

Experiment no:1

Aim: Introduction to Open source technologies.

Theory:

Introduction:

1} What is Open source?

- The term "open source" refers to something people can modify and share because its design is publicly accessible.
- The term originated in the context of software development to designate a specific approach to creating computer programs.
- Today, however, "open source" designates a broader set of values—what we call "the open source way." Open source projects, products, or initiatives embrace and celebrate principles of open exchange, collaborative participation, rapid prototyping, transparency, meritocracy, and community-oriented development.

2} What is open source software?

- Open source software is software with source code that anyone can inspect, modify, and enhance.
- "Source code" is the part of software that most computer users don't ever see; it's the code computer programmers can manipulate to change how a piece of software—a "program" or "application"—works. Programmers who have access to a computer program's source code can improve that program by adding features to it or fixing parts that don't always work correctly.

History:

In the 1950s and early 1960s the early software was free by definition, due to the academic nature of software development, as well as compatibility and porting requirements and the lack of a separate software business model; revenue was generated with hardware, primarily. It was also a concern that closed source software would allow for backdoors used for clandestine purposes, as it was virtually impossible to see what a software was doing while executed. With the recent leaks on the doings of security agencies and large corporations, it is not a stretch to claim that today this is an even more valid concern...

In the late 1960s the software industry was becoming a real business. Software was getting increasingly decoupled from the hardware business, requiring a separate business model. Lawyers drafted restrictive licenses to enable this. ARPANET researchers used RfCs to develop telecom protocols. This collaboration eventually led to the birth of the Internet in 1969. Just as well; a decade later and we would have a plethora of competing commercial proprietary internets.

In the 1970s AT&T released early versions of UNIX. The software was free of charge, but users were not allowed to redistribute or modify it. In the late 1970s and early 1980s charging for software licenses became a dominant business model for software companies and computer vendors. Legal restrictions were imposed through copyrights, trademarks and other contracts.

License enforcement via legal actions began. Software piracy was born.

In the 1980s software was shared via BBS systems. Software written in BASIC and other interpreted languages could only be distributed as source code. A lot of freeware became available. Software moulding became popular and Usenet provided a good collaboration channel for programmers/moulders. Richard Stallman started the GNU Project and founded the Free Software Foundation. The first companies making free software as their primary business emerged.

In the early 1990s the free software community received the first complete free operating system with Linus Torvalds's kernel combined to GNU operating system. Debian, founded by Ian Murdock in 1993, committed to the GNU and FSF principles of free software. Linux adoption by businesses and governments began in the late 1990s. Website-based companies emerged and made extensive use of free web servers, especially the Apache HTTP Server. The LAMP (Linux, Apache, MySQL, PHP) stack gained popularity over expensive proprietary solutions.

Freeware Summit organized by Tim O'Reilly brought together the leaders of free and open source projects. A developer vote decided on Open Source as a new term over Source ware. The Open Source Initiative was formed much to the disdain of Richard Stallman and the FSF, who felt that OSI was selling out on some core values. FSF and OSI remain the main schools of the movement today and so remains their philosophical discord. Fortunately, they do agree on many practical matters and are able to work together for the common cause.

In the early 2000s big software corporations began to see free software as a threat to their core business. Microsoft's Steve Ballmer called Linux a cancer, referring to its copyleft license. In 2003 SCO made claims that Unix IPR had been copied into Linux kernel and decided to jump straight into the deep end of the pool by suing IBM. They did not lack balls, but, as it turned out, they did mostly lack the ownership of the IPR they claimed had been violated. The case is still technically ongoing, even though SCO filed for bankruptcy in 2007, after multiple defeats in court. SCO allegedly received funding from Microsoft.

In the late 2000s the availability and popularity of hosted distributed revision control services, such as GitHub, has reduced barriers on participating in free software projects.

In the recent years the free software movement has seen some worrying corporate acquisitions, such as Sun Microsystems purchasing MySQL and Oracle purchasing Sun Microsystems.

