

Open Source Tools(cs-19)

PREPARED BY: MAKWANA SHAILESH

KAMANI SCIENCE & PRATAPRAI ARTS
COLLEGE, AMARELI (BCA DEPARTMENT)

Open Source Software

- ▶ **Open source software (OSS)** is software that is distributed with its source code, making it available for use, modification, and distribution with its original rights.
- ▶ Source code is the part of software that most computer users don't ever see; it's the code computer programmers manipulate to control how a program or application behaves.

Open Source Software

- ▶ Programmers who have access to source code can change a program by adding to it, changing it, or fixing parts of it that aren't working properly. OSS typically includes a license that allows programmers to modify the software to best fit their needs and control how the software can be distributed.

Definition (OSS)

- ▶ **Open-source software (OSS)** is software that is released with a license that allows anyone to view, modify, and distribute the source code. This model contrasts with proprietary software, where the source code is usually hidden, and users are limited to using the software as provided by the vendor.
- ▶ **GNU General Public License** is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

The principles of open source

Redistribution

- the OSS and its successive modifications can be redistributed.

Source code

- the source code must be included in the program, in order to promote its distribution.

The principles of open source

Modification

- everyone should have access to the source code so that they can easily modify it.

Integrity

- the software can be used, modified or redistributed under a free license, provided that the license remains the same.

The principles of open source

No discrimination

- whether against persons, groups or areas of activity.

Non-specificity

- the software license should not be product specific.

The principles of open source

No restrictions

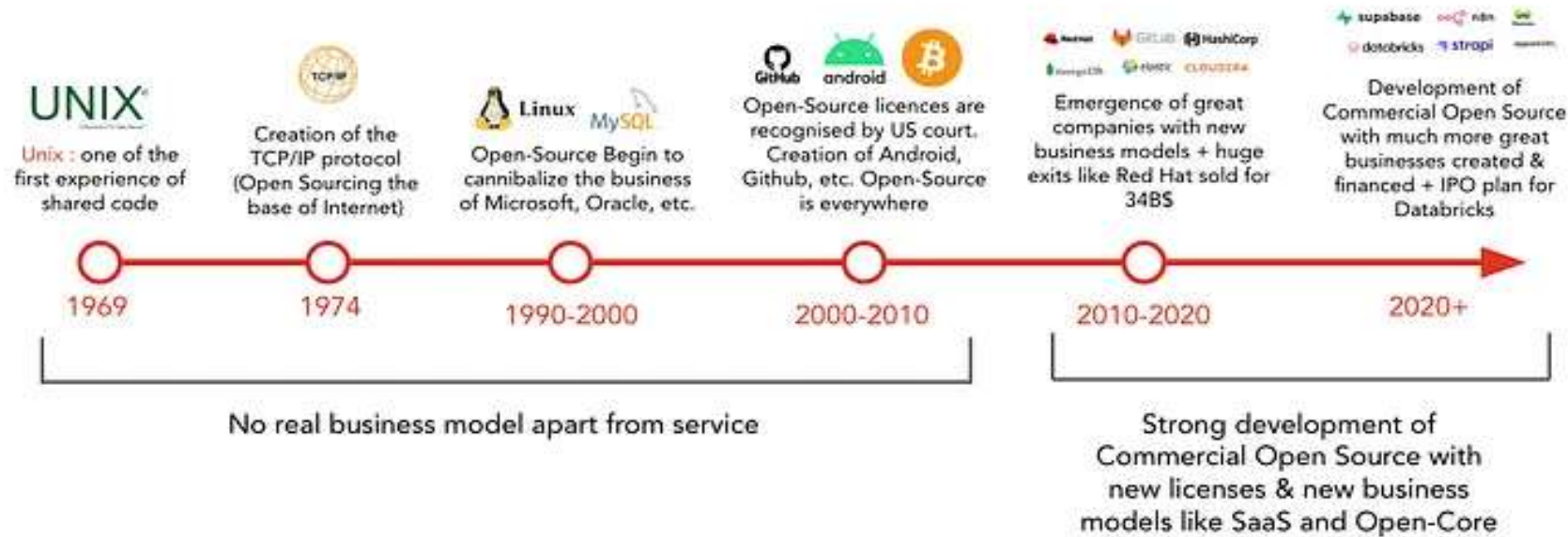
- the program can be used for any purpose.

Neutrality

- the code must be designed in a neutral way, in terms of technology and design.

