A

REPORT

OF

**“Open Source Software laboratory”**

**Code: 3IT475**

**BIE: Expert (E)**

Submitted by

**MR. ANIKET M. LAHAMGE (2016BTEIT00040)**

**Email:** aniketlahamge23@gmail.com

**Mobile:** 7776976391



DEPARTMENT OF INFORMATION TECHNOLOGY

**WALCHAND COLLEGE OF ENGINEERING, SANGLI**

**(An Autonomous Institute)**

**2019-2020**

©2019-20, Ms. Aniket L. and Dr. A. J. Umbarkar.

**CERTIFICATE**



This is to certify that the report entitled “*Open Source Software laboratory (OSS Lab) 3IT475*” submitted by *MR. ANIKET M. LAHAMGE.(2016BTEIT00040)* is a record of student's own work carried out by him during the academic year 2019-2020, as per the curriculum/syllabus laid down for OSS Lab at Final Year B.Tech IT Sem-I. She has carried out experiments/FOSS tools hands-on on *15* assignments under BIE: Expert (E) category successfully.

Dr. A. J. Umbarkar

(Course Teacher)

**Declaration**

I, the undersigned, hereby declare that the B.TECH Report entitled **“***Open Source Software laboratory (OSS Lab) (3IT475)* **“submitted** by me to OSS Lab report at Final Year B.Tech IT Sem-I, is my original/experimented/experience work. I further declare that, to the best of my knowledge and belief, this report has not been previously submitted or copied by me.

I declare that this report reflects my thoughts about the subject in my own words. I have sufficiently cited and referenced the original sources, referred or considered in this work. I have not misrepresented, fabricated, or falsified any idea/data/fact/source in this my submission. I understand that any violation of the above will be cause for disciplinary action by the Course Teacher/Institute.

(Sign)

Date: **Mr. Aniket Lahamge**

Place:

**Acknowledgement**

We feel immense pleasure in submitting the report entitled “*Open Source Software laboratory (OSS Lab) (3IT475)***"**. We are thankful to our guide **Dr. A. J. Umbarkar** for their valuable guidance and kind help during implementing the OSS Lab.

Acknowledged By,

**Mr. Aniket Lahamge**

Sign

**IT 475: Open Source Software Lab**

**(E: Expert) Final Assignment List**

1. **Demonstration****of Linux Distributions OS’s and their purpose with comparisons.**

**(Fedora/CentOS/any other/etc.: Any One)**

**(Submission by Individual [I])**

***Objective:*** *To install and demonstrate Various Linux Distributions and their Purpose/comparison/differences.*

***Outcome:*** *Self learning/lifelong learning (PO: b, k, l)*

Student asks to study at least two Linux Distros, with their comparisons and installation on Virtual Box OR Installation Linux on Live USB pen drive. [https://fedoraproject.org/wiki/How\_to\_create\_and\_use\_Live\_USB]

In Journal, they have to write information of that distros, such as:-

1. Various versions of that distros with code name
2. Default desktop GUI
3. Main purpose of that
4. Package management of that distros
5. List of Default Packages
6. Screenshots of that distros
7. Compare '/etc' hierarchy
8. Compare package managers
9. Pros/cons of both distros
10. Which one is better for development and why?
11. Which one is easy to use (for beginner) and why?
12. Explorer any top 10 commands of that distro on command prompt.
13. Make the Official Repositories of **Fedora/CentOS** on docker store (https://hub.docker.com/) and experiment for above.

**Reference:-**

1. List of Linux Distros:- [http://distrowatch.com](http://distrowatch.com/)/
2. For installation on Virtual Box:- https://help.ubuntu.com/community/ListOfOpenSourcePrograms
3. http://www.psychocats.net/ubuntu/virtualbox
4. https://help.ubuntu.com/
5. **Use of Open Source tools for Project Management.**

* **(Sonar,** [**Targetprocess**](https://blog.capterra.com/free-open-source-project-management-software/#11)**,** [**Redbooth**](https://blog.capterra.com/free-open-source-project-management-software/#10)**,** [**Pivotal Tracker**](https://blog.capterra.com/free-open-source-project-management-software/#9)**, OrangeScrum etc.: Any One)**

**(Submission by Individual [I])**

***Objective:*** *To install and demonstrate the use of various open source software’s that used in day to day life of software Engg.*

***Outcome:*** *(PO: k, l)*

Students have to experiment at last two project management tools / software’s and they have to use for their project/FOSS project/mini project.

In journal, they have to write information about that tool such as:-

1. Purpose behind that tool.
2. Various versions of those tools.
3. Installation and Configuration of that tool.
4. How to use that tool.

Reference:-

1. <https://bitnami.com/stacks>
2. Sphinx for all lab/college documentation by the students.
3. http://www.sonarqube.org/
4. Wikipedia List Of Software’s:- [http://en.wikipedia.org/wiki/List\_of\_free\_and\_open](http://en.wikipedia.org/wiki/List_of_free_and_open-source_software_packages)- [source\_software\_package](http://en.wikipedia.org/wiki/List_of_free_and_open-source_software_packages)s
5. Open Source Software Mega List :-

<http://www.datamation.com/open-source/open->  [source-software-the-mega-list.htm](http://www.datamation.com/open-source/open-source-software-the-mega-list-1.html)l

1. <https://fedoraproject.org/wiki/Education_Spin> This has lots of relevant packages
2. <http://www.methodsandtools.com/tools/targetprocess.php>
3. <https://blog.capterra.com/free-open-source-project-management-software/>
4. <http://www.targetprocess.com/>
5. <https://www.pivotaltracker.com/features/>
6. https://redbooth.com/features
7. **Use of Bug Tracking**

## (Phabricator, Youtrack, Mantis, [Futuramo](https://www.capterra.com/p/158076/Futuramo-Visual-Tickets/), etc.: Any One )

**(Submission by Individual [I])**

***Objective:*** *To install and demonstrate the use of various open source software’s that used in day to day life of software engineering.*

***Outcome:*** *(PO: k, l)*

Students have to experiment at last two bug tracking tools / software’s and they have to use for their project/FOSS project/mini project.

In journal, they have to write information about that tool such as:-

1. Purpose behind that tool.
2. Various versions of that tool.
3. Installation and Configuration of that tool
4. How to use that tool.
5. Make the Official Repositories of any one above **Bug Tracking** on docker store (https://hub.docker.com/) and experiment.

Reference:-

1. <https://bitnami.com/stacks> (https://bitnami.com/stack/trac)
2. Wikipedia List Of Software’s:-[http://en.wikipedia.org/wiki/List\_of\_free\_and\_open](http://en.wikipedia.org/wiki/List_of_free_and_open-source_software_packages)- [source\_software\_package](http://en.wikipedia.org/wiki/List_of_free_and_open-source_software_packages)s
3. Open Source Software Mega List :- [http://www.datamation.com/open-source/open](http://www.datamation.com/open-source/open-source-software-the-mega-list-1.html)-

[source-software-the-mega-list-1.htm](http://www.datamation.com/open-source/open-source-software-the-mega-list-1.html)l

1. **Use of Version Control System.**

**(**[**Mercurial**](https://www.mercurial-scm.org/) **(hg),** [**Bazaar**](http://bazaar-vcs.org/)**, Monotone, etc: Any One).**

**(Submission by Individual and Group [I and G])**

***Objective****: To use the online and offline Version Control System in Open Source/for their project work.*

***Outcome****: lifelong learning (PO: b, c, k, l)*

Students have to experiment any two **Version Control System** and use the tool for their project/FOSS project/mini project/ etc.

Sample code developments example of **Version Control System** on both Windows and Linux clients/server.

Make the Official Repositories of any one **Version Control System** on docker store (https://hub.docker.com/) and experiment.

In Journal, They have to write Basic Information about **Version Control System**, commands, their working, diagrams, differences, pros and cons, developments history, etc .

Reference:-

1. https://try.github.io/levels/1/challenges/1
2. https://github.com/princeton-8/princeton-8.github.io
3. http://wiki.openhatch.org/Open\_Source\_Comes\_to\_Campus/Practicing\_Git/Students
4. GIT Official Documentation:- [http://git-scm.com/documentatio](http://git-scm.com/documentation)n
5. SVN Official Documentation:- <http://svnbook.red-bean.com/en/1.7/index.html>
6. Perforce Helix is a commercial, proprietary revision control system developed by Perforce Software
7. <https://www.smashingmagazine.com/2008/09/the-top-7-open-source-version-control-systems/>
8. http://wiki.bazaar.canonical.com/WindowsDownloads
9. **Installation and Use of CMS software’s.**

**(Joomla,** [**Mahara**](https://bitnami.com/stack/mahara)**, XOOPS,** [**DokuWiki**](https://bitnami.com/stack/dokuwiki)**, etc.: Any One) OR (Social Networking open source:** [**Diaspora**](https://bitnami.com/stack/diaspora) **or other: Any One)**

**(Submission by Individual [I])**

***Objective:*** *To comprehend the use of Content Management System and their Use for personal website/dept CMS.*

***Outcome:*** *Self learning (PO: b, I, j, k, l)*

Students have to study at least experiment one **CMS** and one **Wiki**. Use **Wiki** for giving the information to class student to perform FOSS assignments. Use **CMS** for giving the creating your personal website/blog or FOSS course website/blog.

In Journal, They have to write,

* 1. Administration of CMS/wiki.
  2. How to Use.
  3. Installation on Linux Platform.
  4. Screenshots.
  5. Make the Official Repositories of any one **CMS/Wiki** on docker store (https://hub.docker.com/) and experiment.

References:-

1. www.wordpress.com
2. Drupal Tutorials:- [http://drupal.org/documentation/customization/tutorial](http://drupal.org/documentation/customization/tutorials)s
3. Moodle Tutorials:- http://docs.moodle.org/22/en/Moodle\_video\_tutorials
4. https://bitnami.com
5. **Comprehend the Open Source Software Development for any one Linux distro. (Topic 3rd in Syllabus) (Fedora/CentOS, etc :Any One)**

**(Submission by Individual or Group [I or G])**

***Objective:*** *To comprehend the open source software development.*

***Outcome:*** *Self learning/lifelong learning (PO: b, k, l)*

In this student have to study open source software development process of **any one above Linux distro.**

Get the details following information like –Name of community, website, Mailing List, wiki, version control, bug tracking and documentation of the particular distro to comprehend.

Sample of Ubuntu Development:-

* 1. Development Communities:- [http://www.ubuntu.com/community](http://www.ubuntu.com/community/)/
  2. Mailing List:- [https://lists.ubuntu.com](https://lists.ubuntu.com/)/
  3. IRC channels:- [https://wiki.ubuntu.com/IRC/ChannelLis](https://wiki.ubuntu.com/IRC/ChannelList)t
  4. Ubuntu Wiki:- [https://wiki.ubuntu.com](https://wiki.ubuntu.com/IRC/ChannelList)/
  5. Ubuntu Version Control:- [https://code.launchpad.net/ubunt](https://code.launchpad.net/ubuntu)u
  6. Ubuntu Bug Tracking:- [https://bugs.launchpad.net/ubunt](https://bugs.launchpad.net/ubuntu)u
  7. Ubuntu Localization :- <https://translations.launchpad.net/ubuntu>
  8. Ubuntu Documentation:- [https://help.ubuntu.com/communit](https://help.ubuntu.com/community)y

Sample of Fedora

I. <https://getfedora.org/>

Ii. <https://fedoraproject.org/wiki/IRC>

Iii. <https://fedoraproject.org/wiki/>

Iv.<https://fedoraproject.org/wiki/Packaging:Versioning> / <https://fedoraproject.org/wiki/Infrastructure/VersionControl>

V. <https://fedoraproject.org/wiki/Bugzilla>

Vi. <https://fedoraproject.org/wiki/Category:Localization>

Vii. <https://docs.fedoraproject.org/en-US/index.html>

Sample of Mint

1. mint Version Control http://community.linuxmint.com/software/view/subversion
2. Mint Bug Tracking:- http://forums.linuxmint.com/viewtopic.php?f=60&t=157099
3. Mint Localization :-http://www.linuxmint.com/communities.php
4. Mint Documentation:-http://www.linuxmint.com/documentation.php
5. **Compilation of Linux Kernel selected above.**

**(Fedora/CentOS, etc: Any One)**

**(Submission by Individual or Group [I or G])**

***Objective:*** *To demonstrate how to compile Linux Kernel.*

***Outcome:*** *Self learning (PO: k, l)*

In this student have to do the compilation any one mentioned above Linux distro Linux Kernel on their system/VMware/Virtual box or pen drive or docker container.

Reference:-

1. Installation Of Linux Kernel on Fedora:- [http://www.howopensource.com/2011/08/how-to-install-compile-linux-kernel-3-0](http://www.howopensource.com/2011/08/how-to-install-compile-linux-kernel-3-0-in-fedora-15-and-14/)- [in-fedora-15-and-14](http://www.howopensource.com/2011/08/how-to-install-compile-linux-kernel-3-0-in-fedora-15-and-14/)/

Or/and

1. https://fedoraproject.org/wiki/Building\_a\_custom\_kernel
2. Latest kernel installation on Fedora and Cent OS:- [http://www.tecmint.com/kernel-3-5](http://www.tecmint.com/kernel-3-5-released-install-compile-in-redhat-centos-and-fedora/)- [released-install-compile-in-redhat-centos-and-fedora](http://www.tecmint.com/kernel-3-5-released-install-compile-in-redhat-centos-and-fedora/)/
3. http://tldp.org/guides.html
4. Installation Of Linux Kernel on Ubuntu:-

Latest kernel installation on :- <http://www.backtracklinux.org/forums/showthread.php?t=49347>

Installation of Linux Kernel on Suse:-https://en.opensuse.org/Kernel

In Journal you have to write the step by step process of compilation.

1. **Create of RPM or DEB packages (Any One)**

**(Submission by Individual [I])**

***Objective:*** *To Create package for any above Linux distros.*

***Outcome:*** *(PO: b, I, j, k, l)*

Students have to study RPM or DEB package building for their C, C++ or JAVA Codes(any one programming languages codes). They must build an rpm or debian package and install it through package manager such as YUM or APT-GET

Reference:-

1. Build Simple rpm package:- [http://rhce.dposs.org/index.php](http://rhce.dposs.org/index.php?title=Build_a_simple_RPM_that_packages_a_single_file)? [title=Build\_a\_simple\_RPM\_that\_packages\_a\_single\_fil](http://rhce.dposs.org/index.php?title=Build_a_simple_RPM_that_packages_a_single_file)e
2. Fedora rpm doc:- [http://fedoraproject.org/wiki/How\_to\_create\_an\_RPM\_packag](http://fedoraproject.org/wiki/How_to_create_an_RPM_package)e
3. Simple DEB package for your C code:- [http://linuxconfig.org/easy-way-to-create-a](http://linuxconfig.org/easy-way-to-create-a-debian-package-and-local-package-repository)- [debian-package-and-local-package-repositor](http://linuxconfig.org/easy-way-to-create-a-debian-package-and-local-package-repository)y
4. Simple DEB build guide:- [http://askubuntu.com/questions/90764/how-do-i-create-a](http://askubuntu.com/questions/90764/how-do-i-create-a-deb-package-for-a-single-python-script)- [deb-package-for-a-single-python-scrip](http://askubuntu.com/questions/90764/how-do-i-create-a-deb-package-for-a-single-python-script)t
5. Deb Package Build YouTube:- [http://www.youtube.com/watch?v=nhoRyd2CEV](http://www.youtube.com/watch?v=nhoRyd2CEVs)s

In Journal you have to write the package building process.

