Template of write-up for LA4

Name: Aniket Laxmikant Vaishnav

2017BTEIT00062

Group of LA123::- B

Use of Version Control System

Title: To use and experiment the online and offline Version Control System for Project Work.

Introduction: Version control, also known as source control, is the practice of tracking and managing changes to software code. Version control systems are software tools that help software teams manage changes to source code over time. As development environments have accelerated, version control systems help software teams work faster and smarter. They are especially useful for DevOps teams since they help them to reduce development time and increase successful deployments.

Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.

Software developers working in teams are continually writing new source code and changing existing source code. The code for a project, app or software component is typically organized in a folder structure or "file tree". One developer on the team may be working on a new feature while another developer fixes an unrelated bug by changing code, each developer may make their changes in several parts of the file tree.

Version control helps teams solve these kinds of problems, tracking every individual change by each contributor and helping prevent concurrent work from conflicting. Changes made in one part of the software can be incompatible with those made by another developer working at the same time. This problem should be discovered and solved in an orderly manner without blocking the work of the rest of the team. Further, in all software development, any change can introduce new bugs on its own and new software can't be trusted until it's tested. So testing and development proceed together until a new version is ready.

Steps of works (step by step)

Feature of work performing(List)

1. Create A GitHub Account
2. Create a new repository (GitHub GUI)
3. Create a file

3.1. In terminal type
    3.1.1. mkdir Demo
    3.1.2. cd Demo
    3.1.3. echo “Sample work” > Main.txt
    3.1.4. cat Main.txt
4. Initialise
    4.1. git init
    4.2. git add Main.txt
5. Make a commit
    5.1. git commit -m “message for commit” .
6. Connect GitHub Repository with Computer
    6.1. git remote add origin https://github.com/<your_username>/Demo.git
        i. Default desktop GUI :kde
        ii. Main purpose of that : user friendly os development
        iii. Package management of that distros : rpm
        iv. List of Default Packages : NA
        v. Important Screenshots of that distros:
        vi. Compare '/etc' hierarchy:
        vii. Compare package managers: [2]
        viii. Pros/cons of both distros:

Eg.

1 Mageia

1.1 Information:

Mageia is a fork of Mandriva Linux formed in September 2010 by former employees and contributors to the popular French Linux distribution. Unlike Mandriva, which is a commercial entity, the Mageia project is a community project and a non-profit organisation whose goal is to develop a free Linux-based operating system.

1.2 Screenshot:

Note : draw serperate Flochart and Data Variable/Function Dictionary for client server programs/multipls programs.

Eg.

Application Name	Version
Firefox	18.0.2
bash	4.2
chromium	23.0
gcc	4.7.2
gimp	4.8.2
gnome-shell	3.6.2
grub	0.97
linux	3.8.0

Table 1: Default Application in Mageia

Eg.

Figure 2: Linux Mint

(Note: add imp figures/image and tables with title to figure and table, if table, figure /image copied, mention reference [xx])

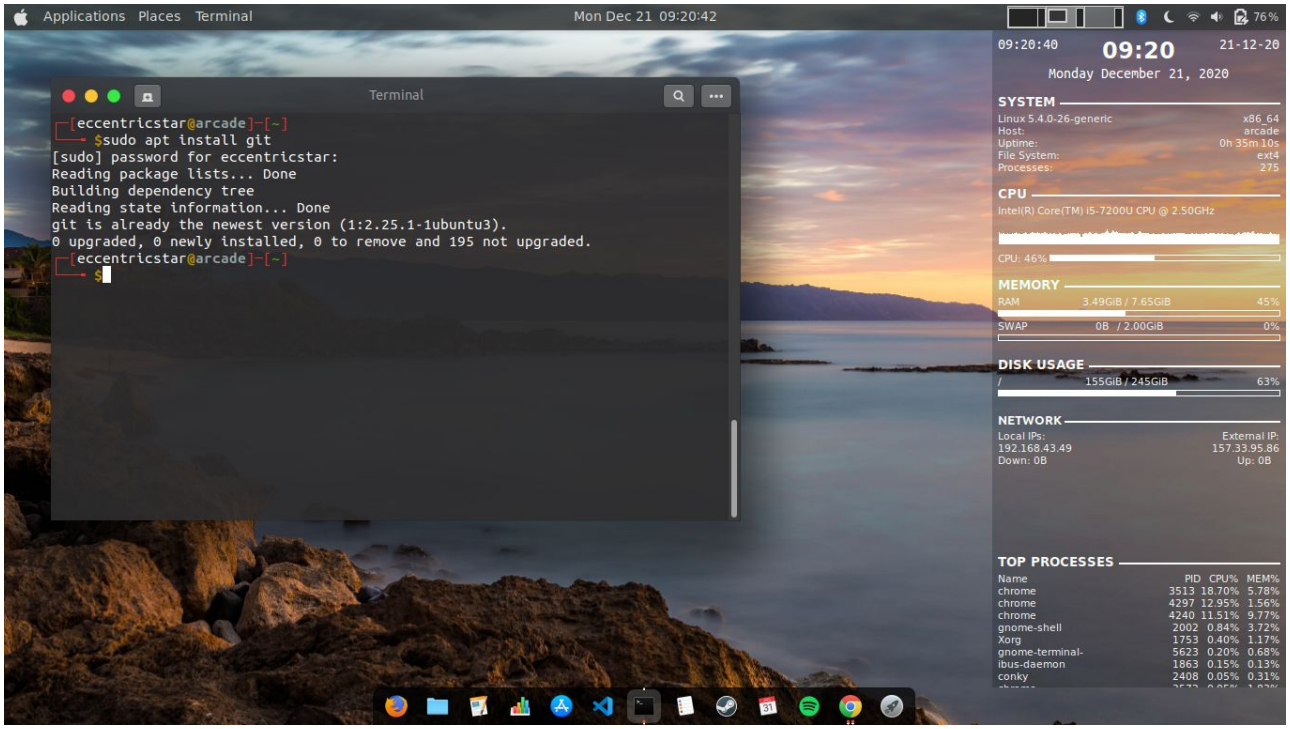
Information of tool with Characteristic /feature/applications of tool etc:

1. Strong support for non-linear development
2. Distributed development
3. Compatibility with existent systems and protocols
4. Efficient handling of large projects
5. Cryptographic authentication of history
6. Toolkit-based design
7. Pluggable merge strategies
8. Garbage accumulates until collected
9. Periodic explicit object packing

Steps of configuration if any:

1. **Install Git client on your local machine.**
2. **Debian users : sudo apt install git**
3. **Fedora users : sudo dnf install git**

Flowcharts or steps of configuration:



Conclusions: (12 normal)

1. Learn to use Git as version control system.
2. Learn the maintaining connectivity between Git and GitHub by creating repository.
3. Learn important commands necessary to carry out this activity.

References:

[1] https://en.wikipedia.org/wiki/Git
[2] https://git-scm.com/doc
[3] https://docs.github.com/en/free-pro-team@latest/github/getting-started-with-github