BCA-5053

Open Source Software

Unit-1

Introduction- Introduction to open sources, need of open sources, advantages of open sources and application of open sources.

Unit-2

Open Source Operating Systems: LINUX- Introduction, general overview, kernel mode and user mode, process, advanced concepts, scheduling, personalities, cloning and signals.

Unit-3

Open Source Database: MySQL- Introduction - setting up account-starting, terminating and

writing your own SQI programs, record selection technology, working with strings - date and time,sorting query results.

Unit-4

Open Source Programming Languages: PHP- Introduction - programming in web environment,

variables, constants, datatypes, operators, statements, functions, arrays and OOP - string

manipulation and regular expression.

Perl: Perl backgrounder, Perl overview, Perl parsing rules, variables and data -statements and

control structures, subroutines, packages, and modules- working with files and data manipulation.

Unit-I

Open Source Software
Open source is a term that originally referred to open source software (OSS). Open source software is code that is designed to be publicly accessible—anyone can see, modify, and distribute the code as they see fit.

Open source software is developed in a decentralized and collaborative way, relying on peer review and community production. Open source software is often cheaper, more flexible, and has more longevity than its proprietary peers because it is developed by communities rather than a single author or company.

Open source has become a movement and a way of working that reaches beyond software production. The open source movement uses the values and decentralized production model

of open source software to find new ways to solve problems in their communities and industries.

History of OSS

The idea of making source code freely available originated in 1983 from an ideological movement informally founded by Richard Stallman, a programmer at MIT. Stallman believed that software should be accessible to programmers so they could modify it as they wished, with the goal of understanding it, learning about it, and improving it. Stallman began releasing free code under his own license, called the GNU Public License. This new approach and ideology surrounding software creation took hold and eventually led to the formation of the Open Source Initiative in 1998.

What is the Open Source Initiative?

The Open Source Initiative (OSI) was created to promote and protect open source software and communities. In short, the OSI acts as a central informational and governing repository of open source software. It provides rules and guidelines for how to use and interact with OSS, as well as providing code licensing information, support, definitions, and general community collaboration to help make the use and treatment of open source understandable and ethical.

What does Open-source mean?

The term **Open-source** is closely related to Open-source software (OSS). Open-source software is a type of computer software that is released under a license, but the source code is made available to all the users. The copyright holders of such software allow the users to use it and do some valuable modifications in its source code to add some new features, to improve the existing features, and to fix bugs if there are any. Because of this reason only Open-source software is mostly developed collaboratively.

Some famous examples of Open-source products are:
- Operating systems –
  Android, Ubuntu, Linux
- Internet browsers –
  Mozilla Firefox, Chromium
- Integrated Development Environment (IDEs) –
  Vs code (Visual Studio Code), Android Studio, PyCharm, Xcode

Open-source community and Contributions:

The open-source community is a worldwide community of programmers and software developers who are continuously working on various open-source projects to make our lives better. This community is self-governing and self-organizing, there are no executives to take the decisions solely. This community plays a very crucial role in the sustainability of various open-source organizations.

The contributions made in any open-source project which improves its usability are called open-source contributions. These contributions can be of any form not only some software codes like we can work on improving its documentation, improving user interface and design, organize meetups, or find new collaborators.

Benefits of Open-source contributions:

- We code for real-world open-source projects.
- It refines our existing knowledge of programming and also helps us to learn new skills.
- Many open-source projects offer mentorship programs to guide and help us through our first few contributions.
- After making any open-source contribution, we get immediate feedback regarding our developmental work.
- While doing open-source contributions, we interact with like-minded developers from all over the world and build connections along the way.
- As we get closer to the open-source community, we get to know much more about our field of interest and other related fields.
- The most important aspect of open-source contributions is it may fetch us a job in our field of interest.

Hence, a large number of students are heading towards open-source contributions because these days quality open-source contributions in some good projects are seen as an alternative to the good internships with developer's profile. The reason is we get the same kind of exposure to work and learn the required skills for software development like any good internships at Google or Microsoft.

Why do people prefer using open source software?

People prefer open source software to proprietary software for a number of reasons, including:

Control. Many people prefer open source software because they have more control over that kind of software. They can examine the code to make sure it's not doing anything they don't want it to do, and they can change parts of it they don't like. Users who aren't programmers also benefit from open source software, because they can use this software for any purpose they wish—not merely the way someone else thinks they should.

Training. Other people like open source software because it helps them become better programmers. Because open source code is publicly accessible, students can easily study it as they learn to make better software. Students can also share their work with others, inviting comment and critique, as they develop their skills. When people discover mistakes in programs' source code, they can share those mistakes with others to help them avoid making those same mistakes themselves.

Security. Some people prefer open source software because they consider it more secure and stable than proprietary software. Because anyone can view and modify open source software, someone might spot and correct errors or omissions that a program's original authors might have missed. And because so many programmers can work on a piece of open source software without asking for permission from original authors, they can fix, update, and upgrade open source software more quickly than they can proprietary software.