1. **Install and demonstrate****of Server based services and their Uses.**

**(**web **server apache or tomcat or IIS, NFS,NIS**: **Any One)**

**(Submission by Individual [I])**

***Objective:*** *To know server installations and Configurations on Linux Platform*

***Outcome:*** *(PO: b, I, j, k, l)*

Students are asked to install and configure at least 2 servers, such as FTP, HTTP server (web server), TELNET, NFS, NIS etc. All configurations must be done on Linux Platform

In Journal, they have to write installations, Configurations and Screenshots of server on which they worked.

Make the Official Repositories of any one above **server** on docker store (https://hub.docker.com/) and experiment.

References:-

* 1. Server World:- [http://www.server-world.info/en](http://www.server-world.info/en/)/
  2. Yolinux :- [http://www.yolinux.com](http://www.yolinux.com/)/
  3. GUI based tool for server configuration Webmin <http://www.webmin.com/index.html>

1. **Development of new Open Source Software or contribution to existing Open Source Software.**

**(Any small application other than Music Player or Calculator or Text Editor in java/python/perl/c/cpp/etc: Any One or New open source development).**

**(Submission by Individual or Group [I or G])**

***Objective:*** *To contribute/introduce the open source software by understanding the GPL Licensing.*

***Outcome:*** *Self learning/lifelong learning (PO: b, I, j, k, l)*

1. Develop simple software for basic needs such as Calculator, editor etc.

Use following:-

1. Language:- C/C++, Python, Perl, PHP, Java, .net
2. Version Control :- GIT or SVN
3. Package Building:- debian or rpm
4. Translation:- Marathi or Hindi
5. Documentation:- Use Mallard for your Help
6. Make the Official login on online repositories of open source projects with valid login by individual and ask group too. Take any above suitable open project from online open source project and add feature/option/GUI/remove error/modules/etc.
7. Appreciation mail may be received from main developer/introducer if your contribution is quality and remarkable.

**References:-**

1. http://teachingopensource.org/start-contributing-using-open-source-software/
2. https://www.fossology.org/get-started
3. http://foss2serve.org/index.php/Category:Projects
4. http://www.hfoss.org/index.php/project\_gallery
5. GIT version control Tutorial:- [http://git-scm.com/documentatio](http://git-scm.com/documentation)n
6. SVN :- [http://michael-zamir.blogspot.in/2012/01/svn-tutorial.htm](http://michael-zamir.blogspot.in/2012/01/svn-tutorial.html)l
7. Translation :-<http://www.tuxamito.com/joomla/index.php/es/component/content/article/60-gettext>- [tutoria](http://www.tuxamito.com/joomla/index.php/es/component/content/article/60-gettext-tutorial)l
8. Using Malarad:- <http://projectmallard.org/about/learn/tenminutes.html>
9. http://www.hfoss.org/index.php/project\_gallery
10. http://www.shlomifish.org/philosophy/computers/open-source/how-to-start-contributing/tos-document.html

In Journal you have to write the process in Brief.

1. **Docker container : An open source software development platform (any two)**

**(Submission by Individual or Group [I or G])**

***Objective:*** *To understand and use the docker virtualization as OSS.*

***Outcome:*** *Self learning/lifelong learning (PO: b, I, j, k, l)*

1. With the help of Docker/Container show the any one above Linux distros selected. (in assignment 1.)
2. 1. Create image/container of any FOSS tool and upload on Docker Hub.

2. Pull images/containers from docker-hub: <https://hub.docker.com/>

1. (FOSS tool bug tracking tool, Project management tool, Version control system, CMS, python, java language running/compilation support, etc. and follow respective tool assignment)
2. Contribute/Introduce the docker/container to make the resource management easy and lighter.
3. Create IPC between two OS container. \*\*\* for TY UOS\*\*\*
4. With the help of Docker-compose deploy the ‘Wordpress’ and ‘Mysql’ container and access the front end of ‘Wordpress’
5. Docker image:

A. Create a simple Hello-world python flask application and create the docker image of that Flask application.

B. Run the docker container from recently created image and run that docker container to 5000 port of host system.

1. Create the ‘nginx’ container from ‘nginx’ image. And create the load balancing so that if we go to tha address of ‘nginx’ it can redirect it to the above created applications (Flask and Wordpress).

**Note.** Docker has to be installed first, to carry out docker based experiments. Prefer the Linux OS to do this assignment.

**References:-**

1. https://www.docker.com/
2. https://opensource.com/resources/what-docker
3. https://mobyproject.org/
4. https://labs.play-with-docker.com/

**Extra Resources docker/container learning:**

1. play with docker

<http://labs.play-with-docker.com/>

1. docker curriculum on github

<https://docker-curriculum.com/>

1. awesome-docker on github

<https://github.com/veggiemonk/awesome-docker>

1. docker cheatsheet

<https://www.docker.com/sites/default/files/Docker_CheatSheet_08.09.2016_0.pdf>

<https://github.com/wsargent/docker-cheat-sheet>

1. basics of docker pdf used in workshop

<ftp://10.10.13.13/Basics%20of%20Docker.pdf>

1. **Find python kernel code and compile it or use any python library for any application.**

**(Submission by Individual or Group [I])**

***Objective:*** *To use the python open source ready module for application development.*

***Outcome:*** *Self learning/lifelong learning (PO: b, I, j, k, l)*

Example:

1. A Python library to write a table in various formats: CSV / Elasticsearch / HTML / JavaScript / JSON / Jupyter Notebo…

2. Python Driver for ArangoDB, a NoSQL graph database

3. A fast image processing library with low memory needs.

4. Any work on Concurrency and Parallelism

Libraries for concurrent and parallel execution.

* [eventlet](http://eventlet.net/) - Asynchronous framework with WSGI support.
* [gevent](http://www.gevent.org/) - A coroutine-based Python networking library that uses [greenlet](https://github.com/python-greenlet/greenlet).
* [multiprocessing](https://docs.python.org/2/library/multiprocessing.html) - (Python standard library) Process-based "threading" interface.
* [threading](https://docs.python.org/2/library/threading.html) - (Python standard library) Higher-level threading interface.
* [Tomorrow](https://github.com/madisonmay/Tomorrow) - Magic decorator syntax for asynchronous code.
* [uvloop](https://github.com/MagicStack/uvloop) - Ultra fast implementation of asyncio event loop on top of libuv.

**Profiler**

* [line\_profiler](https://github.com/rkern/line_profiler) - Line-by-line profiling.
* [memory\_profiler](https://github.com/fabianp/memory_profiler) - Monitor Memory usage of Python code.
* [profiling](https://github.com/what-studio/profiling) - An interactive Python profiler.
* [vprof](https://github.com/nvdv/vprof) - Visual Python profiler.

Game Development : Awesome game development libraries.

* [Cocos2d](http://cocos2d.org/) - cocos2d is a framework for building 2D games, demos, and other graphical/interactive applications. It is based on pyglet.
* [Panda3D](https://www.panda3d.org/) - 3D game engine developed by Disney and maintained by Carnegie Mellon's Entertainment Technology Center. Written in C++, completely wrapped in Python.
* [Pygame](http://www.pygame.org/news.html) - Pygame is a set of Python modules designed for writing games.
* [PyOgre](http://www.ogre3d.org/tikiwiki/PyOgre) - Python bindings for the Ogre 3D render engine, can be used for games, simulations, anything 3D.
* [PyOpenGL](http://pyopengl.sourceforge.net/) - Python ctypes bindings for OpenGL and it's related APIs.
* [PySDL2](http://pysdl2.readthedocs.io/en/rel_0_9_5/) - A ctypes based wrapper for the SDL2 library.
* [RenPy](https://www.renpy.org/) - A Visual Novel engine.

Ref: <https://github.com/vinta/awesome-python>

1. [**Agile s/w engineering by using Tuleap, review board and gitea (tools of agile setup)**](https://app.schoology.com/assignment/2153710625)

https://www.tuleap.org/1. Read and register for online use. (B and I)

* Offline use, setup Tuleap, review board and gitea setup on docker container?(E)
* Share you tool details setup, usage, videos link etc in assignment write-up. Take small final year project in this environment as case study for demonstration.
* This assignment is introduced in year 19-20 first time.

1. **Learn** [**Open source programming language GO (Compulsory for E)**](https://app.schoology.com/assignment/2223936562)

* Write 3 suitable programs using GO language.
* Compare GO language with functional and procedural languages.
* Enlist the features of GO language
* Commands and compiler, debuggers of GO.
* Applications of GO language
* Put programs (with statements) in write-up with compilation steps details and upload.

1. **Bonus Assignments.** 
   1. [**Create a poster by individual for a foss tool, using dia online drawing tool. (for 2 additional bonus mark in ise)**](https://app.schoology.com/assignment/2261684092) **Use Flowchart Maker & Online Diagram Software: https://www.draw.io/**

Uses of tool, Describe working diagrams/steps, information of commands, etc. in poster. Give links of good ref site/videos.

Exported SVG file format and Upload this entire folder one zipped file on schoology.

* 1. [**Emac editor experience for coding, documentation, indentation, foss std/Google std coding in languages (1 mark bonus)**](https://app.schoology.com/assignment/2261692631)

Experiment Emac editor on Linux/ windows for coding, documentation, indentation, foss std/Google std coding in languages like c,cpp, python, java, latex, etc.

In word file upload

1. Give detail/steps of emac editor installation and packages of coding, documentation, indentation, foss std/Google standard d coding.

2. Explain you experience of coding with Emac editor.

3. Compare Emac with eclipse IDE.

4. Give important link/ reference /videos of Emac editor.

* 1. [**FOSS HacktoberFest Pull requests can be made in any GitHub-hosted repositories/projects (Optional to assignment 10)**](https://app.schoology.com/assignment/2273245752)

This is Optional assignment to  assignment 10. Do any one at least.

Visit for this site before start and reads first page carefully.

Every year  October is FOSS month.... hence this is FOSS contribution initiatives.

<https://hacktoberfest.digitalocean.com/>

**Steps**.

1. Create account on above link.

2. Make 4 valid Pull Requests on Github between 1st -31st October.

3. Post the screenshot of your Hacktoberfest profile and screenshot of Github Profile on Schoology. (In write-up of this assignment)

4. Give details of all 4 Pull Requests in write-up.

5. Also attach the screenshot of email in write-up.

* 1. [**(Extra ISE bonus )Sugar is an activity-focused, free/libre open-source software learning platform for children.**](https://app.schoology.com/assignment/2153717117)

https://sugarlabs.org/

Sugar is an activity-focused, free/libre open-source software learning platform for children. Collaboration, reflection, and discovery are integrated directly into the user interface. Through Sugar's clarity of design, children and teachers have the opportunity to use computers on their own terms. Students can reshape, reinvent, and reapply both software and content into powerful learning activities. Sugar's focus on sharing, criticism, and exploration is grounded in the culture of free software (FOSS)

* 1. **Create Video any one FOSS tool liked by you (bonus 1 mark )**

**Upload Video on you tube and give YouTube link here**

* 1. [**Discussion on all assignments (active involvement with technical content will bonus 1 mark in ise )**](https://app.schoology.com/course/2153686479/materials/discussion/view/2228634940) **Put the good url/website pdf link, how to do, video links here for other help on assignments. (active involvement with technical content will bonus 1 mark in ise )**

**Note:-**

1. All assignments first need to write in word soft form as per format of word (attached with assignment list). Finally, at the end of all assignments, merge all assignments in the form of journal using word Template format or latex template (is given on schoology). Final well formatted (English checked/plagiarism checked /reference/ page no/ etc) journal expected at the end of course for getting journal marks.
2. Submit all assignments to ***schoology.com*** on time as per instructions.
3. See ISE plan for marks allotments.

## Course Objective:

1. To Configure the Open Source Software.
2. To contribute/ develop software (system) for open source environment.
3. To use FOSS for Software Engineering.

## Percentage of Objective achieved by students:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objective No. | Not achieved | 40% achieved | 70% achieved | 100% achieved |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

**Course Learning Outcomes:**

1. Use of FOSS in software development
2. Understanding the economics of FOSS
3. Define free software, Open source FOSS Software.

## Percentage of Outcome achieved by students:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Outcomes | Not achieved | 40% achieved | 70% achieved | 100% achieved |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

**Mr. Aniket Lahamge**

(Sign)

**Program Outcomes:**

1. **Engineering Knowledge:** Apply the knowledge of mathematics, engineering run and computational science to the solution of engineering problems.
2. **Problem Analysis:** Identify, formulate. interpret and analyze the complex engineering problems leading to substantiated conclusions
3. **Design/Development of Solutions**: Design systems, components or processes to meet desired needs within realistic constraints such as economic, environmental, societal and ethical considerations.
4. **Conduct investigations of Complex Problems**: Use research based knowledge and methods including design of experiments, analysis, interpretation and synthesis of information to provide valid conclusions.
5. **Modern Tool Usage**: Select and apply appropriate techniques, engineering skills and modem IT tools to prototype the model of complex engineering activities.
6. **The Engineer and Society:** Apply contextual knowledge pertaining social, secure. Legal and cultural issues with consequent responsibilities relevant to IT.
7. **Environment and Sustainability:** Understand the impact of the professional engineering solutions in social, environmental and the global contexts, demonstrating the knowledge of and the need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to the professional ethics with responsibilities and norms of the engineering practices.
9. **Individual and Team Work:** Work effectively as an individual as well as a member or a leader in diverse teams for multidisciplinary settings.
10. **Communication:** Communicate effectively with the engineering community and with society at a large, such as, being able to comprehend and write reports and design documentation to make effective presentations.
11. **Project Management and Finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply those to original work or contemporary issues, as a member or land a leader in a team or an entrepreneur to manage projects in multidisciplinary environments.
12. **Life-long Learning**: Recognize the need and prepare to engage independent and in lifelong learning.