History and Evolution of OSS

The evolution of Open-Source



History and Evolution of OSS

- ▶ **Early Beginnings (1950s-1970s)**
- ▶ **Sharing Culture** : In the early days of computing, software was often shared freely among academics and researchers. This was primarily because software was seen as an extension of hardware, not a standalone product.
- ▶ **IBM and Source Code**: Companies like IBM provided source code with their hardware allowing users to modify and improve the software. The sharing of source code was standard practice

History and Evolution of OSS

- ▶ **Unix and Bell Labs:** In 1969, AT&T's Bell Labs developed Unix, an operating system that would later play a crucial role in the open-source movement. Unix's source code was shared with universities, leading to the development of various Unix variants, many of which were open.

History and Evolution of OSS

- ▶ **The Rise of Proprietary Software (1970s-1980s)**
- ▶ **Shift to Proprietary Models:** As the software industry grew, companies began to see the commercial potential of software. Microsoft, founded in 1975, was among the first to treat software as a proprietary product, keeping the source code closed.

History and Evolution of OSS

- ▶ **GNU Project:** In 1983, Richard Stallman, a programmer at MIT, launched the GNU Project to create a free Unix-like operating system. He introduced the concept of "free software" and later founded the Free Software Foundation (FSF) in 1985 to support the movement.
- ▶ **GNU General Public License (GPL):** In 1989, Stallman created the GPL, a revolutionary open-source license that ensured software licensed under it would remain free and open. This was the birth of the copyleft principle.

History and Evolution of OSS

- ▶ **GNU General Public License (GPL):** In 1989, Stallman created the GPL, a revolutionary open-source license that ensured software licensed under it would remain free and open. This was the birth of the copyleft principle.

History and Evolution of OSS

- ▶ The Birth of Open Source (1990s)
- ▶ **Linux Kernel:** In 1991, Linus Torvalds, a student in Finland, released the Linux kernel, a key component of a Unix-like operating system. When combined with GNU software, it formed a fully functional, free operating system: GNU/Linux. Linux became the flagship of the open-source movement.

History and Evolution of OSS

- ▶ **Apache Web Server:** In 1995, the Apache HTTP Server was released. It quickly became the most popular web server software, demonstrating the power of collaborative development.

History and Evolution of OSS

- ▶ **Mainstream Adoption (2000s)**
- ▶ **Corporate Adoption:** By the early 2000s, major corporations like IBM, Google, and Sun Microsystems began to embrace open source. IBM invested heavily in Linux, and Google built its entire infrastructure on open-source technologies.
- ▶ **Open-Source Applications:** Key software like the Apache web server, MySQL database, and the PHP programming language became the backbone of the web.

History and Evolution of OSS

- ▶ Open-source desktop applications like Mozilla Firefox, OpenOffice, and GIMP gained popularity.
- ▶ **Version Control and Collaboration:** Tools like CVS, Subversion, and later Git (created by Linus Torvalds in 2005) facilitated collaborative development, enabling developers around the world to contribute to open-source projects more effectively.

History and Evolution of OSS

- ▶ **The Cloud and Open Source (2010s)**
- ▶ **Cloud Computing:** The rise of cloud computing saw the proliferation of open-source software in data centers. OpenStack, Kubernetes, and Docker are examples of open-source projects that became essential in cloud environments.
- ▶ **GitHub:** Founded in 2008, GitHub became the largest platform for open-source collaboration, hosting millions of projects and becoming a central hub for developers.

History and Evolution of OSS

- ▶ **Open-Source in AI and Big Data:** Open-source frameworks like TensorFlow, Hadoop, and Apache Spark played significant roles in the advancement of artificial intelligence and big data analytics.

History and Evolution of OSS

- ▶ **Present Day (2020s and Beyond)**
- ▶ **Corporate Open Source:** Companies like Microsoft, once staunchly anti-open-source, have embraced it. Microsoft acquired GitHub in 2018 and now contributes to numerous open-source projects, including the development of the Linux kernel.
- ▶ **Open-Source Governance and Challenges:** As open-source projects scale, issues of governance, sustainability, and security have become more prominent. The community continues to address these challenges through new models of funding, governance structures, and community engagement.

Impact of Open Source

- ▶ **Innovation and Collaboration:** Open-source software has driven innovation, enabling rapid development and collaboration across the globe.
- ▶ **Accessibility:** By providing high-quality software for free, open-source has made technology more accessible to individuals, startups, and developing countries.
- ▶ **Business Model:** Open source has led to new business models, such as open-core, where companies offer free basic versions of software but charge for advanced features, support, or services.

