FOSS

Que. Answer in one line

1. **What is open-source development model? Ans:-** An open source development model is the process used by an open source community project to develop open source software.
2. **What is open-source software? Ans:-** 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.
3. **What is Patents? Ans:-** A patent gives its owner the right to exclude others from making, using, and selling the claimed invention.
4. **What is Wikipedia? Ans:-** Wikipedia is a free, open content online encyclopedia created through the collaborative effort of a community.
5. **What does OSPO stand for? Ans:-** Open Source Program Office (OSPO)

Q1. Explain Open source software(OSS) and Key Characteristics.

Ans:- The concept of open source software dates back to the 1950s and 1960s, but the term "open source" was popularized in the late 1990s. One of the early notable open source projects was the GNU Project initiated by Richard Stallman in 1983. The movement gained momentum with the founding of the Open Source Initiative (OSI) in 1998, which defined the "Open Source Definition" and coined the term "open source."

Open source software (OSS): Open source software (OSS) refers to software that is developed with a collaborative and transparent approach, allowing users to access, modify, and distribute its source code freely. Unlike proprietary software, where the source code is kept confidential and controlled by the software's creator or company, open source software encourages open collaboration and participation from a diverse community of developers. Here's an introduction to open source software:

Key Characteristics:

1. **Open Access to Source Code:** One of the defining features of open source software is that its source code is made available to the public. This transparency allows anyone to view and modify the code, fostering innovation and knowledge sharing.

2. **User Freedom:** Open source software is built on the principles of user freedom. Users have the freedom to use the software for any purpose, study how it works, modify it to suit their needs, and distribute both the original and modified versions.

3. **Collaboration:** Open source projects often involve collaboration among a global community of developers, each contributing their skills and knowledge to improve the software. This collaborative approach helps identify and fix bugs, enhance features, and ensure high-quality development.

4. **Licensing:** Open source software is typically distributed under open source licenses, which outline the terms and conditions under which the software can be used, modified, and redistributed. Various licenses exist, each with different levels of freedom and requirements.

5. **Transparency and Accountability:** The open nature of open source software development promotes transparency, accountability, and peer review. This makes it easier to identify security vulnerabilities and ensure the software's reliability.

Q2. Explain free software.

Ans:- Free software refers to software that respects users' essential freedoms and provides them with the liberty to use, study, modify, and distribute the software as they see fit. It is about the users' freedom rather than just the price of the software. The concept of free software is guided by principles that prioritize user empowerment and control over their digital environment. Here's an explanation of free software:

Key Principles: 1. **Freedom to Use:** Users have the freedom to run the software for any purpose without any restrictions.

2. **Freedom to Study:** Users can access and study the source code of the software to understand how it works and learn from it.

3. **Freedom to Modify:** Users can modify the software's source code to tailor it to their needs, fix issues, or add new features.

4. **Freedom to Distribute:** Users can share and distribute copies of the software to others, allowing them to benefit from the same freedoms.

Four Freedoms: 1. **Freedom 0 - Freedom to Run the Program:** You have the freedom to run the software as you wish, for any purpose.

2. **Freedom 1 - Freedom to Study How the Program Works:** You have access to the source code and can study it to understand its functioning.

3. **Freedom 2 - Freedom to Modify the Program:** You can modify the software's source code to suit your needs, and you can also share your modifications.

4. **Freedom 3 - Freedom to Share the Program:** You have the right to distribute both the original software and your modified versions.

Q3. Explain Public Domain Software and its Key Characteristics.

Ans:- Public domain software refers to software that has no copyright restrictions and is available for unrestricted use by anyone. Here's a detailed explanation: Definition: Public domain software is software that has been intentionally relinquished by its original author or creator, or the copyright has expired, making it available to the public without any legal restrictions. Once software is in the public domain, anyone can use, modify, distribute, and sell it without the need for obtaining permission or adhering to any specific licensing terms.