Meanwhile open source project participation is growing exponentially. Big players, such as telecom operators, are adapting (or seriously considering) free software alternatives. Many new school IT houses and enlightened in-house developers are fighting the good fight, paving way for this tectonic shift. New business models are emerging; creating a popular web service and open sourcing it all would have been, in the recent past, considered financially insane. Today it is becoming a norm.

Many government institutions have adopted free software and open innovation as the fundamental building blocks in building a successful information society. For instance, the Helsinki Region Infoshare program has since 2010 promoted open data publication in the metropolitan area of Finland's capital, opening up more than a thousand datasets online, allowing for free development of applications to support new business.

Need:

- At times, it was not feasible to find, assemble and manage a team of IT professionals for a task. Hence the need was felt to open up the source code to entire world so that interested developers would work on existing code to generate a more secure and robust code. In short

instead of using a workforce of a few thousand professional, anyone with a will to develop will go ahead with it.

- Small enterprises, shops, start-ups and many such low budget institutions cannot afford to pay for closed source, proprietary software to run their day to day business. Thus, an alternative was (and is) required for them.
- Security Issue: It was felt by the coding community that the Hackers used to hack into software for the major reason that it used to be paid and they could make money by finding a bug and selling a pirated version of it. If the debugging team was expanded by making the code open, then at least one from the entire human race (being a good Samaritan) would point out the bug and stop its misuse.
-

Advantages:

1. It's generally free – it has been estimated that open source software collectively saves businesses \$60 billion a year. These days for virtually every paid for proprietary software system you will find an open source version.

2. It's continually evolving in real time as developers add to it and modify it, which means it can be better quality and more secure and less prone to bugs than proprietary systems, because it has so many users poring over it and weeding out problems.

3. Using open source software also means you are not locked in to using a particular vendor's system that only work with their other systems.

4. You can modify and adapt open source software for your own business requirements, something that is not possible with proprietary systems.

5. Highly reliable.

There are two main reasons why open source software are reliable. First of all, they're developed chiefly by skillful and talented experts who do their best to create high-quality programs. Second, they're worked on by tens or hundreds of people, which means there are numerous eyes that can monitor for the presence of bugs and many pairs of hands that can fix these defects within the shortest amount of time. Both of these factors lead to products that have excellent quality and helpful features and perform well most (if not all) of the time.

6. Created by skillful and talented people.

Large and well-established software companies have the financial capability to hire the best talent in the business to create their products. Because of this, many people opt to buy computer programs from these firms because they think they'll get great value for their money by doing so.

The fact that big companies hire experienced, fully trained and highly qualified people is true. But what consumers have to know is this: not all software developers care deeply about money. Sure, most of them do get a job to have a steady income and be able to financially support themselves and their family. However, a lot of these experts don't just base their worth on the salary they earn; rather, they strive to build a program that will earn the admiration of their peers and hopefully make a difference in the world.

This is why many software developers turn to open source products as an outlet for their ideas and creativity. By doing so, they won't be confined by the rigid rules of the corporate world, and they'll have the freedom to experiment and come up with high-quality programs. These, in turn, benefit consumers since they'll have access to world-class and state-of-the-art software without

having to pay too much.

Disadvantages:

1. Because there is no requirement to create a commercial product that will sell and generate money, open source software can tend to evolve more in line with developers' wishes than the needs of the end user.
2. For the same reason, they can be less "user-friendly" and not as easy to use because less attention is paid to developing the user interface.
3. There may also be less support available for when things go wrong – open source software tends to rely on its community of users to respond to and fix problems.
4. Although the open source software itself is mostly free, there may still be some indirect costs involved, such as paying for external support.

5. Vulnerable to malicious users.

Many people have access to the source code of open source software, but not all of them have good intentions. While a lot of people utilize their access to spot defects and make improvements to the program, others use this privilege to exploit the product's vulnerabilities and create bugs that can infect hardware, steal identities or just annoy other users.