Proprietary software vs open source software

COMPARISON OF OPEN-SOURCE AND PROPRIETARY SOFTWARE

	proprietary software	open-source software
licence free		
maintenance support	commercial support	community or commercial support, if available
source code access	no access	access, modification possible

Proprietary software vs open source software



Important Terms

- ▶ **GNU General Public License (GPL)**
- ▶ **Open Source Software(OSS)**
- ▶ **Source Code**
- ▶ **The Free Software Foundation (FSF)**
- ▶ **International Business Machines(IBM)**
- ▶ **Free and Open Source Software (FOSS)**

Open source licensing-Overview

- ▶ **An open-source license is a legal agreement that governs the use, modification, and distribution of open-source software. It grants users the freedom to study, change, and improve the software, with the source code being publicly accessible.**
- ▶ **While there are different types of open-source licenses, each with its own terms and conditions, they all promote the fundamental principle of open-source software that knowledge should be shared openly and freely.**

Types of open-source licenses

Permissive
Licenses

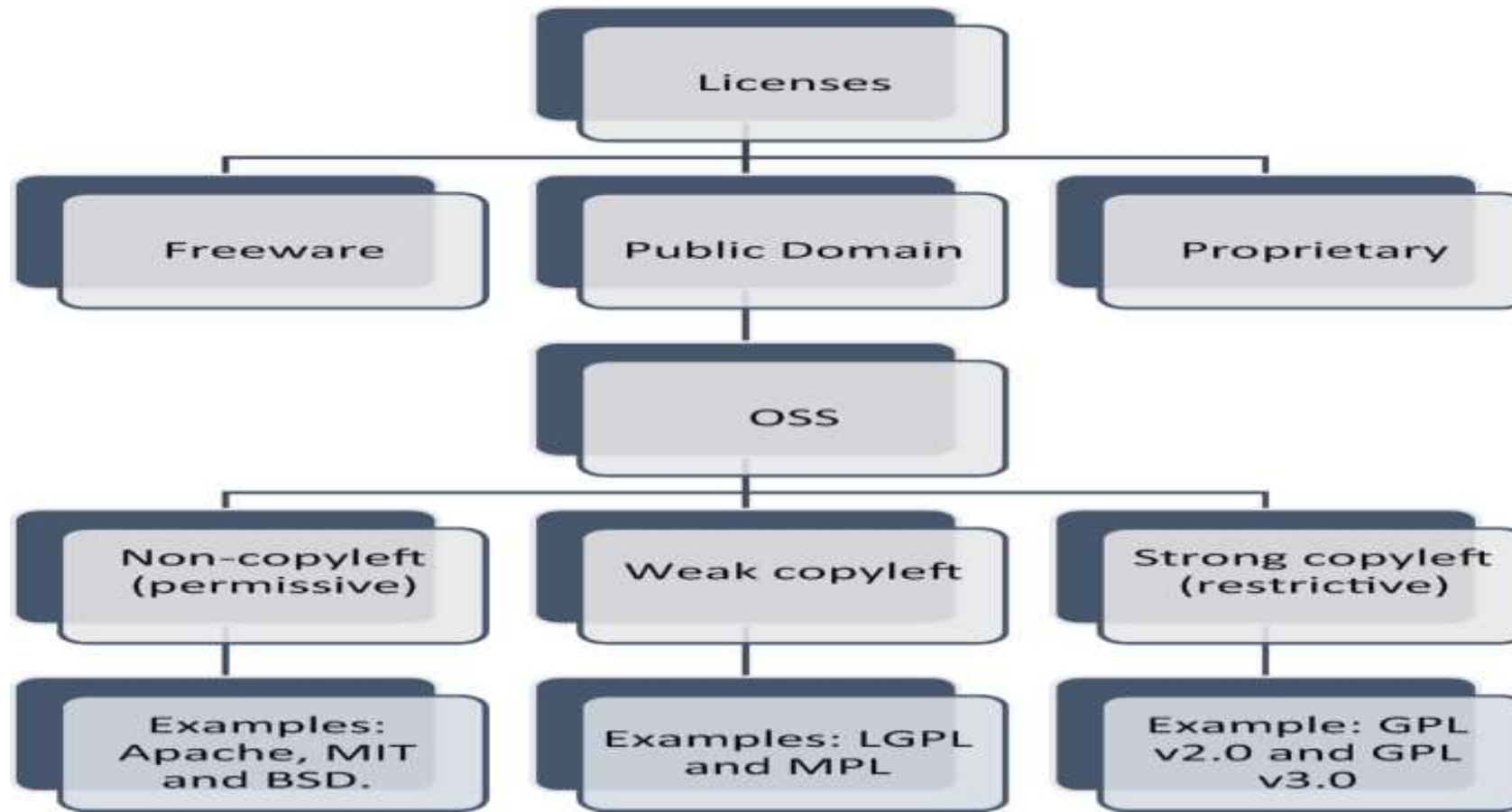
Copyleft
Licenses

Weak
Copyleft
Licenses

Public
Domain and
License-Free

Other
Specialized
License

Types of open-source licenses



Permissive Licenses

- ▶ **Permissive Licenses**
- ▶ Permissive licenses offer maximum freedom with minimal restrictions on how the software can be used, modified, and redistributed. They often allow for integration into proprietary software.