Key Characteristics: 1. **No Copyright Protection:** Public domain software has no copyright protection, which means that there are no legal limitations on how it can be used. Anyone can utilize the software for any purpose without infringing on copyright.

2. **No Attribution Required:** Unlike other open source licenses, there is no requirement to give credit to the original author or to attribute the software's origins when using or distributing public domain software.

3. **No Distribution Limitations:** Users are free to distribute public domain software in any manner they choose. This includes redistributing it as part of another project, selling it commercially, or making it available for free.

4. **No Warranty or Liability:** Since there are no copyright holders or licenses associated with public domain software, there are typically no warranties or liabilities attached. Users and developers assume full responsibility for how they use the software.

5. **Global Availability:** Public domain software can be used and distributed worldwide without geographical limitations.

Q4. Explain BSD(Berkeley Software Distribution) license.

Ans:-The BSD (Berkeley Software Distribution) license is a permissive open source software license that originated from the University of California, Berkeley. Here's a brief history of the BSD license: **1. **Origins:**** The BSD operating system was initially developed at the University of California, Berkeley, in the late 1970s and early 1980s. It was based on the Unix operating system and was used and developed by researchers and students at the university**.**

2. **Original BSD License:** The original BSD license was included with the BSD operating system's source code. It was a permissive license that allowed for the free use, modification, and redistribution of the software, as long as the copyright notice and disclaimers were retained.

3. **BSD Networking Release:** In 1989, the University of California, Berkeley, released the BSD Networking Release, which included a significant amount of networking code. This release contributed to the spread of BSD Unix and its codebase.

4. **Three-Clause and Four-Clause Licenses:** The original BSD license had a four-clause version and a three-clause version. The four-clause version included an advertising clause that required any product using the code to acknowledge the use of BSD software in its advertising materials. The three-clause version removed this advertising clause, making the license less restrictive**.**

5. **Modification by Other Projects:** Over time, various projects and derivatives of the BSD operating system emerged, each with their own modifications to the license. Some projects further simplified the license, resulting in the two-clause BSD license or the simplified ISC (Internet Systems Consortium) license.

6. **BSD License Influence:** The BSD license's permissive nature influenced other open source licenses, including the MIT License and the Apache License. These licenses allow for liberal use and modification of software while maintaining minimal restrictions on developers.

7. **Modern Versions:** In recent years, the original BSD license has been deprecated in favor of more permissive versions. The two-clause BSD license (also known as the "Simplified BSD License") is commonly used today. It retains the permissive terms while eliminating any advertising requirements.

Q5. Explain Open-Source Development Model.

Ans:- Open source software is developed collaboratively, with the source code made available to the public. The Open Source Initiative (OSI) provides the Open Source Definition, which outlines key principles, including free redistribution, access to source code, and allowing modifications. Open source projects often foster community-driven development and innovation.

- **Philosophy**: The open source development model emphasizes transparency, collaboration, and community-driven software development. It is based on the idea that making source code openly accessible to anyone promotes peer review, fosters innovation, and results in higher-quality software.

- **Importance**: Open source software often benefits from a diverse and global community of developers and contributors. This model has been widely adopted in various industries, from web development (e.g., Linux, Apache) to AI (e.g., TensorFlow) and beyond.

Q6. Explain The Apache HTTP Server.

Ans:- The Apache HTTP Server, commonly referred to as Apache, is a widely-used open-source web server software. It is one of the most popular web server applications globally and is known for its stability, security, and flexibility. Here are some key features and information about the Apache web server:

1. **Open Source:** Apache is released under the Apache License, which is an open-source license. This means the source code of the software is freely available for anyone to view, use, modify, and distribute.

2. **Cross-Platform:** Apache is designed to work on a variety of operating systems, including Unix-based systems (e.g., Linux and macOS), Microsoft Windows, and others. This cross-platform compatibility makes it a versatile choice for hosting websites on different systems.

