Version Control with hg

Developed by FOSSEE Team, IIT-Bombay.
Funded by National Mission on Education through ICT
MHRD,Govt. of India

Objectives

At the end of this session, you will be able to:

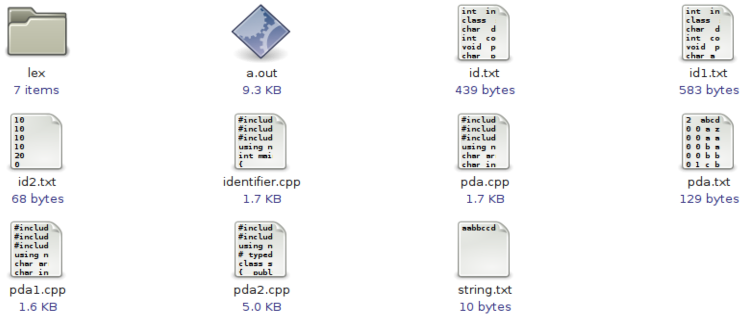
- Understand what is Version Control and the need for it.
- Create and use repository on a daily basis.

What is Version Control?

A way to track changes made to files over time, by keeping copies of files as we change them.

Home-brewed

An example of a home-brew Version Control system



```
$ ls
a.out  id1.txt  id2.txt  identifier.cpp  id.txt
lex  pda1.cpp  pda2.cpp  pda.cpp  pda.txt  string
```

Problems

- Name and changes made are not related or linked.
- Can’t track sequence of changes made to a file.
- Does not scale.

The need for Version Control

- To err is Human . . .
- Tracking the history and evolution of a project
- To collaborate effectively on a project
- To efficiently track down bugs and pin-point the changes that caused it

How does it work? — Analogy

It is, in some ways, similar to playing an Video game.

- We play games in stages
- Once we finish a stage or a task – we SAVE
- We continue playing
- But, if necessary, we could choose from one of the saved states and start from there
- We could alter the course of the game

Mercurial or hg

- Easy to learn and use
- Lightweight
- Scales excellently
- Written in Python

Installation

- sudo apt-get install mercurial
- TortoiseHg
- $ hg
- $ hg version

We need a repo!

- A Repository (repo) is where all the action is!
- Project files along with a special directory that stores all the changes
- We take snapshots of the whole repository; not individual files.

Initializing a repo

- $ hg init
- Creates a fresh repository
- Adds a .hg directory to our *working directory*

.hg directory keeps log of changes made henceforth

Status report

- hg status gives the status of our repo
- Use it often; at least as a beginner
- hg help command gives us help about command

Status codes

```
M = modified
A = added
R = removed
C = clean
! = missing
? = not tracked
I = ignored
```

Adding files

- From hg status we know, none of the files are being tracked, yet.
- hg add — asking hg to track these files
- As expected hg status prepends an A to the file names.
- ? –> A
- ! –> R (hg remove)

Taking Snapshots

- hg commit
- Asking Mercurial to take a snapshot; remember the changes made to the repository.
- -u FirstName LastName <email>
- -m “Commit message” – a description of changes committed.

Thumbnail views

- hg log gives the log of the changes made
- A changeset is an atomic collection of changes to the files (between successive commits)

Log information

- changeset: Identifier for the changeset
- user: Details of user who created the changeset
- date: Date and time of creation
- summary: One line description

User information

- User information is set in the hgrc file
- It can be set globally or local to the project
- Global hgrc
    - $HOME/.hgrc – Unix like systems
    - %HOME%
      .hgrc – Windows

Advice: commits, messages

- Atomic changes; one change with one commit
- Single line summary — 60 to 65 characters long
- Followed by paragraphs of detailed description
  - Why the change?
  - What does it effect?
  - Known bugs/issues?
  - etc.

Summary

In this tutorial, we have learnt to,

-
-
-
-

Evaluation

1.
2.
3.

Solutions

1.
2.

THANK YOU!

For more Information, visit our website

`http://fossee.in/`