## PO Mapping with Tutorial List

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Assignment No *\*PO | (a) | (b) | (c) | (d) | (e) | (f) | (g) | (h) | (i) | (j) | (k) | (l) |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |  |  |  |  |  |

**Mr. Aniket Lahamge**

(Sign)

## Course Learning outcomes:

|  |  |  |  |
| --- | --- | --- | --- |
| CO | After the completion of the course the student should be able to | Bloom’s Cognitive | |
| Level | Descriptor |
| CO1 | Exercise the FOSS tools in software development. | 3 | Applying |
| CO2 | Analyze the economics of FOSS. | 4 | Analyzing |
| CO3 | Create new FOSS or contribute to existing FOSS in FOSS environment. | 6 | Creating |

## CLO mapping with assignment list

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Assignment | BIE | % Copy | Mapping | category |
| 1 | Demonstration of Linux Distributions OS’s and  their purpose with comparisons. | E | 5% | CO1 | Application |
| 2 | Use of Open Source tools for Project Management. | E | 1% | CO2 | Application |
| 3 | Use of Bug Tracking | E | 1% | CO1 | Knowledge |
| 4 | Use of version control system | E | 0% | CO2 | Application |
| 5 | Installation and Use of CMS software | E | 5% | CO1 | Application |
| 6 | Comprehend the open source software development for any one Linux Distro | E | 1% | CO3 | Synthesis  Design |
| 7 | Compilation of Linux kernel | E | 1% | CO3 | Synthesis |
| 8 | Creation of RPM or DEB packages | E | 0% | CO2 | Application |
| 9 | Install and demonstrate server based services and their uses | E | 5% | CO2 | Application |
| 10 | Development of new Open Source Software or contribution to existing Open Source Software. | E | 1% | CO2 | Application |
| 11 | Docker container: An open source software development platform | E | 1% | CO2 | Application |
| 12 | Find python kernel code and compile it or use any python library for any application. | E | 0% | CO3 | Application |
| 13 | 1. [Agile s/w engineering by using Tuleap, review board and gitea (tools of agile setup)](https://app.schoology.com/assignment/2153710625) | E | 5% | CO1 | Synthesis  Design |
| 14 | Learn Open source programming language GO | E | 1% | CO2 | Knowledge |
| 15 | Bonus Assignments. | E | 1% | CO2 | Creating |

## Rubrics Used:

1. Quiz Objective
2. Class Questioning.
3. Quiz Subjective
4. Open Book Test
5. Assignment
6. Program
7. Seminar
8. Mini project
9. PPT
10. Demo Simulator
11. ISE1/ISE2/ESE

## Rubrics mapping with assignment list:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Assignment No./Rubric No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |  |  |  |  |

**Self Evaluation by Student:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name of student | Exam no. | Email ID | Roll no. | Sign |
| Aniket Lahamge | 2016BTEIT00040 | [aniketlahamge23@gmail.com](mailto:dikshayanpallewar@gmail.com) | 40 |  |

**Assignment 1**

# Demonstration of Various Linux Distributions and Their Purpose with Comparisons(Fedora)

Installation of various linux distributions.

## Objective : -

To install and demonstrate Fedora Linux Distributions and their Purpose/comparison/differences.

## Outcome : -

Self learning/lifelong learning (PO: b, k, l) .

## Instruction for the Assignment

Student have to study atleast two Linux Distribution, with their comparisons and installation.

## Various Distributions:

* + 1. **Fedora :-**

Fedora (formerly Fedora Core) is a Unix-like operating system based on the Linux kernel and GNU programs (a Linux distribution), de- veloped by the community- supported Fedora Project and sponsored by the Red Hat company. Fedora contains software distributed un- der various free and open-source licenses and aims to be on the leading edge of such technologies. Fedora is the upstream source of the commercial Red Hat Enterprise Linux distribution.

**Versions:-**

* + - 1. Fedora 27 was released on November 14, 2017.
      2. Fedora 26 was released on July 11, 2017.

**Default Desktop Environment:-**

GNOME is the default desktop environment provided for Fedora

* + 1. **Features:-**
* Support for multiple repositories.
* Simple configuration.
* Dependency calculation based on modern solving technology.
* Faster and less memory-intensive operation.
* RPM-consistent behavior
* Package group support, including multiple-repository groups.
* Simple interface
* Documented, solid Python API
* DNF runs in both Python 2 and Python 3

**Package Management:** The Dandified yum **DNF** is the next generation version of yum. It roughly maintains CLI compatibility with yum and defines a strict API for extensions and plugins. Plugins can modify or extend features of DNF or provide additional CLI commands on top of those mentioned below. If you know the name of such a command (including commands mentioned bellow), you may find/install the package which provides it using the appropriate virtual provide in the form of dnf command where ¡alias¿ is the name of the command; e.g. dnf-command (repo query) for a repo query command (the same applies to specifying dependencies of packages that require a particular command).

* 1. **Installation Steps:**

1. Start VMware Workstation.
2. Select New File.
3. Drag Fedora 22 ISO image and drop it into the Create a Virtual Machine window.
4. Follow the prompts to complete the installation.

## Questions

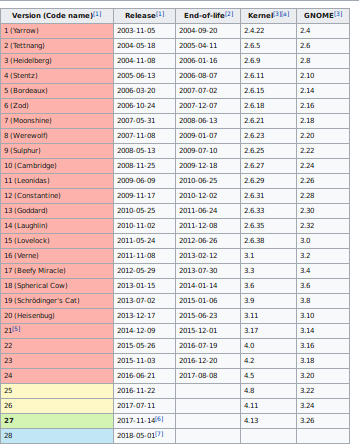
* + 1. Difference between DNF and YUM ?
    2. How to install new packages in Fedora ?
    3. Top 10 commands of Fedora.
  1. ******Versions**

Fig 1 : Version History

## 

## List of Top Commands

1. dnf update Update the system
2. dnf- install package-name –¿ Install the package you want. Re- place package-name with the actual name of the package
3. dnf -d source source-package-name
4. dnf remove package-name
5. dnf reboot

## Difference between Fedora and Ubuntu



Fig 2 : Difference Between Fedora and Ubuntu

## Conclusion

* + 1. Installation of various Linux versions.
    2. Comparision of various distributions have been studied, and their desktop environments ,applications had been studied.

## Reference

[ 1 ] [https://fedoraproject.org/wiki/DNF?rd=Dnf](https://fedoraproject.org/wiki/DNF?rd=Dnf%20)

[ 2 ] <https://fedoraproject.org/wiki/Packagemanagementsystem>

**1.11.1 Video Reference:**

[ 1 ] <https://www.youtube.com/watch?v=aH9YDh8fa0Y>

[ 2 ] <https://www.youtube.com/watch?v=NMzaMVGXjOA>

**Assignment 2**

# Use of Open Source tools for Project Management (Open Atrium)

To use the online and offline Version Control System in Open Source/- for their project work.

## Objective:

To install and demonstrate the use of various open source software’s that used in day to day life of software Engineering.

## Outcome:

Self learning(PO: k, l)

## Theory:

Open Atrium is a set of tools for better team collaboration. It is based on Drupal and uses its strengths to develop a secure, saleable, highly customizable websites. It comes with built-in tools like blog, a wiki, a calendar, a case tracker, a shoutbox and a dashboard etc to successfully develop and manage your websites. You can develop highly interactive, responsive and good looking websites with this tool. The installation process for this application is documented below.

**2.3.1 Features of Atrium:**

These are the main Features of the Open Atrium distribution. There are 6 core features available by default when you install Open Atrium:

* Atrium Blog - Provides a method of sharing stories and information with other members of the group
* Atrium Notebook - Provides a method of building organized documentation within a group
* Atrium Calendar - Adds a group calendar, with the ability to add collective events
* Atrium Case Tracker - Adds the ability to manage and track issues, assign them to members, and discuss them.

Atrium Shoutbox - Provides a rapid discussion board for short discussion or comments. Spaces Dashboard - Adds a configurable dashboard to a group home page in order to easily review new or updated information

## Installation:

* + 1. **Installing Lamp-Stack:**

1.sudo apt-get install lamp-server

2.sudo /etc/init.d/apache2 start

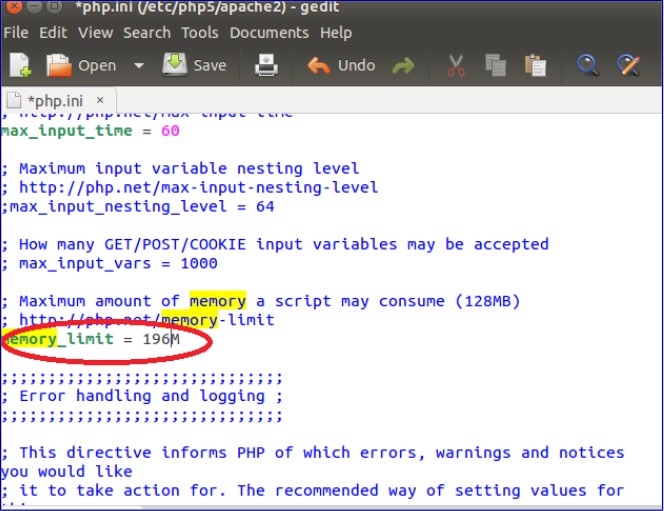
* + 1. **Conftguring PHP :**
       1. sudo gedit /etc/php5/apache2/php.ini
       2. sudo /etc/init.d/apache2 restart

Fig 3 : Modifying the file

* + 1. **Creating Database for Atrium:**

ysql -u root -p reate database openatrium;

* + 1. **Installing Atrium:**
       1. sudo mv openatrium-7.x-2.44-core.tar.gz /var/www/html
       2. mcd /var/www/html
       3. sudo tar xvf openatrium-7.x-2.44-core.tar.gz
       4. sudo mv openatrium-7.x-2.44 openatrium
       5. sudo chmod -R 777 openatrium
       6. cd sites/default
       7. sudo cp default.settings.php settings.php
       8. The rest of the installation process is web based, launch your browser and load <http://localhost/openatrium>

## Oral

1. What is project management tool ?
2. What are various project management tools ?
3. Difference between Sonar and OpenAtrium ?

## Conclusion :

* + - 1. Learn to use Various OpenSource project management tools.
      2. Compared the Open Atrium with other project management tools.
      3. Studied their key features.

## Reference

[1] [https://www.drupal.org/project/atrium*features*](https://www.drupal.org/project/atriumfeatures)

[2] <http://linuxpitstop.com/install-open-atrium-on-ubuntu-linux>

**Assignment 3**

# Use of Bug Tracking (Phabricator)

## Objective: -

To install and demonstrate the use of various open source softwares that used in day to day life of software engg.

## Outcome: -

(PO: k, l)

## Instruction for the Assignment

Students have to study at last 3 various open source tools / softwares that they uses and their day to day life, with their installation and configuration on their system

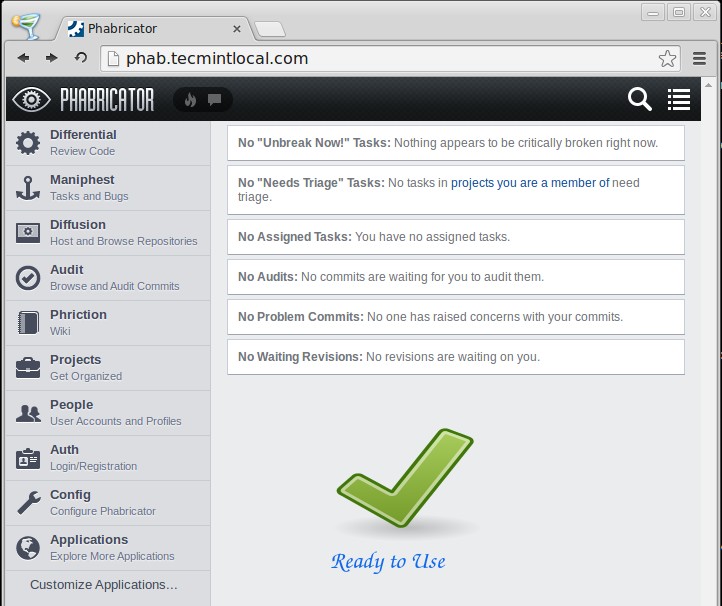
## Theory

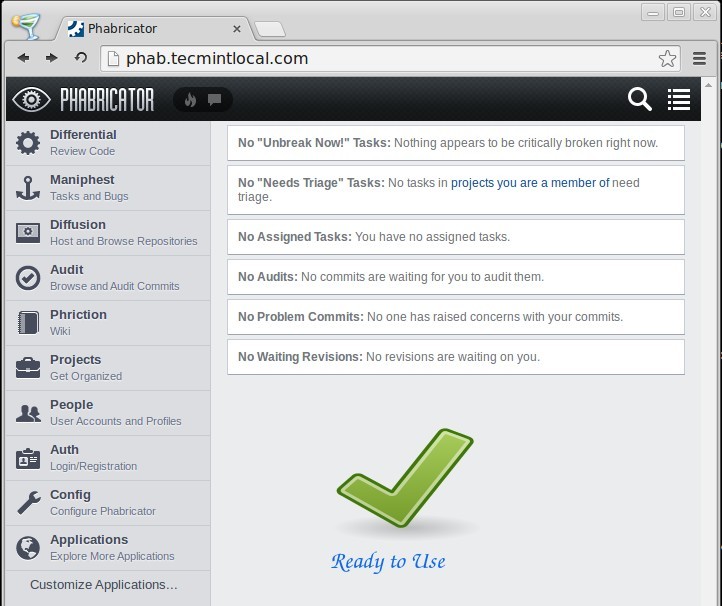
Phabricator is an open source application that helps software com- panies to create/build better software, which is built using PHP language and available under Apache 2.0 open source license for Linux, MacOSX and can be run in any platform, it can even run in windows but it is totally based on Linux support. Phabricator has been used by Facebook before. The first version of phabricator was built by facebook with lots of features such as reviewing and auditing codes, tracking bugs etc. Phabricator (pronounced like the word fabricator) is a suite of web applications which make it easier to build software, particularly when working with teams.

Phabricator is largely based on Facebook’s internal tools. The major components of Phabricator are:

**Differential**, a code review tool; and **Diffusion**, a repository browser; and **Maniphest**, a bug tracker; and **Phriction**, a wiki.

Phabricator also includes a number of smaller tools.





## Installation Requirements

You will need a computer. Options include:

**A Normal Computer:** This is strongly recommended. Many in- stalls use a VM in EC2. Phabricator installs properly and works well on a normal computer.

**A Shared Host:** This may work, but is not recommended. Many shared hosting environments have restrictions which prevent some of Phabricator’s features from working. Consider using a normal computer instead. We do not support shared hosts.