3. **Extensible:** Apache's modular architecture allows users to extend its functionality by adding modules or plugins. This makes it adaptable to various requirements, such as serving dynamic content, handling authentication, and more.

4. **HTTP and HTTPS Support:** Apache primarily serves web content over HTTP (Hypertext Transfer Protocol) and can also support secure communication using HTTPS (HTTP Secure). This is crucial for encrypting data transmitted between the server and clients for secure web browsing.

5. **Virtual Hosting:** Apache supports virtual hosting, which allows multiple websites to be hosted on the same server, each with its own domain name and configuration. This is known as virtual hosting or name-based virtual hosting.

6. **Security Features:** Apache includes features and configurations to enhance server security, such as access control, authentication mechanisms, and support for SSL/TLS encryption.

7. **Community and Ecosystem:** Apache has a large and active community of developers and users who contribute to its development, provide support, and create additional modules and extensions. It is a part of the Apache Software Foundation, which oversees many open-source projects.

8. **Usage:** Apache is widely used by organizations, web hosting providers, and individuals to serve websites and web applications. It can handle a range of workloads, from simple static websites to complex dynamic web applications.

Q7. What is Copyrights & Copyleft?

Ans:- **Copyrights**: - Copyright is a legal concept that grants the creator of an original work (including software) exclusive rights to that work. These rights include the ability to control how the work is used, distributed, and modified.

- In software, copyright is the basis for licensing, as software developers retain copyright over their code, allowing them to specify how others can use it.

- Copyright does not prevent others from independently creating similar software; it only protects the specific expression of an idea.

Copyleft: - Copyleft is a strategy used in open source licenses, like the GPL, to ensure that derivative works also remain open source. It uses copyright law to require that if you modify and distribute copyleft-licensed software, you must also release those modifications under the same license.

- Copyleft licenses aim to preserve software freedom and prevent the enclosure of open source code into proprietary products.

-Copyleft is a specific licensing approach to ensure software remains open source.

Q8. What is Android & its Key features.

Ans:- Android is an open-source operating system designed primarily for mobile devices like smartphones and tablets. It's developed by Google and the Open Handset Alliance. Android is known for its flexibility, customization, and vast app ecosystem through the Google Play Store.

****1. Open-Source OS:****

- Android is an open-source operating system, meaning its source code is freely available for anyone to view, modify, and distribute. This open nature has led to a vibrant developer community and contributed to its widespread adoption.

****2. Mobile Platform:****

- Android is primarily used on mobile devices, including smartphones, tablets, and increasingly, other form factors like smartwatches and televisions.

****3. Developed by Google:****

- Android was initially developed by Android Inc., which was later acquired by Google in 2005. Google is the primary steward of the Android project.

****4. Customization and Flexibility:****

- Android is known for its high degree of customization. Manufacturers and developers can modify the Android OS to suit their needs, resulting in a wide range of Android-based devices with varying user interfaces and features.

****5. App Ecosystem:****

- Android has a vast ecosystem of applications available through the Google Play Store. These apps cover a wide range of categories, from productivity and gaming to social networking and multimedia.

****6. User Interface:****

- Android's user interface is highly customizable. Users can choose from different launchers, themes, and widgets to personalize their devices to their liking.

****7. Frequent Updates:****

- Google releases new versions of Android regularly, each with improvements, new features, and security updates. However, the adoption of these updates can vary among device manufacturers.

****8. Development Environment:****

- Android app development is typically done using the Android Studio integrated development environment (IDE). Developers can write apps in Java or Kotlin and use various libraries and tools provided by Google.

****9. Security and Privacy:****

- Google places a strong emphasis on security and privacy in Android. Features like app sandboxing, permission control, and regular security updates help protect users and their data.

Q9. Explain GitHub.

