

Repository Synchronizer

Submitted by

K.Avinash
P.Ram
Naveen
P.Chaitanya
Revanth

CSE-B

Problem Statement:

So far we have been doing all our tasks locally. This means that all the changes that are committed to the local machine only. So, other developers have neither idea nor access to our work. Other developers in the same team should be able to view and contribute to the project at the same time without overwriting each other's work. To make it possible we can use Repository Synchronization where we use remote repositories.

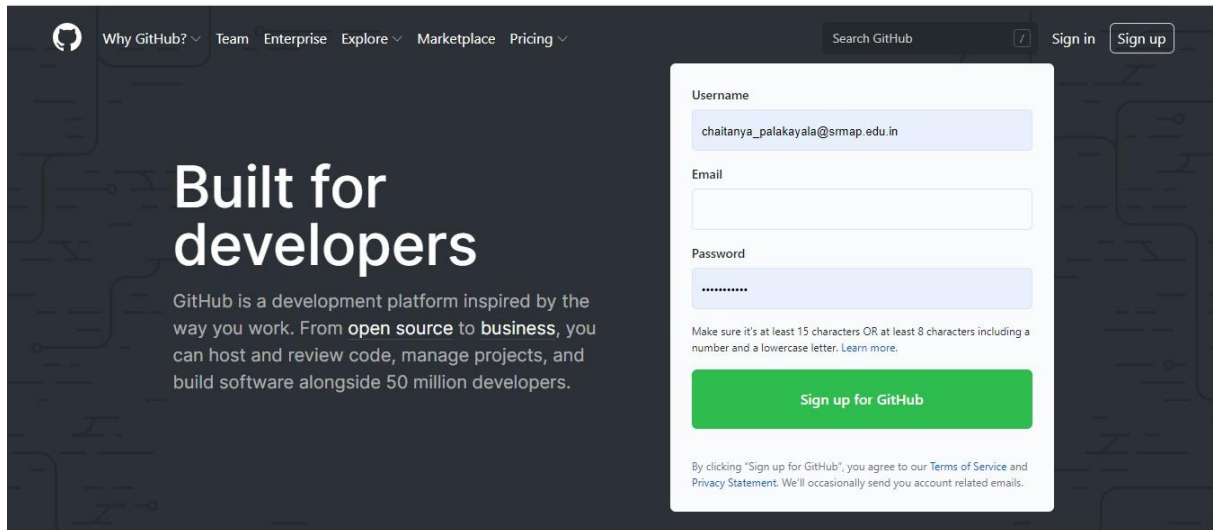
Introduction:

Repository is a storage location for storing files, documents, programs, projects safely etc. Process Synchronization means sharing system resources by processes in such a way that, Concurrent access to shared data is handled thereby minimizing the chance of inconsistent data. Here we are going to do Repository Synchronization. Repository Synchronization is a process of ensuring the files, documents etc. in repository are updated via certain rules (like no overwriting). Here we take GitHub as our server and users as clients.

Steps to show how Repository Synchronization takes place in Client Server application (GitHub):

1. Create GitHub account:

Here GitHub is the server and client is the user.



2. Create a new repository in GitHub.

Repository: it is a place to store important files, documents, etc.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

 chaitanyaPc ▾

Repository name *

/ OSProject. 

Great repository names are short and memorable. Need inspiration? How about **redesigned-tribble**?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

3. Create a Local Folder on your desktop. Insert files into your local folder to insert into your repository.

4. Open gitbash and go to your local folder directory by using “change directory” command
(Cd directory_name)

Steps to Follow:

Step1: cd desktop.

Step2: cd Oproject.

Step3: git init.

The git init command is the first command that we will run on Git. The git init command is used to create a new blank repository. It is used to make an existing project as a Git project. The git init command creates a .git subdirectory in the current working directory. This newly created subdirectory contains all of the necessary metadata. These metadata can be categorized into objects, refs, and temp files. It also initializes a HEAD pointer for the master branch of the repository.

Step5: git status

The lists of all untracked files are displayed by the git status command.

Step4: git add filename

The git add command is used to add file contents to the Index also known as Staging Area. When we add a file in git, it will take place in the staging area. The staging area is a working area where files are not handled by git. These files are also referred to as "untracked files". Staging area files are the files that are going to be a part of the next commit, which make git to know what changes in the file (previous) that are going to occur for the next commit. We can add single or multiple files at once in the staging area.