**A SAN Appliance, Network Router, Gaming Console, Rasp- berry Pi, etc.:** Although you may be able to install Phabricator on specialized hardware, it is unlikely to work well and will be difficult for us to support. Strongly consider using a normal computer instead. We do not support specialized hardware.

**A Toaster, Car, Firearm, Thermostat, etc.:** Yes, many mod- ern devices now have embedded computing capability. We live in interesting times. However, you should not install Phabricator on these devices. Instead, install it on a normal computer. We do not support installing on noncomputing devices.

To install the Phabricator server software, you will need an operating system on your normal computer which is not Windows. Note that the command line interface does work on Windows, and you can use Phabricator from any operating system with a web browser. However, the server software does not run on Windows. It does run on most other operating systems, so choose one of these instead:

**Linux:** Most installs use Linux.

**Mac OS X:** Mac OS X is an acceptable flavor of Linux.

**FreeBSD:** While FreeBSD is certainly not a flavor of Linux, it is a fine operating system possessed of many desirable qualities, and Phabricator will install and run properly on FreeBSD.

**Solaris, etc.:** Other systems which look like Linux and quack like Linux will generally work fine, although we may suffer a reduced ability to support and resolve issues on unusual operating systems.

A general description of what you need to install:

git (usually called ”git” in package management systems) Apache (usually ”httpd” or ”apache2”) (or nginx) MySQL Server (usually ”mysqld” or ”mysql-server”) PHP (usually ”php”)

Required PHP extensions: mbstring, iconv, mysql (or mysqli), curl, pcntl (these might be something like ”php-mysql” or ”php5-mysqlnd”) Optional PHP extensions: gd, apc (special instructions for APC are available below if you have difficulty installing it), xhprof (instruc- tions below, you only need this if you are developing Phabricator)

If you already have LAMP setup, you’ve probably already got ev- erything you need. It may also be helpful to refer to the install scripts above, even if they don’t work for your system.

## How to Use Phabricator Tool

Step 1: install Phabricator and make it track a git repo

Step 2: understand the difference between reviews and audits

Step 3: install Phabricator’s Arcanist (arc) command-line tool

Step 4: make a change to you code using git flow

Step 5: trigger the review process

## Conclusion

Learned how to install and demonstrate the use of various open source software’s like bug tracking tools (Phabricator, mantis, etc.)that used in day to day life of software engineering.

## Reference

1. <https://bitnami.com/stacks/Phabricator>

Wikipedia List Of Softwares:-h[ttp://www.redmine.org/projects/Phabricator](http://www.Phabricator.org/projects/Phabricator)

**Assignment 4**

# Use of Version Control System (Git)

To use the online and offline Version Control System in Open Source/- for their project work.

## Objective : -

To Install and Demonstrate the use of various Open Source Soft- wares that used in day to day life of Software Engineering.

## Outcome : -

Self learning (PO: k, l)

## Instruction for the Assignment

Download and install GIT software version control.

On Ubuntu

## $sudo apt-get install git

## Theory :

Git is a free and open source distributed version control system de- signed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Sub- version, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows. Git is used by big faces like Google, Facebook, Microsoft, Twitter, Net- flix, Android, Gnome, QT, Rails, etc. Git is a member of Software Freedom Conservancy. It’s repository is [hosted](https://github.com/git/git) on GitHub.

* + 1. **Features of GIT:**
       - Git is free and Open Source.
       - Git is based on branching and merging model, which makes it stand apart from every other SCM.
       - Git is small, fast and distributed.
       - Git provides Data Assurance.
       - Git has a staging area, where you can keep track of the devel- opment of your project.

## Steps:

* + 1. **Installation:**
       - sudo apt-get update(Ubuntu)
       - sudo apt-get install git
    2. **Go to git websites https://github.com/ and sign up**
    3. **Initial conftguration**

## git config —global user.name “khemrajrathore”

* + - * config– global user.email [“khemr](mailto:khemrajrathore1997@gmail.com)a[jrathore1997@gmail.com”](mailto:jrathore1997@gmail.com)
    1. **Creating new repository**

**git init** or clone (download)

## git clone *<*link*>*

* + 1. **Other commands to work on**

You can start creating and saving files in your repository, you can

*•*

add them to staging area of Git using the command- **git add** *<***filename***>*. You can check the status of your staging area about the un-

*•*

tracked files, changes yet to be committed, etc using the command-

## git status.

You can commit all your current changes by using the command-

*•*

## git commit –m “message” .

* + - * You can view all commits in your current repository using **git log**. You can create and switch to a new branch using the command- **git checkout “branchname”**

*•*

You can merge changes from one branch into the current branch using the command **git merge “branchname”**

*•*

## Mini Project :Game of Shares

Game of Shares is a share market simulation which could be used to learn basics of Stock Market.

It is a web based platfporm which was built using javascript, HTML CSS.

Link to project - https://github.com/virajvchavan/Game-of-Shares



fig 6: Git

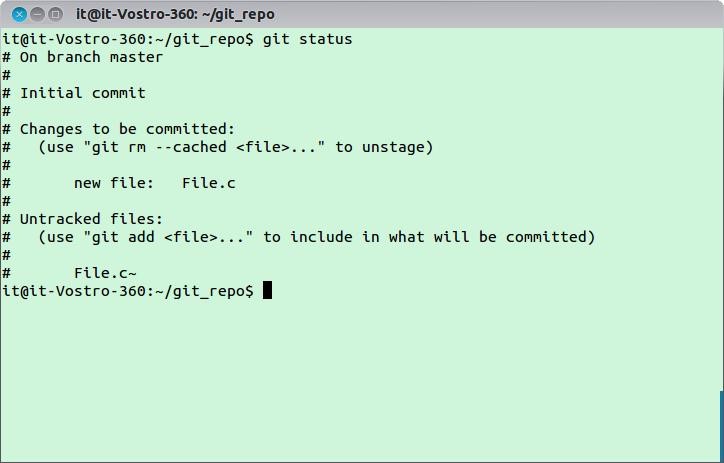


fig 7: Git Status

## Oral

1.What is version control? 2.What is GIT?

1. What is a repository in GIT?
2. What is the command you can use to write a commit message? 5.What is the difference between GIT and SVN?

6.What are the advantages of using GIT? 7.What language is used in GIT?

1. In Git, which commands make remote calls and which are local commands?
2. Explain what is commit message? 10.What is the function of git clone?

## Conclusion :

1.We learnt different VCS tools. 2.Compared Git with other VCS.

3.Overall use of GIT is topp-notch and productive.

## Reference

[ 1 ] <https://en.wikipedia.org/wiki/Git>

[ 2 ] <http://git-scm.com/documentation>

[ 3 ] <https://www.git-tower.com/blog/git-cheat-sheet/>

**Assignment 5**

# Installation and Use of CMS softwares(Joomla)

## Objective :-

To comprehend the use of Content Management System and their Use for personal website/dept CMS.

## Outcome :-

Self learning (PO: b, I, j, k, l)

## Instruction for the Assignment

Students have to study at least 1 CMS such as, word press. Moodle, Drupal, Joomala, MediaWiki, etc

## Theory

A content management system (CMS) manages the creation and modification of digital content. It typically supports multiple users in a collaborative environment. CMS features vary widely. Most CMSs include Web-based publishing, format management, history editing and version control, indexing, search, and retrieval. By their nature, content management systems support the separation of con- tent and presentation.

A web content management system (WCM or WCMS) is a CMS designed to support the management of the content of Web pages. Most popular CMSs are also WCMSs. Web content includes text and embedded graphics, photos, video, audio, maps, and program code (e.g., for applications) that displays content or interacts with the user.

Joomla! is a free and open-source content management system (CMS) for publishing web content, developed by Open Source Mat- ters, Inc. It is built on a model–view–controller web application framework that can be used independently of the CMS.

Joomla! is written in PHP, uses object-oriented programming (OOP) techniques (since version 1.5) and software design patterns, stores data in a MySQL, MS SQL (since version 2.5), or PostgreSQL (since version 3.0) database, and includes features such as page caching, RSS feeds, printable versions of pages, news flashes, blogs, search, and support for language internationalization.

## Joomla

* + 1. **History**

Joomla is based on Mambo CMS which was developed by an Australian company in 2001 and initially released on August 17, 2005. The official version of Joomla 1.0 was released on September 22, 2005.

* + 1. **Features**

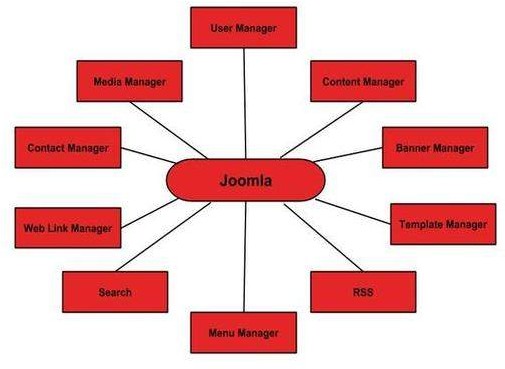


fig 8: Features of Joomla

**User Manager -** It allows managing the user information such as permission to edit, access, publish, create or delete the user, change the password and languages. The main part of the user manager is Authentication.

*•*

**Content Manager -** It allows managing the content using WYSIWYG editor to create or edit the content in a very simple way.

*•*

**Banner Manager -** It is used to add or edit the banners on the website.

*•*

**Template Manager -** It manages the designs that are used on the website. The templates can be implemented without changing the content structure within a few seconds.

*•*

**Media Manager -** It is the tool for managing the media files and folder in which you can easily upload, organize and manage your media files into your article editor tool.

*•*

**Contact Manager -** It allows to add contacts, managing the contact information of the particular users.

*•*

**Web Link Manager -** The link resource is provided for user of the site and can be sorted into categories.

*•*

**Search -** It allows users to search the appropriate information on the site. You can use smart indexing, advanced search options, auto suggest searches to make Joomla search best.

*•*

**Menu Manager -** It allows to create menus and menu items and can be managed subsequently. You can put menu in any style and in multiple places.

*•*

**RSS -** It stands for Really Simple syndication which helps your site contents and RSS files to be automatically updated.

*•*

* + 1. **Advantages**
       - It is an open source platform and available for free.

Joomla is designed to be easy to install and set up even if you’re not an advanced user.

Since Joomla is so easy to use, as a web designer or developer, you can quickly build sites for your clients. With minimal in- structions to the clients, clients can easily manage their sites on their own.

*•*

It is very easy to edit the content as it uses WYSIWYG editor (What You See Is What You Get is a user interface that al- lows the user to directly manipulate the layout of the document without having a layout command).

*•*

*•*

It ensures the safety of data content and doesn’t allow anyone to edit the data.

*•*

* + - * By default, Joomla is compatible with all browsers.
      * The templates are very flexible to use.
      * Media files can be uploaded easily in the article editor tool.
      * Provides easy menu creation tool.
    1. **Disadvantages**

It gives compatibility problem while installing several modules, extensions and plugins simultaneously.

*•*

* + - * Plugins and modules are not free in Joomla.

Development is too difficult to handle when you want to change the layout.

*•*

* + - * Joomla is not much SEO (Search Engine Optimization) friendly.
      * It makes website heavy to load and run.

## Real World Examples of What Joomla Can Create?

* + - Corporate web sites or portals
    - Corporate intranets and extranets
    - Online magazines, newspapers, and publications
    - E-commerce and online reservations
    - Government applications
    - Small business web sites
    - Non-profit and organizational web sites
    - Community-based portals
    - School and religious web sites
    - Personal or family homepages

## Conclusion:

* + - 1. We learned what is Joomla, its features, how to install it.

## References:

[ 1 ] https://docs.joomla.org/References

[ 2 ] https://en.wikipedia.org/wiki/Joomla

**Assignment 6**

# Comprehend the Open Source Software De- velopment for any one Linux distro(Ubuntu)

## Objective :-

To comprehend the open source software development.Outcome: Self learning/lifelong learning (PO: b, k, l)

## Theory

* + 1. **The Ubuntu community**

The Ubuntu Forums are a huge and fast-moving online forum for user chat and problem solving. Users can share your technical know- how with others by answering questions on Ask Ubuntu, powered by StackExchange. Canonical is responsible for delivering regular LTS releases for enterprise production use, as well as security up- dates, support and the entire online infrastructure for community interaction.Various Ubuntu communities are :

ubuntu-users - Ubuntu user technical support, not for general discussions

*•*

* + - * edubuntu-users - Edubuntu Users Help and Discussion
      * kubuntu-users - Kubuntu user technical support
      * launchpad-users - Discussion for Launchpad users
      * lubuntu-users - Lubuntu Help and User Discussions
      * ubuntu-studio-users - Ubuntu Studio Users
      * ubuntukylin-users - Ubuntu Kylin Support and User Discussions
      * xubuntu-users - Xubuntu Support and User Discussions

## IRC

There are many real-time discussion channels for the Ubuntu com- munity via the freenode IRC service. Users need to connect to server port to access this service. Each channel has a particular topic, and when user is talking in that channel everyone else sees what he/she says.

Their main channel is ubuntu on irc.freenode.net. In almost every country we have a LoCo team channel, you can join ubuntu-¡iso code of your country¿ to jump into that channel and start talking with people near you. Each channel has a contact person.

## Lists

* + 1. **Mailing List**

If you want to be able to post, most lists will require that you sub- scribe first. For some subjects this also requires that a moderator authorises you to send email. Mailing list etiquette guidelines are designed to keep the productivity of the Ubuntu mailing lists at a high level. The ubuntu-devel mailing list is for discussions among Ubuntu developers about their projects. This is a moderated mail- ing list. Also, announcement-only lists.

* + 1. **Ubuntu security announcement list**

The ubuntu-security-announce receives only announcements of se- curity updates to Ubuntu releases.

* + 1. **Ubuntu announcement list**

The ubuntu-announce list has very few emails and will keep you up-to-date on new releases of Ubuntu.

* + 1. **Ubuntu news**

The ubuntu-news list is a low volume list, primarily for the Ubuntu Weekly News. Every mailing list deals with a specific subject matter (a topic). It is important to limit discussion to that topic so that the mailing list remains productive. Many users read mailing lists by the thread. This means that when reading, emails are placed in a tree structure according to the subject.

* + 1. **Listwashing**

It is the process through which individual entries in mailing lists are removed.

## Ubuntu Wiki

It is community driven. Each team’s Wiki Home Pages provide information about who they are, what they do, when their meet- ings are, and how to contact them. Using these pages, teammates are able to communicate and coordinate projects. It gives informa- tion about software repositories, installing and removing applica- tions, guides to specific applications and free software equivalents. Category-wise browsing options are available. Also users can learn about most commonly used terms in whole wiki.

## Version Control System

Version control is the art of managing changes to information. Bazaar is a new version control system sponsored by Canonical, the com- mercial company behind Ubuntu. In particular, Bazaar is designed to maximize the level of community participation in open source projects. Git is an open source distributed version control system to support the development of the linux kernel. Subversion is an open source version control system which records the history of source files and documents.

