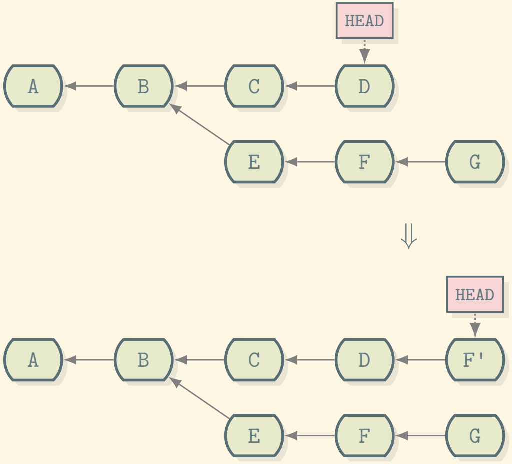


Git essentials

Cherry picking

```
git cherry-pick <commit hash>
```

→ Select commit by hash and copy it on end of current branch as new commit.



Scenarios

It's time to break some stuff.

Resources

- Google. *Tech Talk: Linus Torvalds on git*. URL: https://www.youtube.com/watch?v=4XpnKHJAok8.
- Mark Erikson. *Git Under the Hood*. URL: https://blog.isquaredsoftware.com/presentations/2019-03-git-internals-rewrite/#/0.
- GitHub. *Git Cheat Sheet*. URL: https://github.github.com/training-kit/downloads/github-git-cheat-sheet.pdf.
- Git community. *Git –distributed-even-if-your-workflow-isnt*. URL: https://git-scm.com.

Q & A

Thank you for your attention.