Stability. Many users prefer open source software to proprietary software for important, long-term projects. Because programmers publicly distribute the source code for open source software, users relying on that software for critical tasks can be sure their tools won't disappear or fall into disrepair if their original creators stop working on them. Additionally, open source software tends to both incorporate and operate according to open standards.

Community. Open source software often inspires a community of users and developers to form around it. That's not unique to open source; many popular applications are the subject of meetups and user groups. But in the case of open source, the community isn't just a fan base that buys in (emotionally or financially) to an elite user group; it's the people who produce, test, use, promote, and ultimately affect the software they love.

Doesn't "open source" just mean something is free of charge?

No. This is a common misconception about what "open source" implies, and the concept's implications are not only economic.

Open source software programmers can charge money for the open source software they create or to which they contribute. But in some cases, because an open source license might require them to release their source code when they sell software to others, some programmers find that charging users money for *software services and support* (rather than for the software itself) is more lucrative. This way, their software remains free of charge, and they make money helping others install, use, and troubleshoot it.

While some open source software may be free of charge, skill in programming and troubleshooting open source software can be quite valuable. Many employers specifically seek to hire programmers with experience working on open source software.

Advantages of Open Source Software

Open source software can have a major impact on your entire organization. There are several advantages of using open source software. The following are a list of the advantages of opting for open source software.

1. **Lesser hardware costs**

   Since Linux and open source solutions are easily portable and compressed, it takes lesser hardware power to carry out the same tasks when compared to the hardware power it takes on servers, such as, Solaris, Windows or workstations. With this less hardware power advantage, you can even use cheaper or older hardware and still get the desired results.

2. **High-quality software**

   Open source software is mostly high-quality software. When you use the open source software, the source code is available. Most open source software are well-designed. Open source software can also be efficiently used in coding. These reasons make open source software an ideal choice for organizations.

3. **No vendor lock-in**

   IT managers in organizations face constant frustration when dealing with vendor lock-ins'. Lack of portability, expensive license fees and inability to customize software are

some of the other disadvantages. Using open source software gives you more freedom and you can effectively address all these disadvantages.

4. **Integrated management**

By using open source software, you can benefit from integrated management. Open source software uses technologies, such as, common information model (CIM) and web based enterprise management (WBEM). These high-end technologies enable you to integrate and combine server, application, service and workstation management. This integration would result in efficient administration.

5. **Simple license management**

When you use open source software, you would no longer need to worry about licenses. Open source software enables you to install it several times and also use it from any location. You will be free from monitoring, tracking or counting license compliance.

6. **Lower software costs**

Using open source software can help you minimize your expenses. You can save on licensing fees and maintenance fees. The only expenses that you would encounter would be expenditure for documentation, media and support.

7. **Abundant support**

You will get ample support when you use open source software. Open source support is mostly freely available and can be easily accessed through online communities. There are also many software companies that provide free online help and also varied levels of paid support. Most organization who create open source software solutions also provide maintenance and support.

8. **Scaling and consolidating**

Linux and open source software can be easily scaled. With varied options for clustering, load balancing and open source applications, such as email and database, you can enable your organization to either scale up and achieve higher growth or consolidate and achieve more with less.

Open-Source Software Criteria

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost, preferably downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of the Author's Source Code

The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination against Persons or Groups

The license must not discriminate against any person or group of persons.

6. No Discrimination against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business.

7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Restrict Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. License Must Be Technology-Neutral

No provision of the license may be predicated on any individual technology or style of interface.

Advantages

While its lack of cost is a key advantage, OSS has several additional benefits:

- Its quality can be easily and greatly improved when its source code is passed around, tested, and fixed.
- It offers a valuable learning opportunity for programmers. They can apply skills to the most popular programs available today.
- It can be more secure than proprietary software because bugs are identified and fixed quickly.
- Since it is in the public domain, and constantly subject to updates, there is little chance it can become unavailable or quickly out moded—an important plus for long-term projects.

Application of open source

Open-source technologies helped establish much of the internet. Furthermore, many of the programs in use every day are based on open-source technologies. Cases in point: Android OS and Apple's OS X are based on the kernel and Unix/BSD open-source technologies, respectively.

Other popular open-source software is:

- Mozilla's Firefox web browser
- Thunderbird email client
- PHP scripting language
- Python programming language
- Apache HTTP web server

Developers

OSS projects are collaboration opportunities that improve skills and build connections in the field. Areas that developers can work on include:

Communication Tools

Email, real-time messaging, forums, and wikis help developers to find solutions or bounce ideas off each other.

Distributed Revision Control Systems

When multiple developers in different geographical locations modify data and files, these systems manage the different versions and updates.

Bug Trackers and Task Lists

These features allow large-scale projects to monitor issues and keep track of their fixes.

Testing and Debugging Tools

These features automate testing during system integration and debug other programs.

The Bottom Line

Open-source software is an alternative to proprietary software. Participating in an OSS project can be a pathway to building a career in software development, allowing programmers to sharpen their skills by working on the biggest software programs in the world. Facebook, Google, and LinkedIn all release OSS, so developers can share knowledge, innovate solutions, and contribute to stable, functional products.