Permissive Licenses

- ▶ **MIT License**
- ▶ **Features:** Allows nearly unrestricted use, modification, and distribution, as long as the original copyright and license notices are included.
- ▶ **Notable Projects:** jQuery, Rails.
- ▶ **Apache License 2.0**

Permissive Licenses

- ▶ **Apache License 2.0**
- ▶ **Features:** Similar to MIT but includes provisions for patent rights and requires documentation of changes made to the code.
- ▶ **Notable Projects:** Apache HTTP Server, Kubernetes.

Permissive Licenses

- ▶ **BSD License**
- ▶ **Features:** Includes the 2-clause and 3-clause variants, allowing for broad usage and redistribution, with the primary requirement being attribution.
- ▶ **Notable Projects:** FreeBSD, OpenBSD.

Copyleft Licenses

- ▶ Copyleft licenses require that any derivative works of the software be distributed under the same license, ensuring that the software and its derivatives remain open-source.
- ▶ **GNU General Public License (GPL)**
- ▶ **Features:** Strong copyleft license that mandates derivatives and any software incorporating GPL-licensed code must also be licensed under the GPL.
- ▶ **Notable Projects:** Linux kernel, GNU software.

Copyleft Licenses

- ▶ **GNU Lesser General Public License (LGPL)**
- ▶ **Features:** Weaker copyleft that allows the software to be linked with proprietary programs but requires modifications to the LGPL-covered components to be open-sourced.
- ▶ **Notable Projects:** GNU C Library (glibc), FFmpeg.
- ▶ **Mozilla Public License (MPL)**

Weak Copyleft Licenses

- ▶ These licenses offer a balance between permissive and strong copyleft, allowing some flexibility while ensuring that changes to the licensed code itself remain open-source.
- ▶ **MPL (Mozilla Public License)**
- ▶ **Features:** Requires modifications to MPL-covered files to be open-source but permits other parts of the project to use different licenses.

Public Domain and License-Free

- ▶ These options place software in the public domain or waive all copyright claims, effectively allowing unrestricted use.
- ▶ **Unlicense**
 - **Features:** A public domain dedication that allows anyone to do anything with the software without restrictions.
- ▶ **Creative Commons Zero (CC0)**
- ▶ **Features:** Similar to the Unlicense, CC0 allows creators to waive all rights and place their work in the public domain.

Other Specialized Licenses

- ▶ Some licenses address specific concerns or use cases, such as patent rights or network interaction.
- ▶ **Eclipse Public License (EPL)**
- ▶ **Features:** A copyleft license used by the Eclipse Foundation, similar to LGPL but with a focus on patent issues.
- ▶ **GNU Affero General Public License (AGPL)**
- ▶ **Features:** A copyleft license that requires source code to be made available when the software is used over a network.(MongoDB)

GNU Affero General Public License

[Home](#) / [Licenses](#)

GNU Affero General Public License v3.0

GNU AGPLv3

Permissions of this strongest copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights. When a modified version is used to provide a service over a network, the complete source code of the modified version must be made available.