To add multiple files:

We use git add.

Step5: git Commit -m “some message”

It is used to record/save the changes in the repository. After adding files we need to use this command. Every commit contains the index data and the commit message.

Step6: git remote add origin

<https://github.com/chaitanyaPc/Osproject.git>

Remote repository: is a common repository that all team members use to exchange their changes.

To **add** a new **remote (repository)**, we use **the git remote add** command on the terminal, in the directory your repository is stored at. We have to store our remote repository URL into remote variable origin.

Step7: git branch -M main

It is used to rename old branch name to new branch name. Here we are renaming master branch as main.

Step8: git push -u origin main

This command helps to create a new tracking connection with origin/main branch. -u helps us to capture the remote branch we intend to track After commit push files to GitHub repository.

Outputs:

```
MINGW64/c/Users/PC/desktop/OSproject

PC@DESKTOP-2Q9GAF3 MINGW64 ~
$ cd desktop

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop
$ cd OSproject

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (master)
$ git init
Reinitialized existing Git repository in C:/Users/PC/Desktop/OSproject/.git/

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    OSlab.pdf

nothing added to commit but untracked files present (use "git add" to track)

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (master)
$ git add OSlab.pdf

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (master)
$
```

```
MINGW64/c/Users/PC/desktop/OSproject

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (master)
$ git commit -m "OS Project"
[master (root-commit) 4a63298] OS Project
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 OSlab.pdf

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (master)
$ git remote add origin https://github.com/chaitanyaPc/OSproject.git
fatal: remote origin already exists.

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (master)
$ git branch -M main

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (main)

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 1.25 MiB | 926.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/chaitanyaPc/OSproject.git
 * [new branch]    main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (main)
$
```

```
MINGW64:/c/Users/PC/desktop/OSproject

PC@DESKTOP-2Q9GAF3 MINGW64 ~
$ cd desktop

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop
$ cd OSproject

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (main)
$ git commit -m "OS Project"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

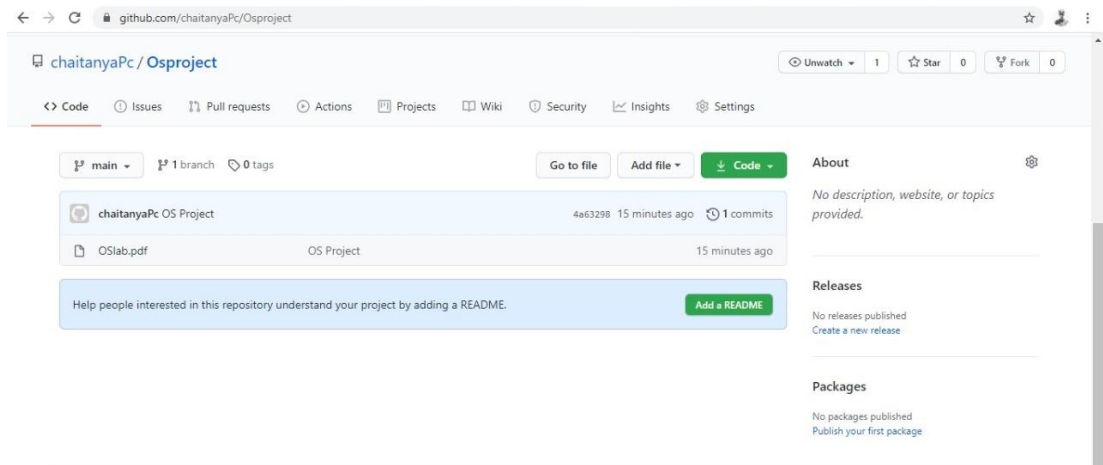
PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (main)
$ git remote add origin https://github.com/chaitanyaPc/OSproject.git
fatal: remote origin already exists.

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (main)
$ git branch -M main

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (main)
$ git push -u origin main
Everything up-to-date
Branch 'main' set up to track remote branch 'main' from 'origin'.

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (main)
$
```