## Bug tracking

Ubuntu uses Launchpad to keep track of bugs and their fixes. Launch- pad will search for similar bugs in case the bug has already been reported. If the bug has already been reported, users can mark that bug as also affecting you. Users can also subscribe to the bug re- port to receive updates about progress with fixing it. If the bug has not already been reported fill in the description field with as much information as you can. It’s important that you specify three things:

* + 1. What you expected to happen?
    2. What actually happened?

3.If possible, a minimal series of steps necessary to make it happen, where step 1 is ”start the program”

The preferred way to report a bug is with the apport-cli com- mand. Bugs can be marked as critical, high, medium or low impor- tance. Also they are as marked according to their status as to new or old. Each report has a ID number.

## Documentation

Documentation contains various releases, desktop and server infor- mation and installation guides for various releases depending upon various architectures. The ’HTML’ links above are available in many different languages.. It doesn’t display unsupported releases. Users can find detailed information for a variety of methods for installing Ubuntu. More advanced installation options are covered. Commu- nity wiki help is also available for various tips, tricks and answers to various how to’s. Users can also contribute

## Conclusion

Learnt about ubuntu community and how to contribute to that community.

*•*

* Learnt about mailing lists of ubuntu.
* Learnt about how to track bug in ubuntu.
* Learnt about version control system in ubuntu.

## Reference:

[1[] hrefhttp://www.tuxradar.com/con](http://www.tuxradar.com/content/how-compile-linux-kernel)ten[t/ho](http://www.tuxradar.com/content/how-compile-linux-kernel)w[-compile-linux-kernel](http://www.tuxradar.com/content/how-compile-linux-kernel)

**Assignment 7**

# Compilation of Linux Kernel selected above.

## Objective : -

To demonstrate how to compile Linux Kernel.

## Outcome : -

(PO: b, I, j, k, l)

## Instruction for the Assignment

In this student have to learn compilation and installation of Linux Kernel on their system/VMware/Virtual box (Mint in Lab).

## Theory:

* + 1. **Dependencies:**
       - gcc latest version
       - ncurses development package and
       - system packages should be up-to date
    2. **Commands to install dependencies:**

For gcc:

sudo yum install gcc.

For ncurses development package : sudo yum install ncurses-devel

After installing the above packages then update your system by run- ning the following command:

sudo yum update.

Now download the Linux Kernel 3.0 from kernel.org or by using the below command.

If you are trying with Linux Kernel 3.5 / 3.4.6 / 3.3 / 3.2 / 3.1 /

3.0.x then just replace the name linux-3.0 with linux-3.5 / linux3.4.6

/ linux-3.3 / linux-3.2 / linux-3.1 / linux- 3.0.x in all commands be- low.

wget <http://www.kernel.org/pub/linux/kernel/v3.0/linux-3.0.tar.bz2> Once the download complete move to the directory where you have downloaded the kernel package linux-3.0.tar.bz2. Now extract the

tar le to the location /usr/src/.

To move to the directory, for example if the downloaded package is in your Downloads directory. Use the below command. cd Down- loads/

To extract the tar le run the following command. Type the pass- word for the user when prompted.

sudo tar -xvf linux-3.0.tar.bz2 -C /usr/src/

Now move to the directory where the extracted le is, or copy and paste the below command in terminal. cd /usr/src/linux-3.0/

Now you can congure, compile and install Linux Kernel 3.0 in your system. Run the commands one by one and type the password for the user, when prompted.

sudo make menucong To Compile:

sudo make To Install

sudo make modules install

## Conclusion

Hence from this assignment we comes to know that how to compile linux kernal. We have studied how to build debian package.

## Reference

1. Installation Of Linux Kernel on Mint
2. Latest kernel installation on Fedora and Cent OS:-<http://www.tecmint.com/kernel-3-5-released-install-compile-in-redhat-> Mint-and-fedora/

**Assignment 8**

# Create of RPM or DEB packages

## Objective : -

To Create package building process in Linux

## Outcome : -

(PO: b, I, j, k, l)

## Instruction for the Assignment

Students have to study RPM or DEB package building for their simple C, C++ or JAVA

Codes. They must build an rpm or deb and install it through package manager such as YUM or APT-GET

## Theory (Functions of tool /How to Use /Drawbacks)

A Debian package, or a Debian archive file, contains the executable files, libraries, and documentation associated with a particular suite of program or set of related programs. Normally, a Debian archive file has a filename that ends in .deb.

Debian control file: Package: hello Priority: optional Section: devel Installed-Size: 45 Maintainer: Adam Heath [¡do](mailto:doogie@debian.org)[ogie@debian.org¿](mailto:ogie@debian.org) Architecture: i386 Version: 1.3-16 Depends: libc6 (¿= 2.1) Description: The classic greeting, and a good example The GNU hello pro- gram produces a familiar, friendly greeting. It allows nonprogram- mers to use a classic computer science tool which would otherwise be unavailable to them. . Seriously, though: this is an example of how to do a Debian package. It is the Debian version of the GNU Project’s ‘hello world’ program (which is itself an example for the GNU Project).

The Package field gives the package name. The Version field gives both the upstream developer’s version number. The Architecture field specifies the chip for which this particular binary was compiled. The Depends field gives a list of packages that have to be installed in order to install this package successfully. The Installed-Size indi- cates how much disk space the installed package will consume. The Section line gives the section where this Debian package is stored at the Debian FTP sites. The Priority indicates how important is this package for installation, so that semi-intelligent software like deselect or aptitude can sort the package into a category of e.g. packages optionally installed. The Maintainer field gives the e-mail address of the person who is currently responsible for maintaining this pack- age. The Description field gives a brief summary of the package’s features.

## Steps

* + - #include¡iostream¿ Using namespace std;

Int main()

*{*

Cout¡¡”Hello World”; Return 0;

*}*

Save the above code as helloworld.cc. At this point make sure that you have compiler installed on your system by executing:

$ sudo apt-get install build-essential

* + - Compile and execute your code with a following command: 1.$ g++ helloworld.cc -o helloworld

2.$ ./helloworld

At this point you should have a binary executable called linuxconfig which prints some string on the screen.

Now that we have program ready in form of executable binary we can package it up into a debian package. To do that we would use a dpkg-deb tool. But first we need to create a debian package structure. The only files required in to build a debian package are:

*•*

1. DEBIAN/control
2. custom files to be part of the package ( not required )

First create a directory called Calc. This directory will hold all necessary package files:

$ mkdir helloworld

Next, create a control file: 1.$ cd helloworld

2.$ mkdir DEBIAN

* When ready open up DEBIAN/control file

$ vi DEBIAN/control

and enter a following information: Package: helloworld

Version: 1.0 Section: custom Priority: optional Architecture: all Essential: no Installed-Size: 1024

Maintainer: helloworld.org Description: Display String.

Still in root of helloworld directory create a directory which will be used to install helloworld program and copy program into this directory. Best choice for us will be /usr/bin:

*•*

1.$ mkdir -p usr/bin/

2.$ cp /home/sdk/helloworld usr/bin/

At this point we are ready to create a debian package.

*•*

$ cd ..

$ dpkg-deb –build helloworld

dpkg-deb: building package ‘helloworld’ in ‘helloworld.deb’.

$ ls

helloworld helloworld.deb

To install file

$ sudo apt-get install helloworld

## Quiz

1. What are two types of Debian packages?

2. How do I install a source package?

1. What is a Virtual Package?
2. What is the format of a Debian binary package?
3. What is a Debian conffile?

## Conclusion

1. We Learned about RPM and DEB.
2. We Build our own debian package.
3. We have studied how to build debian package.

## Reference

[1] <http://linuxconfig.org/>

[2]<http://askubuntu.com/questions/90764/>

**8.8.1 Video Reference:**

[1] h[ttps://www.youtube.com/watch?v=lUvgv-waAco](https://www.youtube.com/watch?v=lUvgv-waAco)

**Assignment 9**

# Install and demonstrate of various Server based services and their Uses.

## Objective: -

To know server installations and Configurations on Linux Platform

## Outcome: -

(PO: b, I, j, k, l)

## Instruction for the Assignment

Students are asked to install and configure at least 2 servers, such as FTP, HTTP, TELNET, NFS,NIS etc.All configurations must be done on Linux Platform

## Theory (Functions of tool /How to Use /Drawbacks)

The File Transfer Protocol (FTP) is a standard system convention used to exchange PC documents starting with one host then onto the next host over a TCP-based system, such as the Internet. FTP is based on a customer server design and uses partitioned control and information associations between the customer and the server. The Hypertext Transfer Protocol (HTTP) is an application convention for distributed, community oriented, hypermedia data frame- works. HTTP is the foundation of information correspondence for the World Wide Web.

Telnet (Terminal Network) is a system convention utilized on the Internet or neighborhood to give a bidirectional intelligent content arranged communication office utilizing a virtual terminal association.

The Network Information Service, or NIS (initially called Yellow Pages or YP) is a customer server index benefit convention for circulating framework configuration information, for example, client and host names between PCs on a PC organize.

System File System (NFS) is a conveyed record framework convention initially created by Sun Microsystems in 1984, permitting a client on a customer PC to get to documents over a system much like neighborhood stockpiling is gotten to. NFS, in the same way as other different conventions, expands on the Open Network Computing Remote Procedure Call (ONC RPC) framework.

**9.4.1 Main Functions:**

The capacity of a PC server is to store, recover and send or ”serve” records what’s more, information to different PCs on its system. Numerous organizations of all sizes utilize a neighborhood system or ”intranet” in their office offices.

On a bigger scale, the overall PC arrange we know as the ”Web” relies on countless situated far and wide. The records, information, and usefulness of a given site depend on web servers.

## Steps:

* + 1. **FTP server:**

To install VSFTPD

## sudo apt-get install vsftpd

To open vsftpd.conf file.

**sudo gedit /etc/vsftpd.conf** Edit the 3 lines and save the file **anonymous enable=YES local enable=YES**

## write enable=YES

Restart the server

## sudo service vsftpd restart

Open FTP from the browser. -**ftp://10.10.13.13(IP)**

* + 1. **HTTP server**

## $ sudo apt-get install apache2

Now type [”http://local-host”](http://local-host/) in browser

If ”it works” message occurs then it means apache is in- stalled

* + 1. **NIS server**

To install nis

## sudo apt-get install nis

To open nis file

**sudo gedit /etc/defaults/nis** Edit the line and save the file **NISSERVER= master**

To open makefile file.

## sudo gedit /var/yp/makefile

## Add the line and save the file

## ALL = passwd shadow group hosts rpc services netid protocols netgrp

update NIS database

## /usr/lib/yp/ypinit -m

To start server

## service ypserv start

* + 1. **Telnet:**

## $ sudo apt-get install telnetd

Restart inetd service:

## $ sudo /etc/init.d/openbsd-inetd restart

connect to telnet server from any other computer machine

## $ telnet (IP)10.10.13.13

* + 1. **NFS server:**

To install nfs.

## sudo apt-get install nfs-kernel-server

To open idmapd.conf file.

## sudo gedit /etc/idmapd.conf

Uncomment and change to your domain name

## Domain = ravi.itdept

To start service

## sudo initctl restart idmapd

Mount the directory.

## sudo mount -t nfs ram.itdept:/home /home

Change home directory this server mounts to the one on NFS in

## ram.itdept:/home /home nfs defaults 0 0

## Screenshot

## Quiz

1. What is FTP?
2. On which port VSFTP server works ?
3. How to restart VSFTP server ?

4. How to allow Anonymous FTP ? 5.What is Passive mode?

6. Explain directive “session support”?

7. What is FTP ?

How to deny specific users access to the FTP server ?

## Conclusion:

1. Considered Installation and design of FTP, HTTP, NIS, NFS and TELNET servers.
2. Learned about various Server based services.

## Reference:

* + 1. <http://www.server-world.info/en/>

**Assignment 10**

# Development of new Open Source Software or contribution to existing Open Source Soft- ware.

## Objective : -

To know how open source software is developed with Licensing

## Outcome : -

Self learning/lifelong learning (PO: b, I, j, k, l)

## Instruction for the Assignment :

Develop simple software for basic needs such as editor etc.

## Theory :-

A Debian ”bundle”, or a Debian document record, contains the ex- ecutable files, libraries, and documentation related with a specific suite of program or set of related projects. Ordinarily, a Debian file document has a filename that closes in .deb.

Debian control record: Package: hi Priority: discretionary Sec- tion: devel Installed- Measure: 45 Maintainer: Adam Heath ¡doo- [gie@debian.org¿](mailto:gie@debian.org) Architecture: i386 Variant: 1.3-16 Depends: libc6 (¿= 2.1) Description: The great welcome, and a decent illustration The GNU hi program creates a recognizable, amicable welcome. It enables nonprogrammers to utilize an exemplary software engi- neering instrument which would generally be inaccessible to them. instructions to do a Debian bundle. Truly, however: this is a case of It is the Debian rendition of the GNU Project’s ‘hello world’ pro- gram (which is itself a case for the GNU Project).

The Package field gives the bundle name. The Version field gives both the upstream engineer’s form number. The Architecture field indicates the chip for which this specific twofold was gathered. The Depends field gives a rundown of bundles that must be introduced so as to introduce this bundle achievement completely. The Installed- Size shows how much circle space the introduced bundle will expend. The Section line gives the ”area” where this Debian bundle is put away at the Debian FTP locales. The Priority shows how impera- tive is this bundle for establishment, with the goal that semi-clever programming like dselect or fitness can sort the bundle into a class of e.g. bundles alternatively introduced. The Maintainer field gives the email address of the individual who is as of now responsible for keeping up this bundle. The Description field gives a short rundown of the bundle’s highlights.