Permissions

- Commercial use
- Distribution
- Modification
- Patent use
- Private use

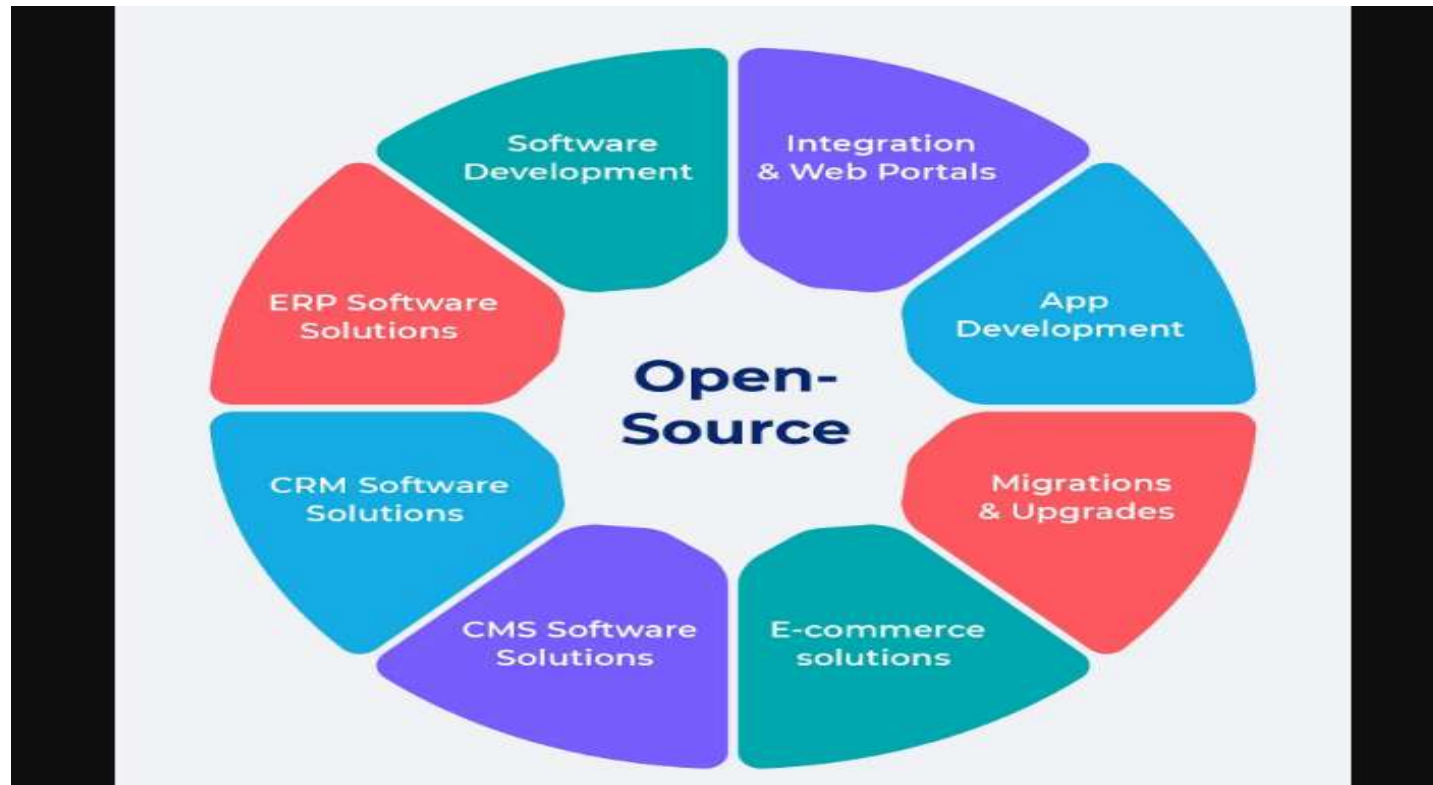
Conditions

- Disclose source
- License and copyright notice
- Network use is distribution
- Same license
- State changes

Limitations

- Liability
- Warranty

Application of open source software



Application

Frameworks	 ZEND FRAMEWORK   smarty  Symfony  CodeIgniter
Language	    python 
eCommerce	 Magento <small>Open Source eCommerce</small>  openCart  osCommerce <small>Open Source e-Commerce Solutions</small>  VirtueMart  zenCart <small>The art of e-commerce</small>
CMS	 WORDPRESS  Drupal  Joomla!  moodle  TYP03
Others	 SUGARCRM. <small>THE CLOUD IS OPEN</small>  OpenX  ubuntu  Asterisk  Solr <small>Apache</small>

Open Source License vs Commercial License

Aspect	Open Source License	Commercial License
Cost	Typically free of charge	Requires payment or subscription fees
Source Code Availability	The source code is freely available and can be modified	The source code may or may not be available for viewing/modification
Redistribution	Allows redistribution and modification	Restrictions may apply on redistribution and modification
Support	Community-driven support	Typically includes vendor-provided support and maintenance

Open Source License vs Commercial License

Updates	Community-driven updates and patches	Vendor-provided updates and patches
Customization	Users can customize the software to meet specific needs	Customization may be limited or require additional agreements
Warranty	Typically provided "as is" without warranty	May include warranties or guarantees
Licensing Terms	Governed by various open-source licenses (e.g., GPL, MIT, Apache)	Governed by proprietary license agreements

Important Terms

- ▶ GNU Lesser General Public License (LGPL)
- ▶ Mozilla Public License (MPL)
- ▶ Creative Commons Zero (CC0)
- ▶ GNU Affero General Public License (AGPL)
- ▶ Eclipse Public License (EPL)

Rights of Open-source licenses

- ▶ Open-source licenses govern the rights of both users and developers, outlining how software can be used, modified, and distributed.
- ▶ There are several open-source licenses, each with different terms. Here's a general overview of the rights for both users and developers under common open-source licenses

Rights of the Developer

- ▶ **Control over License Terms:** Developers can choose a license that fits their goals. Some prefer permissive licenses that allow others to use their code freely, while others may prefer licenses that require any modifications or derivative works to be open-sourced as well (copyleft).
- ▶ **Credit for the Original Work:** Most open-source licenses require that developers are credited for their original work, even if others modify or distribute it.

Rights of the Developer

- ▶ **Liability Protection:** Open-source licenses often disclaim warranties and liabilities, protecting developers from legal claims related to the use of their software.