Ans:-

GitHub, the easiest way for developers to write software together, has scaled into a collaboration of 5.8 million developers across more than 12 million repositories worldwide. GitHub worked with Fastly to customize their CDN set up, ensuring rapid and efficient delivery of their content. Fastly serves all static assets and sits in front of GitHub.com, Pages (their website hosting service), and raw.github.com.

Open-Source Software (OSS) has been adopted not only for the personal purpose software products but also for core systems of companies and public institutions. It becomes indispensable for our society. In order to evolve and grow OSSs, it is important to acquire not only stakeholders but also contributors widely from outside. Especially, in the case of large OSS, a large number of contributors are needed. Therefore, it is extremely important to explore the acquisition mechanism of contributors.

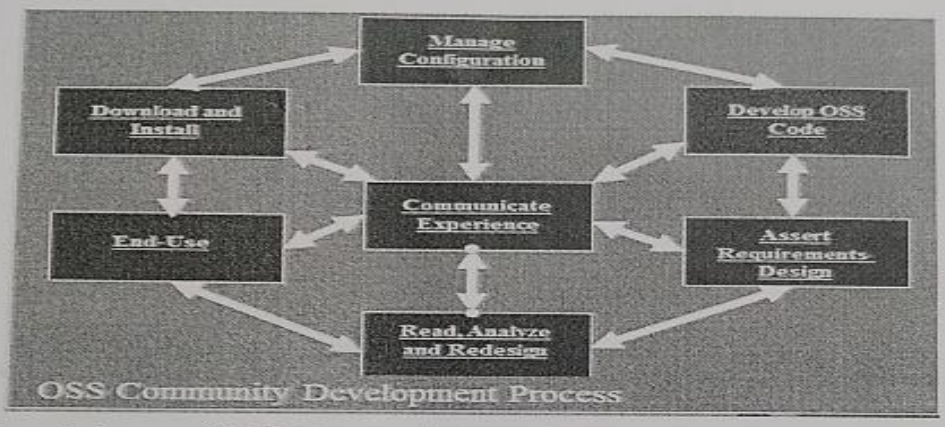
In this research, we try to elucidate the mechanism by analysing the OSS projects on GitHub. Although GitHub is an Internet hosting service aimed at supporting software collaborative development, there are many other uses that actually do not require contributors, such as free file storage or learning of version control system. To analyse collaborative software development projects, it is necessary to exclude projects other than the purpose.

To exclude unrelated use projects, we pay attention on so called contribution file. GitHub has a common file called contribution file (contributing.md/contributing.txt) to describe the necessary contribution for the project.

Q10. Explain Open source development model.

Open Source Development Model

The Open Source Development Model also involves an interconnected OSS Community Development Process in which each stage or phase plays a vital role in building the ethics of the community, keeping contributions of each developer in mind, working with the latest technologies, keeping a track on the version control system and fixing the bugs in the software.



The role of each phase of OSS Community Development Process are as follows,

- **Assert Requirements Design**

  This phase is all about the need for the software that is being proposed to the community for development and also the requirements during the development process are kept in mind in this phase. The design or approach to be followed during the development is also a part of this phase.

- **Develop OSS Code**

  The requirements are fulfilled; the need has been understood and the design is ready. Now, it's time for the contributors to start developing the backbone or the skeleton of the software by the means of Coding.

- **Manage Configuration**

  Once the basic OSS Code is developed it is necessary to provide an initial configuration too. Its integration works perfectly fine along with all the features provided with that particular version.

- **Download and Install**

  Once the initial version is ready and configured properly it is ready to go in the markets for general use.

- **End-Use**

  The users who require the software to fall into this phase and use the OSS daily to provide experience and feedback to the developers or for personal benefits.

- **Communicate Experience**

  Once it is deployed in the market, the users share their experience with the OSS and give feedback, suggestions, and reviews on the functions that are good to be intact and also on the features that could be enhanced or added in the later versions of the software.

- **Read, Analyze and Redesign**

  Once the feedbacks are registered, it all comes back to the developers to work on the feedbacks, keep updating their Software and also track control of version.