## Steps:

* + 1. **Source code for Tic Tac Toe Game in c++:**

**#include***<*b i t s / s t d c ++.h*>*

**using namespace** s t d ;

**#define** COMPUTER 1

**#define** HUMAN 2

**#define** SIDE 3 *// Leng th o f t h e b o a rd*

*// Computer w i l l move w i t h ’O ’*

*// and human w i t h ’X ’*

**#define** COMPUTERMOVE ’O ’

**#define** HUMANMOVE ’X ’

*// A f u n c t i o n t o show t h e c u r r e n t b o a rd s t a t u s*

**void** showBoard ( **char** board [ ] [ SIDE ] )

{

p r i n t f ( ”\n\n” ) ;

p r i n t f ( ”\ t \ t \ t %c | %c | %c \n” , board [ 0 ] [ 0 ] ,

board [ 0 ] [ 1 ] , board [ 0 ] [ 2 ] ) ;

p r i n t f ( ”\ t \ t \ t**− − − − − − −** \n” ) ;

p r i n t f ( ”\ t \ t \ t %c | %c | %c \n” , board [ 1 ] [ 0 ] ,

board [ 1 ] [ 1 ] , board [ 1 ] [ 2 ] ) ;

p r i n t f ( ”\ t \ t \ t**− − − − − − −** \n” ) ;

p r i n t f ( ”\ t \ t \ t %c | %c | %c \n\n” , board [ 2 ] [ 0 ] ,

board [ 2 ] [ 1 ] , board [ 2 ] [ 2 ] ) ;

**return** ;

}

*// A f u n c t i o n t o show t h e i n s t r u c t i o n s*

**void** s h o w I n s t r u c t i o n s ( )

{

p r i n t f ( ”\ t \ t \ t Tic−Tac−Toe\n\n” ) ;

p r i n t f ( ” Choose a c e l l numbered from 1 t o 9 a s below ”

” and p l a y \n\n” ) ;

p r i n t f ( ”\ t \ t \ t 1 | 2 | 3 \n” ) ;

**return** ( **f a l s e** ) ;

}

**i f** ( board [ i ] [ 0 ] == board [ i ] [ 1 ] &&

board [ i ] [ 1 ] == board [ i ] [ 2 ] &&

board [ i ] [ 0 ] != ’ ’ )

**return** ( **true** ) ;

{

**f or** ( **int** i =0; i *<*SIDE ; i ++)

{

}

*// A f u n c t i o n t h a t r e t u r n s t r u e i f any o f t h e row*

*// i s c r o s s e d w i t h t h e same p l a y e r ’ s move*

**bool** ro w Cro ss e d ( **char** board [ ] [ SIDE ] )

p r i n t f ( ”HUMAN has won\n” ) ;

**return** ;

p r i n t f ( ”COMPUTER has won\n” ) ;

**e l s e**

**i f** ( whoseTurn == COMPUTER)

{

*// randomise t h e moves*

r a n d o m s h u f f l e ( moves , moves + SIDE∗SIDE ) ;

**return** ;

}

*// A f u n c t i o n t o d e c l a r e t h e winner o f t h e game*

**void** d e c l a r e W i n n e r ( **int** whoseTurn )

moves [ i ] = i ;

**f or** ( **int** i =0; i *<*SIDE∗SIDE ; i ++)

**f or** ( **int** j =0; j *<*SIDE ; j ++)

board [ i ] [ j ] = ’ ’ ;

}

*// F i l l t h e moves w i t h numbers*

{

*// I n i t i a t e t h e random number g e n e r a t o r so t h a t*

*// t h e same c o n f i g u r a t i o n doesn ’ t a r i s e s*

s ra n d ( ti me (NULL ) ) ;

*// I n i t i a l l y t h e b o a rd i s empty*

**f or** ( **int** i =0; i *<*SIDE ; i ++)

{

}

*// A f u n c t i o n t o i n i t i a l i s e t h e game*

**void** i n i t i a l i s e ( **char** board [ ] [ SIDE ] , **int** moves [ ] )

p r i n t f ( ”\ t \ t \ t**− − − − − − −** \n” ) ;

p r i n t f ( ”\ t \ t \ t 7 | 8 | 9 \n\n” ) ;

p r i n t f ( ”−\t −\t −\t −\t −\t −\t −\t −\t −\t −\n\n” ) ;

**return** ;

p r i n t f ( ”\ t \ t \ t 4 | 5 | 6 \n” ) ;

p r i n t f ( ”\ t \ t \ t**− − − − − − −** \n” ) ;

**int** moves [ SIDE∗SIDE ] ;

*// I n i t i a l i s e t h e game*

i n i t i a l i s e ( board , moves ) ;

*// Show t h e i n s t r u c t i o n s b e f o r e p l a y i n g*

s h o w I n s t r u c t i o n s ( ) ;

**int** moveIndex = 0 , x , y ;

*// A 3*∗*3 Tic*−*Tac*−*Toe b o a rd f o r p l a y i n g*

**char** board [ SIDE ] [ SIDE ] ;

{

*// A f u n c t i o n t o p l a y Tic*−*Tac*−*Toe*

**void** playTicTacToe ( **int** whoseTurn )

| | d i a g o n a l C r o s s e d ( board ) ) ;

}

**return** ( ro w Cro sse d ( board ) | | columnCrossed ( board )

{

**i f** ( board [ 0 ] [ 0 ] == board [ 1 ] [ 1 ] &&

board [ 1 ] [ 1 ] == board [ 2 ] [ 2 ] &&

board [ 0 ] [ 0 ] != ’ ’ )

**return** ( **true** ) ;

**i f** ( board [ 0 ] [ 2 ] == board [ 1 ] [ 1 ] &&

board [ 1 ] [ 1 ] == board [ 2 ] [ 0 ] &&

board [ 0 ] [ 2 ] != ’ ’ )

**return** ( **true** ) ;

**return** ( **f a l s e** ) ;

}

*// A f u n c t i o n t h a t r e t u r n s t r u e i f t h e game i s o v e r*

*// e l s e i t r e t u r n s a f a l s e*

**bool** gameOver ( **char** board [ ] [ SIDE ] )

{

**return** ( **f a l s e** ) ;

}

*// A f u n c t i o n t h a t r e t u r n s t r u e i f any o f t h e d i a g o n a l*

*// i s c r o s s e d w i t h t h e same p l a y e r ’ s move*

**bool** d i a g o n a l C r o s s e d ( **char** board [ ] [ SIDE ] )

}

**i f** ( board [ 0 ] [ i ] == board [ 1 ] [ i ] &&

board [ 1 ] [ i ] == board [ 2 ] [ i ] &&

board [ 0 ] [ i ] != ’ ’ )

**return** ( **true** ) ;

{

**f or** ( **int** i =0; i *<*SIDE ; i ++)

{

}

*// A f u n c t i o n t h a t r e t u r n s t r u e i f any o f t h e column*

*// i s c r o s s e d w i t h t h e same p l a y e r ’ s move*

**bool** columnCrossed ( **char** board [ ] [ SIDE ] )

*// Keep p l a y i n g t i l l t h e game i s o v e r or i t i s a draw*

**while** ( gameOver ( board ) == **f a l s e** && moveIndex != SIDE∗SIDE )

{

**i f** ( whoseTurn == COMPUTER)

{

x = moves [ moveIndex ] / SIDE ;

y = moves [ moveIndex ] % SIDE ; board [ x ] [ y ] = COMPUTERMOVE;

COMPUTERMOVE, moves [ moveIndex ] + 1 ) ;

showBoard ( board ) ; moveIndex ++; whoseTurn = HUMAN;

p r i n t f ( ”COMPUTER has put a %c i n c e l l %d\n” ,

}

**e l s e**

{

**i f** ( whoseTurn == HUMAN)

x = moves [ moveIndex ] / SIDE ;

y = moves [ moveIndex ] % SIDE ; board [ x ] [ y ] = HUMANMOVE;

HUMANMOVE, moves [ moveIndex ] + 1 ) ;

showBoard ( board ) ; moveIndex ++; whoseTurn = COMPUTER;

p r i n t f ( ”HUMAN has put a %c i n c e l l %d\n” ,

}

}

*// I f t h e game has drawn*

**i f** ( gameOver ( board ) == **f a l s e** && moveIndex == SIDE ∗ SIDE )

**e l s e**

{

p r i n t f ( ” I t ’ s a draw \n” ) ;

*// T o g g l i n g t h e u s e r t o d e c l a r e t h e a c t u a l*

*// winner*

**i f** ( whoseTurn == COMPUTER) whoseTurn = HUMAN;

**e l s e i f** ( whoseTurn == HUMAN) whoseTurn = COMPUTER;

*// D e c l a r e t h e winner*

d e c l a r e W i n n e r ( whoseTurn ) ;

}

**return** ;

}

*// D r i v e r program*

**int** main ( )

{

*// Let us p l a y t h e game w i t h COMPUTER s t a r t i n g f i r s t*

playTicTacToe (COMPUTER) ;

**return** ( 0 ) ;

}

* + 1. **Steps for create deb package:**

Spare the above code as Tic-Tac-Toe.cpp. Now ensure that you have compiler introduced on your framework by executing:

## sudo apt-get install build-essential

* + 1. **Compile and execute your code with a following command:**
       - $ g++ Tic-Tac-Toe.cpp -o Tic
       - $ ./Tic

Now you ought to have a double executable called linuxcong which prints some string on the screen.

* + - * DEBIAN/control
      * custom files to be part of the package ( not required )

First create a directory called ”Tic”.This directory will hold all nec- essary package files:

* + - * $ mkdir Tic
    1. **Next, create a control ftle:**
       - $ cd Tic
       - $ mkdir DEBIAN
    2. **When ready open up DEBIAN/control ftle**

$ vi DEBIAN/control

and enter a following information: Package: Tic

Version: 1.0 Section: custom Priority: optional Architecture: all Essential: no Installed-Size: 1024

Maintainer: Calc.org Description: Display Calculator

Link to git : https://github.com/virajvchavan/Tic

* + - * $ mkdir -p usr/bin/
      * $ cp /home/sdk/Tic usr/bin/
    1. **At this point we are ready to create a debian package.**
       - $ cd ..
       - $ dpkg-deb -build
       - $ dpkg-deb: building package ‘Tic’ in ‘Tic.deb’.
       - $ ls
       - $ Tic Tic.deb
    2. **To install ftle**

$ sudo apt-get install Tic

## Conclusion

1.we have examined how to construct debian bundle for number cruncher application.

## References

1. GIT version control Tutorial:- <http://git-scm.com/documentation>
2. SVN :- [http://michael-zamir.blogspot.in/2012/01/svn-](http://michael-zamir.blogspot.in/2012/01/svn-tutorial.html)tutorial. [html](http://michael-zamir.blogspot.in/2012/01/svn-tutorial.html)
3. Translation :- [http://www.tuxamito.com/joomla/index.php/es/ component/content/article/60-gettext-tutorial](http://www.tuxamito.com/joomla/index.php/es/component/content/article/60-gettext-%20tutorial)
4. Using Malarad:- [http://projectmallard.org/abo](http://projectmallard.org/about/learn/tenminutes.html)ut/learn/tenminutes. [html](http://projectmallard.org/about/learn/tenminutes.html)

**Assignment 11**

# Docker Container: An open source software development platform

## Objective:

To understand and use the docker virtualization as OSS.

With the help of Docker-compose deploy the ‘Wordpress’ and ‘Mysql container and access the front end of ‘Wordpress’

*•*

* Docker image:
  + Create a simple Hello-world python flask application and cre- ate the docker image of that Flask application.
  + Run the docker container from recently created image and run that docker container to 5000 port of host system.

Create the ‘nginx’ container from ‘nginx’ image. And create the load balancing so that if we go to tha address of ‘nginx’ it can redirect it to the above created applications (Flask and Wordpress).

*•*

## Introduction

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker’s methodologies for shipping, testing, and deploying code quickly, you can signif- icantly reduce the delay between writing code and running it in production

## Docker Features

1.Easy and Faster Configuration 2.Increase productivity 3.Application Isolation

4.Swarm 5.Routing Mesh

6.Security Management

## Installation

* + 1. **Uninstall old versions**

1.sudo apt-get remove docker docker-engine docker.io

* + 1. **Install Docker CE**

1. sudo apt-get update
2. sudo apt-get install docker-ce 3.apt-cache madison docker-ce

docker-ce — 17.09.0 ce-0 ubuntu — https://download.docker.com/linux/ubuntu

xenial/stable amd64 Packages

## Wordpress through docker Screenshots

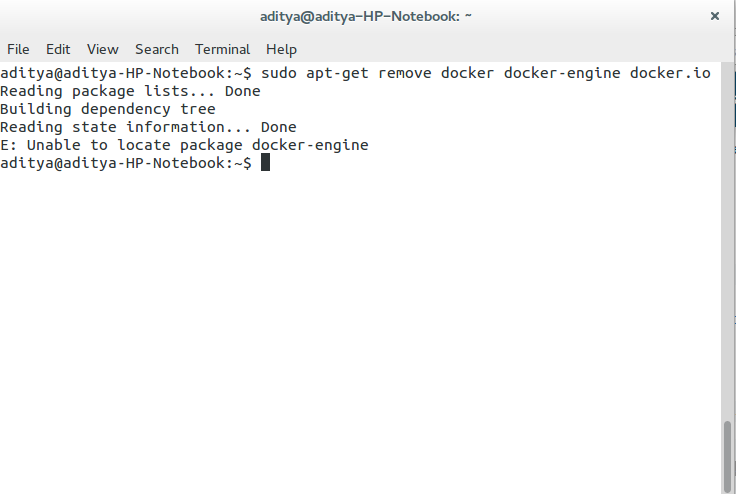


fig 14: uninstall previous dockers

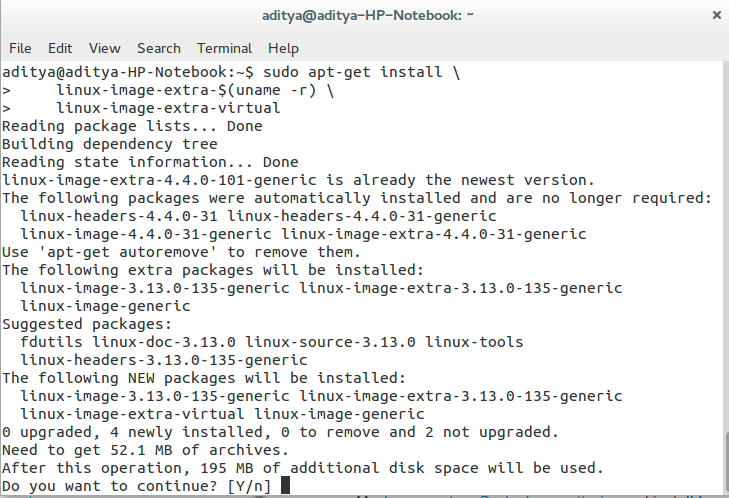
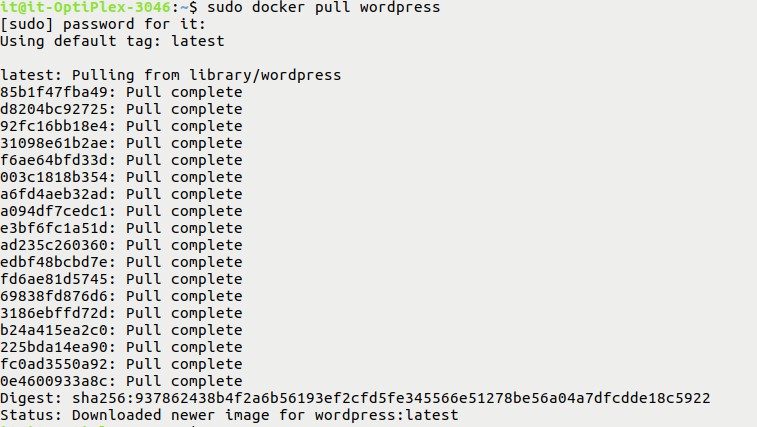