Rights of the User

- ▶ **Right to Use:** Users can download, install, and run the software for any purpose (commercial or non-commercial).
- ▶ **Right to Modify:** Users can modify the source code to suit their needs. They can also combine the software with other projects.

Rights of the User

- ▶ **Right to Distribute:** Users can redistribute copies of the original or modified software. The conditions for redistribution vary depending on the license (e.g., under GPL, modified versions must also be open-source).
- ▶ **Right to Access Source Code:** Most open-source licenses require that the source code is made available to the user, allowing them to study, modify, or contribute to the code.

Responsibility of user and developer (OSL)

- ▶ In the context of open-source licenses, both developers and users have specific responsibilities to ensure that the rights granted under the license are respected. These responsibilities vary depending on the type of license (permissive vs. copyleft)

Responsibilities of the Developer:

- ▶ Developers are responsible for clearly stating the license under which their software is distributed. This includes making the license available alongside the source code, so users understand their rights and obligations.
- ▶ If the license requires attribution (e.g., MIT License), the developer must ensure their name, copyright notice, and any required legal disclaimers are included in the source code and accompanying documentation.

Responsibilities of the Developer:

- ▶ **Consistency with License Terms:** Developers must to the terms of any external libraries or software that they incorporate into their project. If they are using code licensed under the GPL, for instance, they cannot relicense that code under a permissive license without proper authorization.
- ▶ **No Warranty Disclaimer:** Open-source licenses generally include a no-warranty clause, but developers must ensure this is explicitly stated so users know they are using the software "as-is" without guarantees.

Responsibilities of the Developer:

- ▶ **Maintaining Compatibility with License:** Developers using a copyleft license like the GPL are responsible for ensuring that derivative works are also licensed under the same terms. This helps ensure that the software remains open-source and that any redistributed versions to the same principles.
- ▶ **Sharing Modifications:** In the case of copyleft licenses like the GPL, users who distribute modified versions of the software are required to make their changes available to others under the same license.

Responsibilities of the User:

This includes providing access to the source code for their modifications.

- ▶ **No Warranty Acceptance:** Users must accept that the software is provided without any warranty. This means they bear the risk of using the software, and the developer is not responsible for any potential issues or damages.

Fedora

- ▶ Fedora (or Flexible Extensible Digital Object Repository Architecture) is a digital asset.
- ▶ Fedora will feel significantly faster than Windows
- ▶ Fedora is developed by the Fedora Project and sponsored by Red Hat.

Fedora

- Note that the memory requirements do not mean that Fedora CoreOS will consume 2GB of RAM at all times. It means that to have a reasonably working experience, we expect system to have 2GB or RAM at the minimum. Once the system is booted, a lot of those 2GB are free for applications to use

Fedora

- ▶ Fedora Linux is a free, open-source operating system (OS) based on the Linux kernel.
- ▶ It was created by the Fedora Linux Project and sponsored by Red Hat Linux, a leading provider of open-source solutions.
- ▶ Fedora Core 1, the first version of Fedora, was released in November 2003.
- ▶ Fedora Linux is a free and open-source operating system based on the Linux kernel.

Fedora

- ▶ The Linux kernel is the part of the operating system that communicates with the computer's hardware and manages system resources such as memory and processing power.

Fedora Memory Requirements

- ▶ **Fedora Workstation (GUI/Desktop Environment)**
- ▶ **Minimum RAM: 2 GB**
- ▶ **Recommended RAM: 4 GB or more for smoother performance**

Fedora Memory Requirements

- ▶ **Fedora Server (No GUI)**
- ▶ Minimum RAM: 512 MB
- ▶ Recommended RAM: 1 GB or more for better performance with basic services

Fedora Memory Requirements

- ▶ **Fedora Silverblue (Immutable Desktop OS)**
- ▶ **Minimum RAM: 4 GB**
- ▶ **Recommended RAM: 8 GB or more, especially if using Flatpak applications extensively**

Fedora Memory Requirements

- ▶ **Fedora CoreOS (Container-focused variant)**
- ▶ **Minimum RAM: 512 MB**
- ▶ **Recommended RAM: 1 GB or more, depending on the workload**

Fedora was launched

- ▶ Fedora was launched in 2003 as the successor to Red Hat Linux after Red Hat shifted its focus to the enterprise market with Red Hat Enterprise Linux (RHEL). Fedora emerged as a free, community-driven distribution, quickly establishing itself as a leading platform for those interested in the latest software and features

Mission

- ▶ The Fedora Project aims to advance the development of free and open-source software through a collaborative community. Its mission is to create a robust, flexible, and innovative operating system that provides a platform for users and developers to test and use the latest advancements in open-source technology.

