EXPERIMENT 9

AIM: Change specifications and make different versions using SCM tool

Project Name: College Library Management System

Team Members : Prem Chhabria(07), Prasad Govekar(19**),** Abhay Gupta(20)

THEORY :

Version Control :

➢ Version control combines procedures and tools to manage different versions of configuration objects that are created during the software process.

➢ Clemm [CLE89] describes version control in the context of SCM: Configuration management allows a user to specify alternative configurations of the software system through the selection of appropriate versions.

➢ This is supported by associating attributes with each software version, and then allowing a configuration to be specified [and constructed] by describing the set of desired attributes.'

Software Configuration Management (SCM):

➢ Also called software configuration management (SCM).

➢ It is an umbrella activity that is applied throughout the software process.

➢ Its goal is to maximize productivity by minimizing mistakes caused by confusion when coordinating software development.

➢ SCM identifies, organizes, and controls modifications to the software being built by a software development team.

Reason to Use Github :

Git is an incredibly fast tool. It is very efficient with large projects and it has an incredible branching system for non-linear development. Some of the goals of the new system were as follows:

➢ Speed

➢ Simple design

➢ Strong support for non-linear development (thousands of parallel branches)

➢ Fully distributed

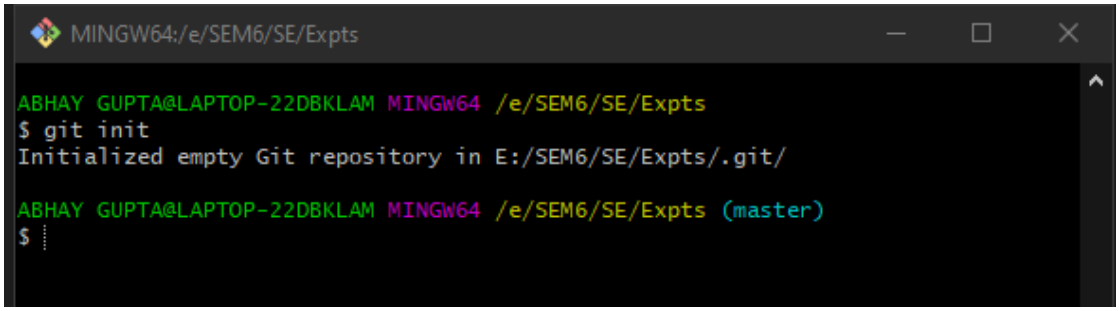
➢ Able to handle large projects like the Linux kernel efficiently (speed and data size)

COMMANDS :

Git init :

Usage: `git init [repository name]`

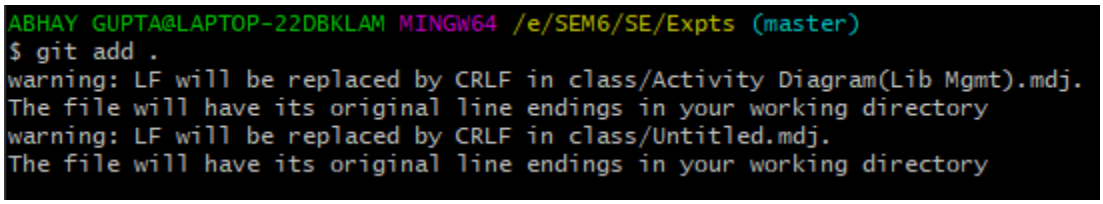
This command is used to start a new repository.



Git add :

Usage: `git add [file]`

This command adds a file to the staging area.



Git Log :

Usage: `git log`

This command is used to list the version history for the current branch.

Git commit:

Usage: `git commit -m "[ Type in the commit message]"`

This command records or snapshots the file permanently in the version history.

```
ABHAY GUPTA@LAPTOP-22DBKLAM MINGW64 /e/SEM6/SE/Expts (master)
$ git commit -m "performing expt 9"
[master (root-commit) ca5ab17] performing expt 9
 21 files changed, 11925 insertions(+)
 create mode 100644 Expt 1(Problem Statement).pdf
 create mode 100644 Expt 11(Component_Deployment).pdf
 create mode 100644 Expt 2(SRS).pdf
 create mode 100644 Expt 4(RMMM Plan).pdf
 create mode 100644 Expt 6(DFD_Data Dictionary).pdf
 create mode 100644 Expt 8(Sequence_Collaboration).pdf
 create mode 100644 Expt3(Project Schedule).pdf
 create mode 100644 Expt5(Use caseClass Dig).pdf
 create mode 100644 Expt7(Activity Diagram).pdf
 create mode 100644 Theory/SE_Expt11(Theory).pdf
 create mode 100644 Theory/SE_Expt8(Theory).pdf
 create mode 100644 class/Activity Diagram(Lib Mgmt).mdj
 create mode 100644 class/Main.png
 create mode 100644 class/Untitled.mdj
 create mode 100644 class/Untitled.pdf
 create mode 100644 class/class diagram.pdf
 create mode 100644 gantt/LibManagement.pdf
 create mode 100644 gantt/LibMgmt.pdf
 create mode 100644 gantt/Library Management.gan
 create mode 100644 gantt/Library Management.pdf
 create mode 100644 gantt/lib mgmt.gan
```

Git branch :

Usage: `git branch [branch name]`

This command creates a new branch.

```
ABHAY GUPTA@LAPTOP-22DBKLAM MINGW64 /e/SEM6/SE/Expts (master)
$ git branch -M main

ABHAY GUPTA@LAPTOP-22DBKLAM MINGW64 /e/SEM6/SE/Expts (main)
$
```

Usage: `git branch`

This command lists all the local branches in the current repository.

```
ABHAY GUPTA@LAPTOP-22DBKLAM MINGW64 /e/SEM6/SE/Expts (main)
$ git branch
* main
```

Git remote

Usage: `git remote add [variable name] [Remote Server Link]`

This command is used to connect your local repository to the remote server.

```
ABHAY GUPTA@LAPTOP-22DBKLAM MINGW64 /e/SEM6/SE/Expts (main)
$ git remote add origin https://github.com/abhay8463/EXPT-9-SE.git
```

Git push:

Usage: `git push [variable name] master`

This command sends the committed changes of master branch to your remote repository.

```
ABHAY GUPTA@LAPTOP-22DBKLAM MINGW64 /e/SEM6/SE/Expts (main)
$ git remote add origin https://github.com/abhay8463/EXPT-9-SE.git

ABHAY GUPTA@LAPTOP-22DBKLAM MINGW64 /e/SEM6/SE/Expts (main)
$ git push -u origin main
Enumerating objects: 26, done.
Counting objects: 100% (26/26), done.
Delta compression using up to 8 threads
Compressing objects: 100% (26/26), done.
Writing objects: 100% (26/26), 4.57 MiB | 899.00 KiB/s, done.
Total 26 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), done.
To https://github.com/abhay8463/EXPT-9-SE.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Git status :

Usage: `git status`

This command lists all the files that have to be committed.

```
ABHAY GUPTA@LAPTOP-22DBKLAM MINGW64 /e/SEM6/SE/Expts (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Expt 10(White Box Testing).pdf

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

CONCLUSION: Hence we have studied Software Configuration Management (SCM) and implemented the same for our project using Git Tool.