fig 15: install Docker



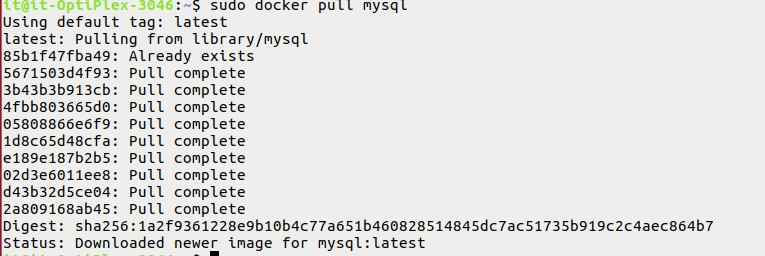
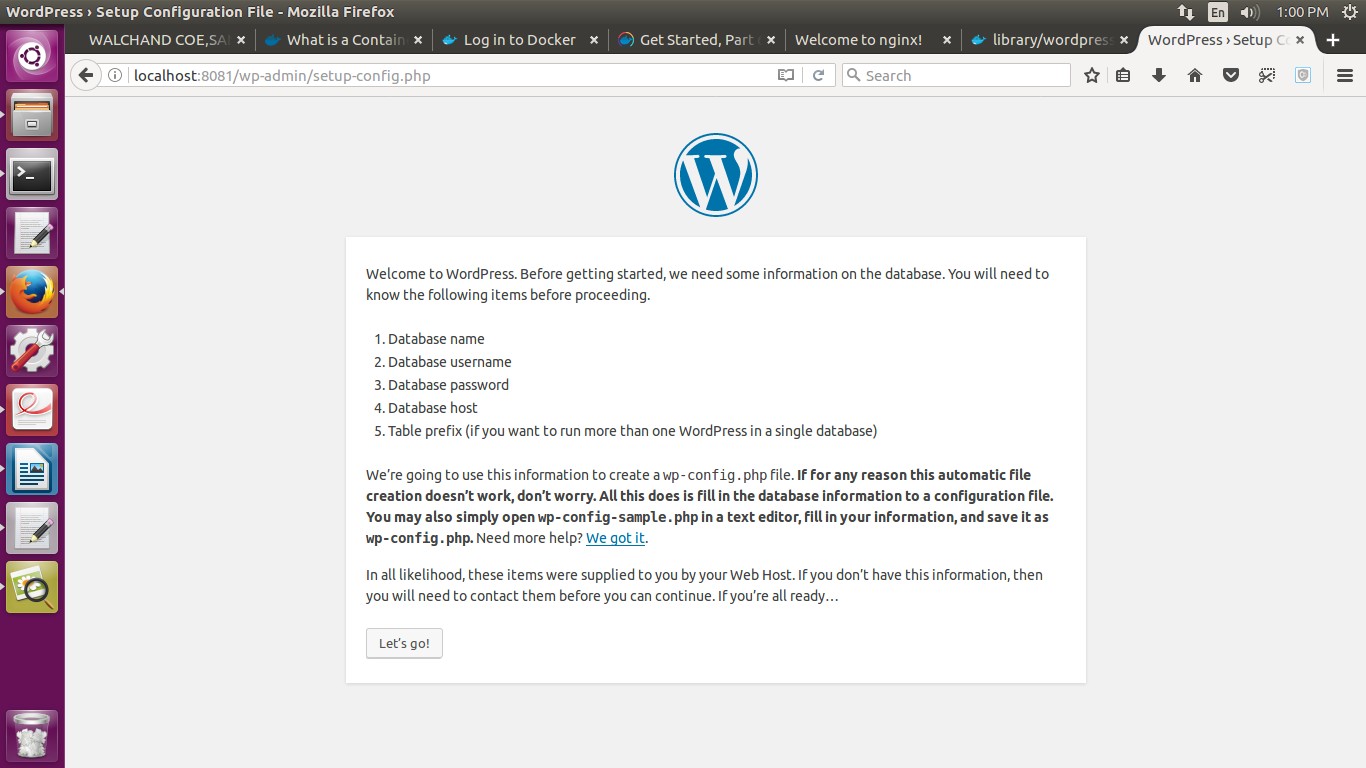
fig 16: pull Docker image of Wordpress

fig 17: pull docker image of MySQl



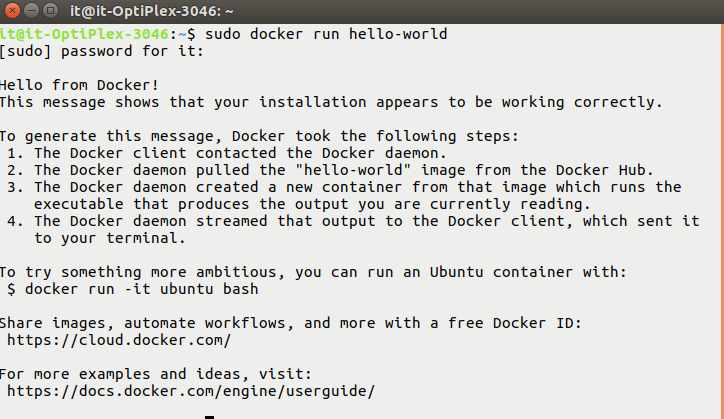
fig 18: Wordpress

fig 19: Running Docker Container

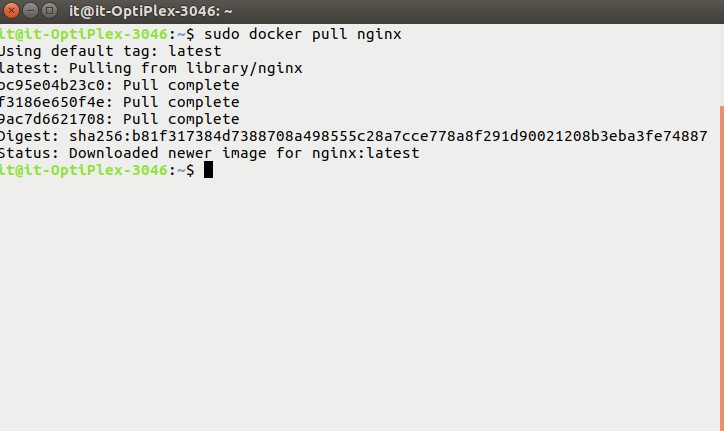


fig 20: Pulling nginx container

## Quiz

1.What is different between a Docker container and a VM? 2.How far do Docker containers scale?

1. What platforms does Docker run on?
2. How do I report a security issue with Docker? 5.Does Docker run on Linux, macOS, and Windows?

## Conclusion :

1. We learned about Docker Container.
2. We used Docker Containers for Wordpress and MySql.

## Refference :

[1]h[ttps://www.do](http://www.docker.com/)ck[er.com/](http://www.docker.com/) [2]https://opensource.com/resources/what-docker

**Assignment 12**

# Find python kernel code and compile it or use any python library for any application. (Submission by Individual or Group [I])

## Objective : -

To know how open source software is developed with Licensing

## Outcome : -

Self learning/lifelong learning (PO: b, I, j, k, l)

## Instruction for the Assignment :

Develop simple software for basic needs such as editor etc.

## Theory

ArangoDB is a native NoSQL multi-model database system devel- oped by triAGENS GmbH. In a book published in 2015, it was referred to as the most popular NoSQL database available that had an open-source license. It has also been referred to as a universal database. Its creators refer to it as a ”native multi-model” database to indicate that it was designed specifically to allow key/value, doc- ument, and graph data to be stored together and queried with a common language.

ArangoDB provides scalable, highly efficient queries when work- ing with graph data. The database uses JSON as a default storage format, but internally it uses ArangoDB’s VelocyPack – a fast and compact binary format for serialization and storage. ArangoDB can natively store a nested JSON object as a data entry inside a col- lection. Therefore, there is no need to disassemble the resulting JSON objects. Thus, the stored data would simply inherit the tree structure of the XML data.

## Feature

1.Clean, Pythonic interface

2.Lightweight

3.High ArangoDB REST API coverage

## Compatibility

1.Python versions 2.7.x, 3.4.x, 3.5.x and 3.6.x are supported

2.Latest version of python-arango (3.x) supports ArangoDB 3.x

only

3.Older versions of python-arango support ArangoDB 1.x 2.x only

## installation

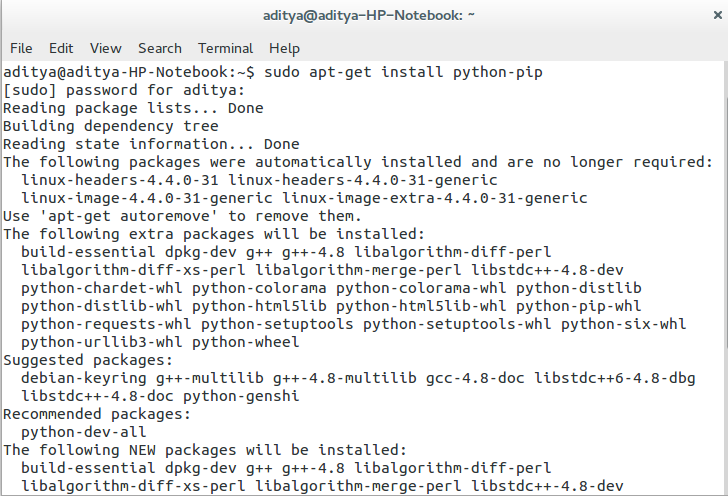
1. pip install python-arango
2. pip install -e git+git@github.com:joowani/python- arango.git@masteregg=python- arango

fig 13: Installation of arango

## Conclusion

1.Used Python Driver for ArangoDB, a NoSQL graph database.

2.Used Python libraries in small applications.

## Reference

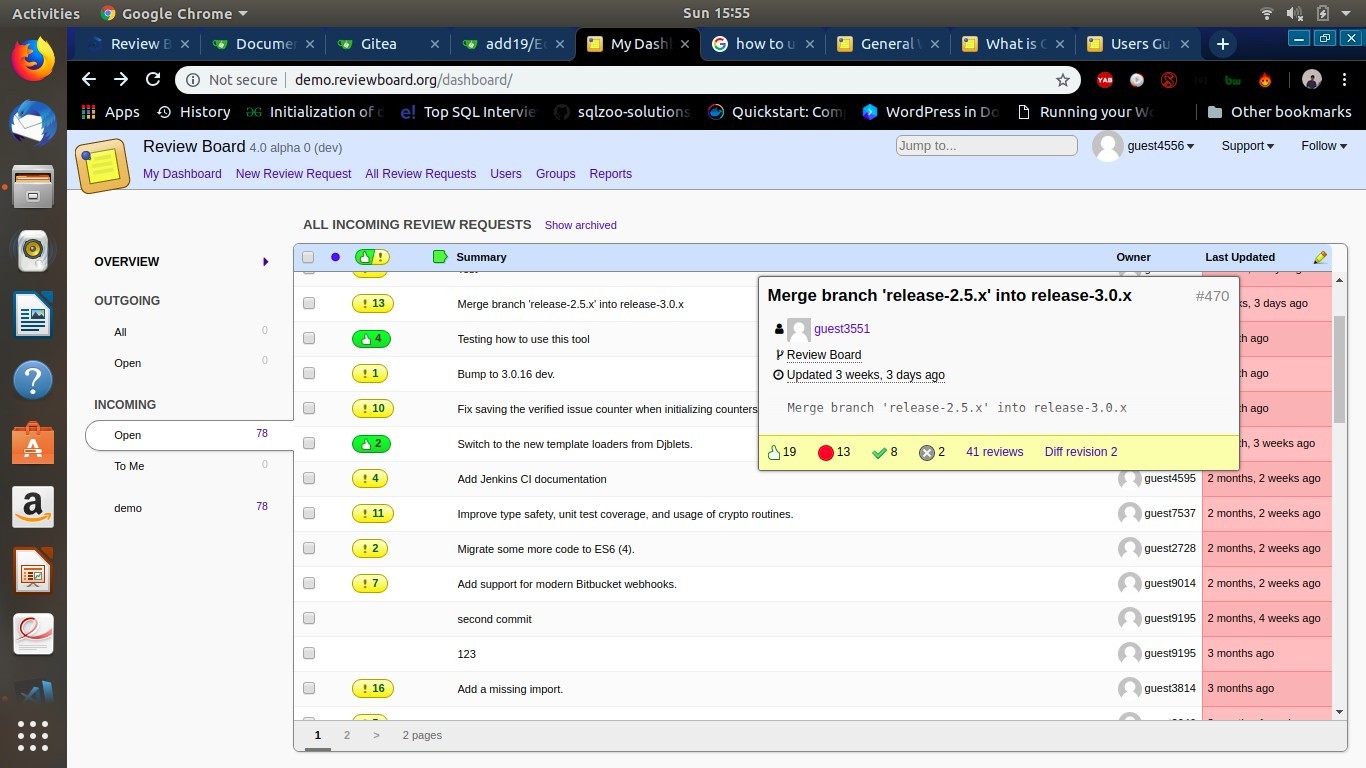
[1] <https://github.com/vinta/awesome-python>

**Assignment 13.**

**13. Agile s/w engineering by using Tuleap, review bord and gitea (tools of agile setup)**

## 13.1. Reviewboard:

Bitnami installers are available. Docker containers are available as well



Usage: It is a tool for managing code review.

What is code review:

Code review is the process of making pieces of source code available for other developers to review, with the intention of catching bugs and design errors before the code becomes part of the product. Code review dramatically helps in the quality of products. By catching mistakes early, a lot of bad problems in a product can be avoided. Not all companies or developers take advantage of code review, but more and more are making it a part of their development culture and requirements.

There are two types of code review: pre-commit and post-commit.

**Pre-commit review**i s a form of code review where code is reviewed *before* going into the codebase. In this method, a diff file is uploaded to Review Board, which reviewers can comment on, and once there’s approval the code is committed to the repository.

**Post-commit review** is where the code is reviewed *after* going into the codebase. The code is committed to the repository and, at some point later, the code is reviewed. Any fixes that need to be made are then committed again later.