Fedora use

- ▶ You can use it for home use like browsing the Web, watching streaming video, editing photos, and playing games. You can use it for work creating documents, crunching numbers in spreadsheets, or programming. All of the software provided with Fedora is open source and free to download and use.

Editions

- ▶ **Fedora Workstation:** This edition is intended for personal and professional use on desktop computers and laptop computers.
- ▶ **Fedora Server:** This edition is intended for use on servers and includes a variety of server management and administration tools and features. It includes the Fedora Server Management Tool, which simplifies server configuration and management, as well as a variety of tools for deploying and managing servers at scale.

Fedora Workstation



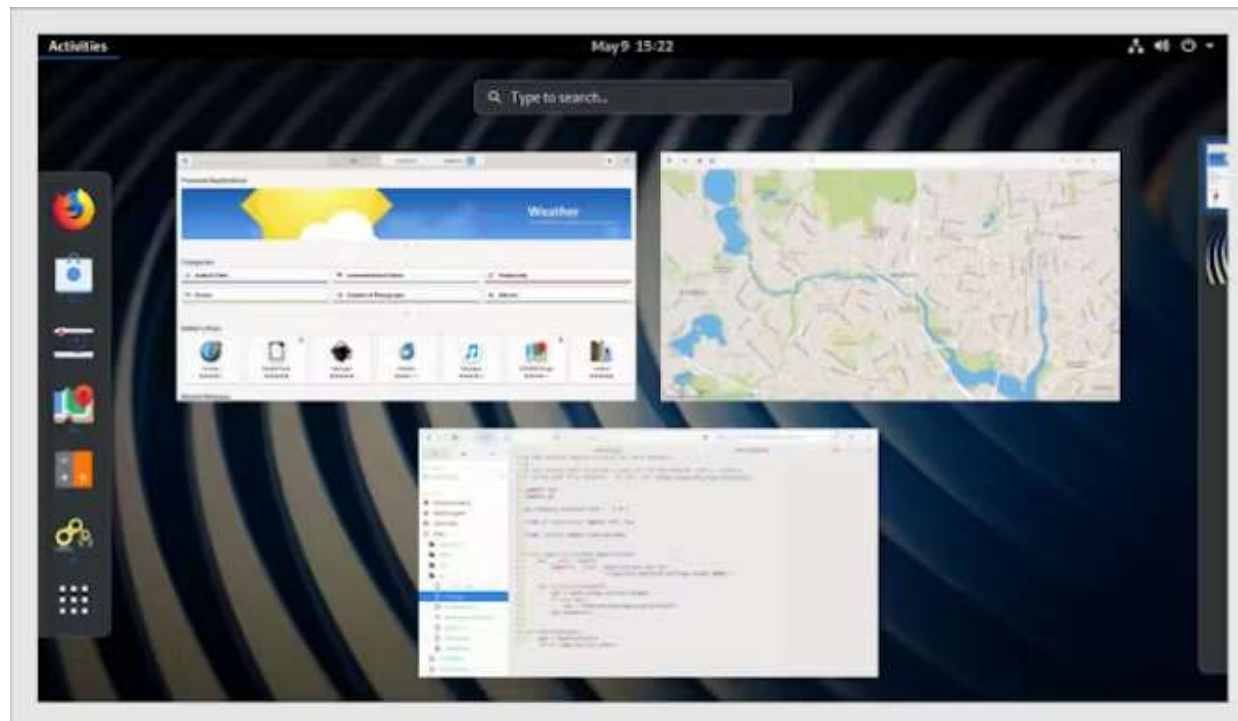
Editions

- ▶ **Fedora CoreOS:** This edition is intended for use in cloud computing environments and contains only the software packages required to run cloud-native applications. It's designed for use in containerized environments like Kubernetes, and it comes with tools for creating and managing container images.
- ▶ **Fedora IoT:** This edition is intended for use on Internet of Things (IoT) devices and includes the bare minimum of software required to run IoT applications