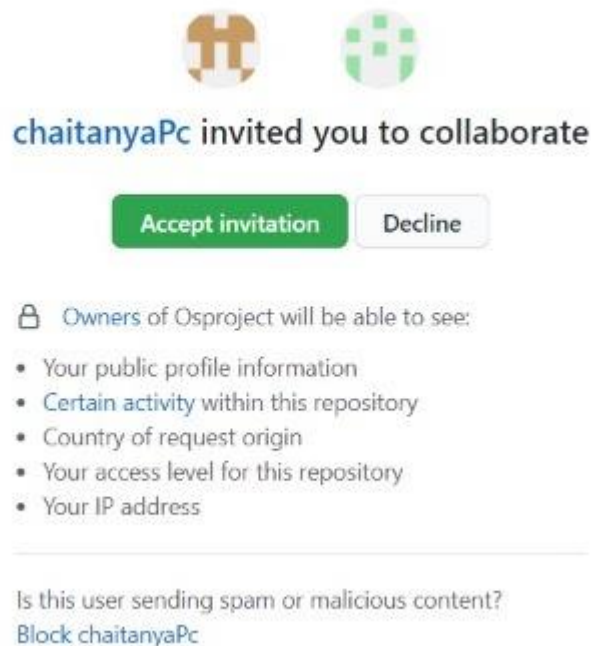
When we use push command it asks for our GitHub Account



In the above output we can see that that files are inserted.

Steps to access others repository/masters repository (cloning):

1. **First collaborate with other users/developers.**



2. **After contributor accepts request from other users, users can clone and modify the master project.**

Steps to follow:

Step 1: git clone URL (repository branch)

The git clone command copies an existing Git repository .git clone is primarily used to point to an existing repo and make a clone or copy of that repo at in a new directory, at another location.

Step 2: git init

Initializing new blank repository.

Step 3: git remote –verbose

From which repository files are to be fetched or to be added. Showing remote repositories.

Step 4: git add.

Adding files.

Step 5: git status

To check how many new files to be committed.

Step 6: git commit -m "OS" To save the changes.

Step 7: git status

Step 8 : git branch -M main

On which repository sub file we are working on.

Step 9 : git push -u origin main

Pushing new changes to the repository

```
MINGW64/c/Users/PC/desktop/OSproject

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Osproject
    new file:   WhatsApp Image 2020-11-26 at 6.59.16 PM.jpeg
    new file:   WhatsApp Image 2020-11-26 at 7.16.28 PM.jpeg
    new file:   WhatsApp Image 2020-11-26 at 7.26.04 PM.jpeg
    new file:   WhatsApp Image 2020-11-26 at 7.26.53 PM.jpeg
    new file:   WhatsApp Image 2020-11-26 at 7.37.18 PM.jpeg
    new file:   WhatsApp Image 2020-11-26 at 7.47.40 PM.jpeg
    new file:   WhatsApp Image 2020-11-26 at 7.51.32 PM.jpeg

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (main)
$ git commit -m "OS"
[main eca52ca] OS
 8 files changed, 1 insertion(+)
 create mode 160000 Osproject
 create mode 100644 WhatsApp Image 2020-11-26 at 6.59.16 PM.jpeg
 create mode 100644 WhatsApp Image 2020-11-26 at 7.16.28 PM.jpeg
 create mode 100644 WhatsApp Image 2020-11-26 at 7.26.04 PM.jpeg
 create mode 100644 WhatsApp Image 2020-11-26 at 7.26.53 PM.jpeg
 create mode 100644 WhatsApp Image 2020-11-26 at 7.37.18 PM.jpeg
 create mode 100644 WhatsApp Image 2020-11-26 at 7.47.40 PM.jpeg
 create mode 100644 WhatsApp Image 2020-11-26 at 7.51.32 PM.jpeg

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (main)
$ git branch -M main

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/OSproject (main)
$ git push -u origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 392.36 KiB | 14.53 MiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
```


MINGW64:/c/Users/PC/desktop/Osproject/Osproject

```
PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/Osproject (main)
$ git clone https://github.com/chaitanyaPc/Osproject.git
Cloning into 'Osproject'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), 1.25 MiB | 1.21 MiB/s, done.

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/Osproject (main)
$ cd Osproject

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/Osproject/Osproject (main)
$ git init
Reinitialized existing Git repository in C:/Users/PC/Desktop/Osproject/Osproject/.git/

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/Osproject/Osproject (main)
$ git remote --verbose
origin https://github.com/chaitanyaPc/Osproject.git (fetch)
origin https://github.com/chaitanyaPc/Osproject.git (push)

PC@DESKTOP-2Q9GAF3 MINGW64 ~/desktop/Osproject/Osproject (main)
$
```