General Workflow:

Review Board supports many different workflows, but most people use a pre-commit review model. The general process for using Review Board for pre-commit review is as follows:

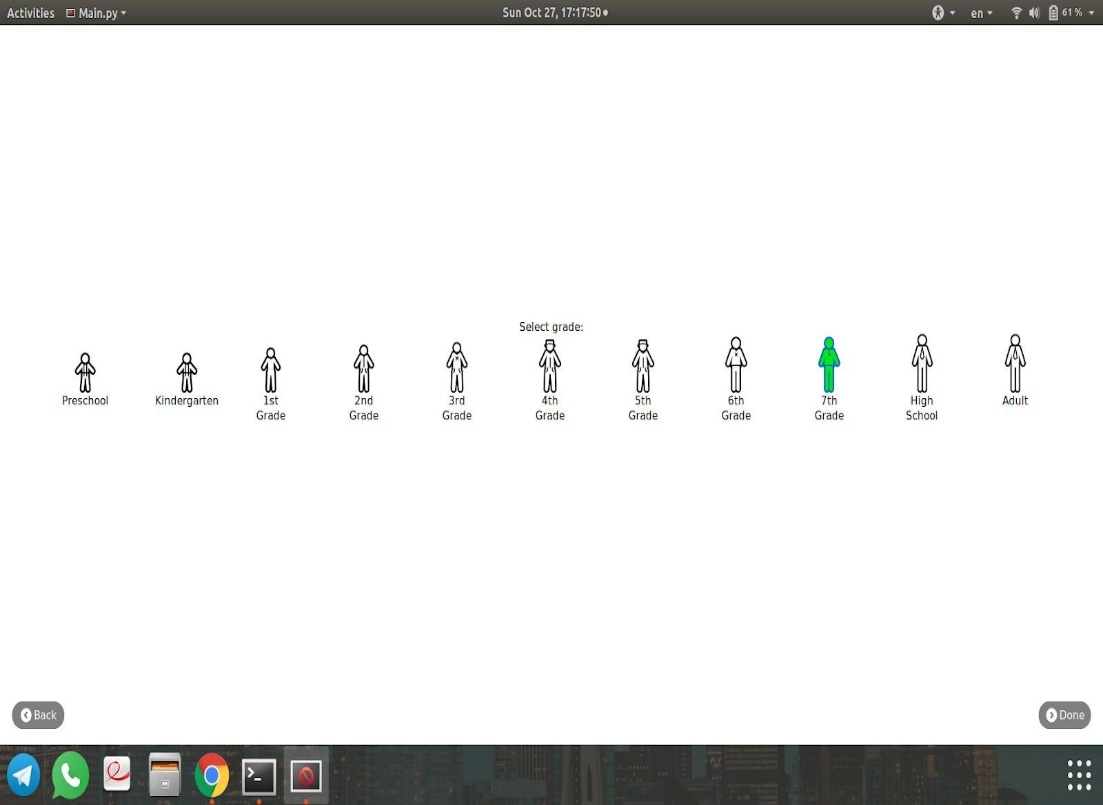
1. Make a change to your local source tree.

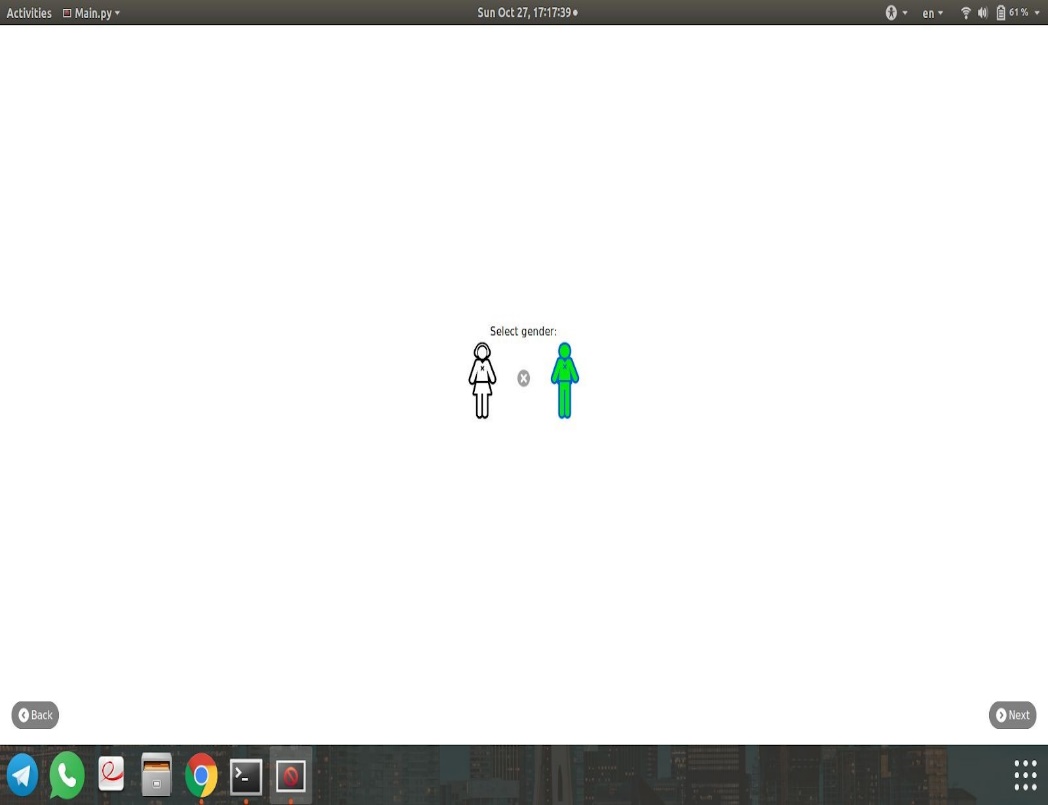
**Assignment 14**

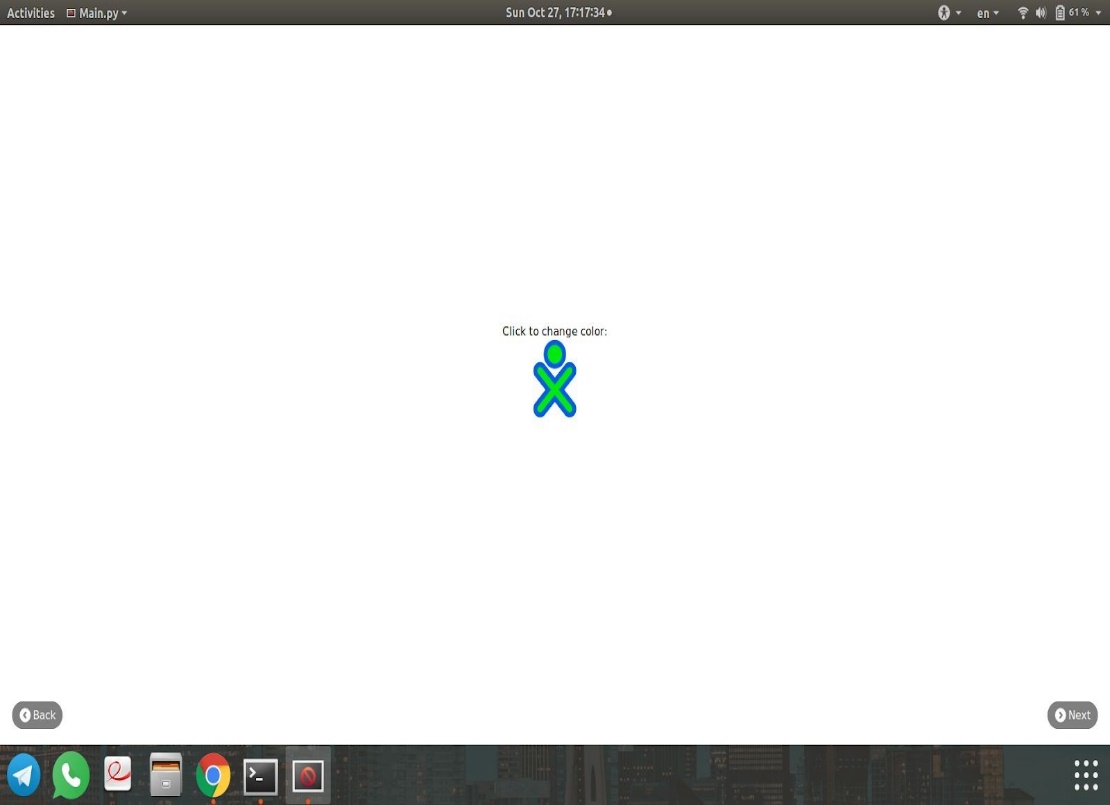
# 14. Sugar is an activity-focused, free/libre open-source software learning platform for children .

**SubTitle:** To install and study/use Sugar on ubuntu.

# 14.1 Theory:

To install Sugar on Ubuntu: sudo apt install sucrose





# Conclusions: 1. Learn to use Sugar.

# References:

1. <https://github.com/sugarlabs/sugar/blob/master/docs/ubuntu.md>

# 

# Assignment 15

**15 Open source programming language GO**

package main

import ( "fmt"

)

var toBeSorted [10]int = [10]int{1,3,2,4,8,6,7,2,3,0} func bubbleSort(input [10]int) {

n := 10

swapped := true for swapped { swapped = false

for i := 1; i < n; i++ {

if input[i-1] > input[i] { fmt.Println("Swapping")

input[i], input[i-1] = input[i-1], input[i] swapped = true

}

}

}

fmt.Println(input)

}

func main() { fmt.Println("Hello World") bubbleSort(toBeSorted)

}

This program is run by the following command – go run bubbleSort.go

**binarySearch.go** package main import "fmt"

func binarySearch(needle int, haystack []int) bool { low := 0

high := len(haystack) - 1

for low <= high{

median := (low + high) / 2

if haystack[median] < needle { low = median + 1

}else{

high = median - 1

}

}

if low == len(haystack) || haystack[low] != needle { return false

}

return true

}

func main(){

items := []int{1,2, 9, 20, 31, 45, 63, 70, 100}

fmt.Println(binarySearch(63, items))

}

This program is run by the command: go run binarySearch.go

# fibo.go

package main import "fmt" func main(){ var n int t1:=0

t2:=1

nextTerm:=0

fmt.Print("Enter the number of terms : ") fmt.Scan(&n)

fmt.Print("Fibonacci Series :") for i:=1;i<=n;i++ {

if(i==1){ fmt.Print(" ",t1) continue

}

if(i==2){ fmt.Print(" ",t2) continue

}

nextTerm = t1 + t2 t1=t2 t2=nextTerm

fmt.Print(" ",nextTerm)

}

}

This program is run by the command: go run fibo.go

# Features in Go:

Go is essentially an updated version of C designed to be more accessible to developers and to address some modern computing priorities (concurrency and network systems architecture - although the solution to the latter is mostly addressed through the concurrency solution). For their improvements to C, the designers chose

* Slightly cleaner syntax with some enforced standards
* Garbage collection
* Slightly safer pointers (no pointer arithmetic)
* Concurrency primitives
* Static compilation by default (no dependencies)
* A simple object syntax with no inheritance

# What they kept from C:

* Fast compilation as a top priority
* Syntax which makes abstraction hard.

Go’s best use case is still systems programming. Adding GC and making the syntax more accessible increases the pool of developers you can use and will even make C developers more productive if they can get over the GC thing. Within its sweet spot, it can be used productively to create high performance code more quickly and with fewer bugs than would be the case with C.

The opinionated priorities of the language work well in Google which has a large ecosystem enforcing the opinions and mitigating the trade-offs that any opinionated design brings with it. Companies which are not Google may have some issues.

But even procedural programmers can find the constraints difficult. Go developers often try to work around the extensibility limits with the very unsafe empty interface, which puts them right back into the fragile world of C type casting.

Abstraction is core to both FP and OOP. Go’s constraints make those abstractions very hard and the designers don’t want you to use the language that way. You can do object-based design and some use of functions rather than procedures but Go’s design does not align with those paradigm

# Go is neither a functional nor an object-oriented language. Rather, it is a procedural language similar to C or even Perl (before it got an object system).

**Tools:**

The main Go distribution includes tools for building, testing, and analyzing code:

* go build, which builds Go binaries using only information in the source files themselves, no separate makefiles
* go test, for unit testing and microbenchmarks
* go fmt, for formatting code
* go get, for retrieving and installing remote packages
* go vet, a static analyzer looking for potential errors in code
* go run, a shortcut for building and executing code
* godoc, for displaying documentation or serving it via HTTP
* gorename, for renaming variables, functions, and so on in a type-safe way
* go generate, a standard way to invoke code generators

# Applications:

Some notable open source applications written in Go include:

* [Caddy](https://en.wikipedia.org/wiki/Caddy_(web_server)), an open source HTTP/2 web server with automatic HTTPS capability.
* [CockroachDB](https://en.wikipedia.org/wiki/CockroachDB), an open source, survivable, strongly consistent, scale-out SQL database.
* [Docker](https://en.wikipedia.org/wiki/Docker_(software)), a set of tools for deploying [Linux](https://en.wikipedia.org/wiki/Linux) containers
* [Ethereum](https://en.wikipedia.org/wiki/Ethereum), The go-ethereum implementation of the Ethereum Virtual Machine blockchain for the Ether cryptocurrency
* [Hugo](https://en.wikipedia.org/wiki/Hugo_(software)), a static site generator
* [InfluxDB](https://en.wikipedia.org/wiki/InfluxDB), an open source database specifically to handle time series data with high availability and high performance requirements.
* [InterPlanetary File System](https://en.wikipedia.org/wiki/InterPlanetary_File_System), a content-addressable, peer-to-peer hypermedia protocol.
* [Juju](https://en.wikipedia.org/wiki/Juju_(software)), a service orchestration tool by [Canonical](https://en.wikipedia.org/wiki/Canonical_Ltd), packagers of [Ubuntu](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)) Linux
* [Kubernetes](https://en.wikipedia.org/wiki/Kubernetes) container management system
* [Lightning Network](https://en.wikipedia.org/wiki/Lightning_Network), a [bitcoin](https://en.wikipedia.org/wiki/Bitcoin) network that allows for fast Bitcoin transactions and scalability.
* [Mattermost](https://en.wikipedia.org/wiki/Mattermost), a teamchat system
* [OpenShift](https://en.wikipedia.org/wiki/OpenShift), a cloud computing platform as a service by [Red Hat](https://en.wikipedia.org/wiki/Red_Hat)
* [Snappy](https://en.wikipedia.org/wiki/Snappy_(package_manager)), a package manager for [Ubuntu Touch](https://en.wikipedia.org/wiki/Ubuntu_Touch) developed by Canonical.
* [Syncthing](https://en.wikipedia.org/wiki/Syncthing), an open-source file synchronization client/server application
* [Terraform](https://en.wikipedia.org/wiki/Terraform_(software)), an open-source, multiple [cloud](https://en.wikipedia.org/wiki/Cloud_computing) infrastructure provisioning tool from [HashiCorp](https://en.wikipedia.org/wiki/HashiCorp).