# Git

**Step 1** − You modify a file from the working directory.

**Step 2** − You add these files to the staging area.

**Step 3** − You perform commit operation that moves the files from the staging area. After push operation, it stores the changes permanently to the Git repository.



Modified sort.c and commit

git add sort.c

git commit –m “Added sort operation”

Branch

By default, Git has a master branch, which is same as trunk in Subversion. Usually, a

branch is created to work on a new feature. Once the feature is completed, it is merged

back with the master branch and we delete the branch.

Clone

Clone operation creates the instance of the repository. Mirror the complete repository.

### **Pull**

Pull operation copies the changes from a remote repository instance to a local one.

The pull operation is used for synchronization between two repository instances.

Push

Push operation copies changes from a local repository instance to a remote one.

This is used to store the changes permanently into the Git repository.

Configuration

git config --global user.name "Jerry Mouse"

git config --global user.email "jerry@tutorialspoint.com"

### **Avoid merge commits for pulling**

You pull the latest changes from a remote repository, and if these changes are divergent, then by default Git creates merge commits. We can avoid this via following settings.

git config --global branch.autosetuprebase always

Workflow

* You clone the Git repository as a working copy.
* You modify the working copy by adding/editing files.
* If necessary, you also update the working copy by taking other developer's changes.
* You review the changes before commit.
* You commit changes. If everything is fine, then you push the changes to the repository.
* After committing, if you realize something is wrong, then you correct the last commit and push the changes to the repository.



## Push Changes to the Repository

For user 1:

$ pwd

/home/tom

$ mkdir tom\_repo

$ cd tom\_repo/

$ git init

Initialized empty Git repository in /home/tom/tom\_repo/.git/

#To add file A

# show files present in the staging area

$ git status -s

?? README

$ git add .

$ git status -s

A README

$ git commit -m 'Initial commit'

[master (root-commit) 19ae206] Initial commit

1 files changed, 1 insertions(+), 0 deletions(-)

create mode 100644 README

$ git log

commit 19ae20683fc460db7d127cf201a1429523b0e319

Author: Tom Cat <tom@tutorialspoint.com>

Date: Wed Sep 11 07:32:56 2013 +0530

Initial commit

#File added to local repo now

# One time operation to add remote repo – get link from gitweb

$ git remote add origin [gituser@git.server.com:project.git](mailto:gituser@git.server.com:project.git)

$ git push origin master

For user 2:

$ mkdir jerry\_repo

$ cd jerry\_repo/

#Clone repo

$ git clone gituser@git.server.com:project.git

Initialized empty Git repository in /home/jerry/jerry\_repo/project/.git/

remote: Counting objects: 3, done.

Receiving objects: 100% (3/3), 241 bytes, done.

remote: Total 3 (delta 0), reused 0 (delta 0)

$ cd project/

$ ls

README

#Modified and going to update string.c file

$ git status -s

?? string

?? string.c

$ git add string.c

$ git status -s

A string.c

?? string

$ git commit -m 'Implemented my\_strlen function'

[master cbe1249] Implemented my\_strlen function

1 files changed, 24 insertions(+), 0 deletions(-)

create mode 100644 string.c

#He likes to change a file committed. He didn’t do push yet

$ git log

commit cbe1249b140dad24b2c35b15cc7e26a6f02d2277

Author: Jerry Mouse <jerry@tutorialspoint.com>

Date: Wed Sep 11 08:05:26 2013 +0530

Implemented my\_strlen function

#To view commit details

$ git show cbe1249b140dad24b2c35b15cc7e26a6f02d2277

**<File contents>**

**<'+'** sign before lines, which are newly added and **'−'** for deleted lines.>

#To review changes

$ git diff

< sample output :

- printf("string lenght of %s = %d\n", s[i], my\_strlen(s[i]));

+ printf("string lenght of %s = %lu\n", s[i], my\_strlen(s[i]));

>

#The amend operation changes the last commit including your commit message; it creates a new commit ID.

[jerry@CentOS project]$ git commit --amend -m 'Changed return type of my\_strlen to size\_t'

[master d1e19d3] Changed return type of my\_strlen to size\_t

1 files changed, 24 insertions(+), 0 deletions(-)

create mode 100644 string.c

$ git log

commit d1e19d316224cddc437e3ed34ec3c931ad803958

Author: Jerry Mouse <jerry@tutorialspoint.com>

Date: Wed Sep 11 08:05:26 2013 +0530

Changed return type of my\_strlen to size\_t

commit 19ae20683fc460db7d127cf201a1429523b0e319

Author: Tom Cat <tom@tutorialspoint.com>

Date: Wed Sep 11 07:32:56 2013 +0530

Initial commit

#Before push operation, he wants to review his changes, so he uses the **git show** command to review his changes.

$ git show d1e19d316224cddc437e3ed34ec3c931ad803958

< Contents >

$ git push origin master

Counting objects: 4, done.

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 517 bytes, done.

Total 3 (delta 0), reused 0 (delta 0)

To gituser@git.server.com:project.git

19ae206..d1e19d3 master −> master

**Modify delivered file by another user.**

$ git clone gituser@git.server.com:project.git

$ cd project/

$ git log

$ git diff

$ git status -s

$ git add string.c

$ git commit -m 'Changed char pointer to const char pointer'

$ git log

$ git push origin master

**Meanwhile another user wants to add new function but local file is not updated/sync with remote repo**

**If he tries to push**

$ git push origin master

To gituser@git.server.com:project.git

! [rejected]

master −> master (non-fast-forward)

error: failed to push some refs to 'gituser@git.server.com:project.git'

**The user have to update his local repo first and then push it.**

**Get latest contents and then push**

$ git pull

$ git push origin master

Stash operation. Keep changes and commit later.

$ git status -s

M string.c

?? string

$ git stash

Saved working directory and index state WIP on master: e86f062 Added my\_strcpy function

HEAD is now at e86f062 Added my\_strcpy function

$ git status -s

?? string

$ git stash list

stash@{0}: WIP on master: e86f062 Added my\_strcpy function

Now try to commit save changes

$ git stash pop

Move operation

$ pwd

/home/tom/project

$ ls

README string string.c

$ mkdir src

$ git mv string.c src/

$ git status -s

R string.c −> src/string.c

?? string

To make changes permanent

$ git commit -m "Modified directory structure"

$ git push origin master

If other user pull the folders, it will get auto updated in their local repo.

$ ls

README string string.c

$ git pull

$ ls

README src string

$ ls src/

string.c