Fedora CoreOS



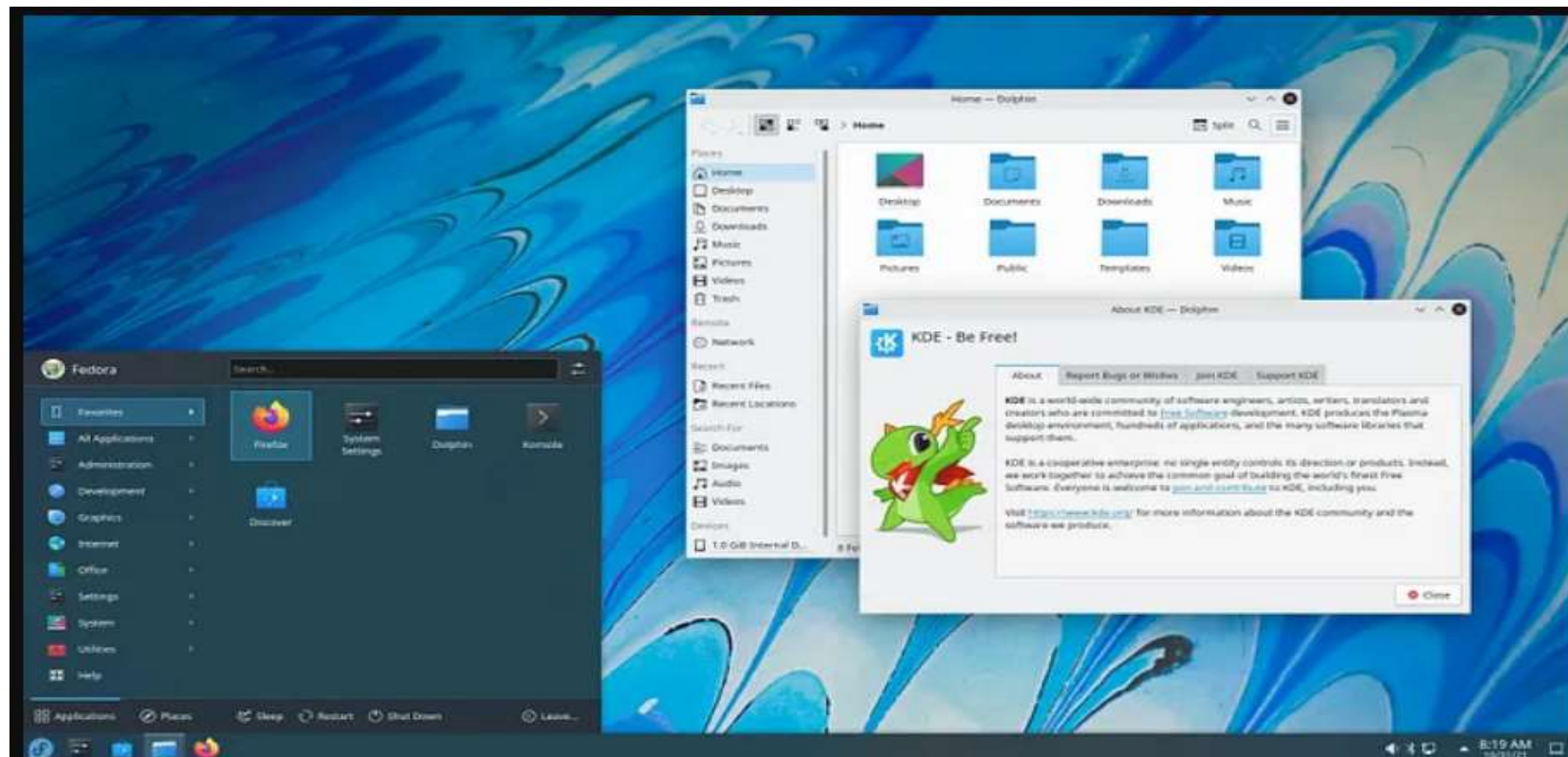
Fedora IoT



Editions

- ▶ **Fedora SilverBlue:** This is the best choice for users who want immutable desktops and for developers who use container-based workflows.
- ▶ In addition to these main editions, there are several Fedora spins tailored for specific use cases, such as scientific computing, design and artistic work, and gaming. These spins include a customized selection of software packages and tools tailored to these users' specific needs.

Fedora SilverBlue



Advantages

- ▶ Fedora OS is a very reliable and stable operating system.
- ▶ It enhances the security in this operating system.
- ▶ It offers many graphical tools.
- ▶ This operating system updates automatically.
- ▶ This OS supports many file formats.
- ▶ It also offers many education software.
- ▶ It supports a large community.
- ▶ It provides unique security features.

Disadvantages

- ▶ It requires a long time to set up.
- ▶ It requires additional software tools for the server.
- ▶ It does not provide any standard model for multi-file objects.
- ▶ Fedora has its own server, so we can't work on another server in real-time.

Assignments questions

- ▶ What is Open source software?
- ▶ What is need of open source software ?
- ▶ What are the advantages of open source software?
- ▶ What is Freeware?
- ▶ describe principles of open source software.
- ▶ describe open source software Evolution.

Assignments questions

- ▶ compare open source and closed source software.
- ▶ Explain various open source license.
- ▶ what is the application of open source software?
- ▶ what is Rights of Open-source licenses ?
- ▶ what is Rights of the Developer?

Assignments questions

- ▶ what is Rights of the user?
- ▶ what is Responsibilities of the Developer?
- ▶ what is Responsibilities of the Developer
- ▶ what is Fedora?
- ▶ describe Fedora Editions.
- ▶ What are the advantages of Fedora?