6. Might not be as user-friendly as commercial versions.

This is not true for all open source software since many of them (such as LibreOffice, Mozilla Firefox and the Android operating system) are incredibly easy to use. However, there are several programs which are created mainly to cater to the developer's wishes and bring his ideas to life. As a result, not much attention is given to the software's user interface, making it difficult to use especially for those who aren't really tech-savvy.

Types of Open source softwares:

Examples of Open Source Software		
TYPE OF APPLICATION	OSS EXAMPLES	DESCRIPTION
Office Automation (word processing, spreadsheet and presentation software)	AbiWord	<ul style="list-style-type: none">• Word-processing program• Similar to predominant proprietary word-processing programs• Suitable for a wide variety of word processing tasks
	OpenOffice	<ul style="list-style-type: none">• Office suite• Suitable for individuals and businesses• Includes a word processor (compatible with predominant proprietary word-processing programs), spreadsheet (compatible with predominant proprietary spreadsheet programs) and presentation system (compatible with predominant proprietary presentation systems)

	KOffice	<ul style="list-style-type: none"> • Integrated office suite • Intended for the KDesktop Environment • Includes a word processor, spreadsheet application, and presentation program
Web Design	Nvu	<ul style="list-style-type: none"> • Intended for those with no technical or programming expertise • Enables creation of web pages and management of websites
	GIMPShop	<ul style="list-style-type: none"> • Image editor similar to Adobe Photoshop
	Bluefish	<ul style="list-style-type: none"> • Editor targeted towards programmers and web designers • Supports many programming and mark-up languages • Focused on editing dynamic and interactive websites

Communications	Pidgin	<ul style="list-style-type: none"> • Free Instant Messaging (IM) client • Allows use of all IM accounts at once.
	Thunderbird	<ul style="list-style-type: none"> • Cross-platform e-mail and news client
	PhpBB	<ul style="list-style-type: none"> • Internet forum package written in computer scripting language PHP
E-Commerce	OsCommerce	<ul style="list-style-type: none"> • E-commerce and online store-management application • Offers a wide range of features that allows online stores • Can be used on any web server that has PHP web scripting language and the MySQL database
	VirtueMart	<ul style="list-style-type: none"> • E-commerce solution intended for use with the content-management system Joomla or Mambo • Written in PHP • Made for easy use in a PHP/MySQL environment
	Zen Cart	<ul style="list-style-type: none"> • Free, shopping-cart system • Features multiple customer modes, unlimited category depth, multiple sales and discounts, multiple display modes, multiple ad banner controller, multiple payment options, etc.
Content Management Systems	Drupal	<ul style="list-style-type: none"> • Free, modular, content-management framework, content-management system and blogging engine • Written in PHP • Allows an individual or a community of users to easily publish, manage and organize a wide variety of content on a website

	Joomla	<ul style="list-style-type: none"> • Free open source content-management system meant for publishing content on the Web and intranets using the MySQL database • Written in PHP • Includes features such as page caching to improve performance, RSS feeds, printable versions of pages, news flashes, blogs, polls, website searching, and language internationalization
	PHP-Nuke	<ul style="list-style-type: none"> • Free, web-based automated news publishing and content-management system • Based on PHP and MySQL • Fully controlled using a web-based user interface
Operating Systems (all Linux distributions)	Ubuntu	<ul style="list-style-type: none"> • Largest community maintained Linux OS – enables users to draw upon a wide network for support
	Fedora	<ul style="list-style-type: none"> • Open source Fedora is a general purpose Linux operating system, developed by the community-supported Fedora Project and sponsored by Red Hat (a company committed to open source software, and a major Linux distribution vendor).

Conclusion:

Thus, the concept of Open source software, its history, advantages and disadvantages were studied in detail .Various misconceptions and myths were shattered and the concept and idea of open source as a new technological advancement was conceived.