The Open Source Project Development Model is followed when a request is made to the developers to build a product to best suit their needs to which the developers keep in mind certain factors and phases to deliver a fully functional and a product with no compromises to the client.

Q11. Explain open source technology.

Ans:-

Open-source technology is a means of developing computer software through a more collaborative approach than most traditional software. When software is considered "open source," all or part of its source code is made available to the public or purchasers of the software. This allows programmers to modify or augment the software for their needs or the needs of others.

However, "open source" is a term with varying degrees of "openness." At its most basic, it implies that modifications can be made. Going a step further, it can also mean that users are kept up to date by the creators on the development of the software and can influence the choices made. It also depends on the terms of the licensing agreement as to whether or not modified versions of the software can be sold commercially.

Developing open-source software boasts a number of advantages for developers and customers. Since the source code is available for all to see, this allows both developers and users to search for bugs. This potentially allows for a more stable, well-tested product. Open source allows for more ideas to be pooled together, making for quicker refinement and innovation. It also helps develop brand loyalty, as users have a hand in the product development and become invested in its success. Most importantly, open-source software is frequently far less expensive to develop and purchase.

The downsides of open-source software are few but notable. If a company is especially concerned with how the modification of its software may affect its image should avoid open source. Allowing everyone to see source code means that competitors will see the source code. While this makes for good competition, it can also make for copycats developing inferior products aping the original. As far as users are concerned, caution should be exercised when utilizing some open-source software, as there may be few resources for technical support when a problem arises.

Q12. What is FreeBSD? Explain it.

Ans:- FreeBSD is an operating system for a variety of platforms which focuses on features, speed, and stability. It is derived from BSD, the version of UNIX® developed at the University of California, Berkeley. It is developed and maintained by a large community.

Cutting edge features

FreeBSD offers advanced networking, performance, security and compatibility features today which are still missing in other operating systems, even some of the best commercial ones.

Powerful Internet solutions

FreeBSD makes an ideal Internet or Intranet server. It provides robust network services under the heaviest loads and uses memory efficiently to maintain good response times for thousands of simultaneous user processes.

Advanced Embedded Platform

FreeBSD brings advanced network operating system features to appliance and embedded platforms, from higher-end Intel-based appliances to ARM, PowerPC, and MIPS hardware platforms. From mail and web appliances to routers, time servers, and wireless access points, vendors around the world rely on FreeBSD's integrated build and cross- build environments and advanced features as the foundation for their embedded products. And the Berkeley open-source license lets them decide how many of their local changes they want to contribute back.

Run a huge number of applications

With over 33,000 ported libraries and applications, FreeBSD supports applications for desktop, server, appliance, and embedded environments.

Easy to install

FreeBSD can be installed from a variety of media including CD- ROM, DVD, or directly over the network using FTP or NFS.

FreeBSD is free

While you might expect an operating system with these features to sell for a high price, FreeBSD is available free of charge and comes with the source code.

Contributing to FreeBSD

It is easy to contribute to FreeBSD. All you need to do is find a part of FreeBSD which you think could be improved and make those changes (carefully and cleanly) and submit that back to the Project by means of a bug report or a committer, if you know one. This could be anything from documentation to artwork to source code.

Q13. What is Android operating system & explain it?

Ans:- The Android OS was originally created by Android, Inc., which was bought by Google in 2005. Google teamed up with other companies to form the Open Handset Alliance (OHA), which has become responsible for the continued development of the Android OS.

Android's underlying kernel is based on Linux, but it has been customized to suit Google's directions. There is no support for the GNU libraries, and it does not have a native X Windows system. Inside the Linux kernel are found drivers for the display, camera, flash memory, keypad, Wi-Fi and audio. The Linux kernel serves as an abstraction between the hardware and the rest of the software on the phone. It also takes care of core system services like security, memory management, process management and the network stack.

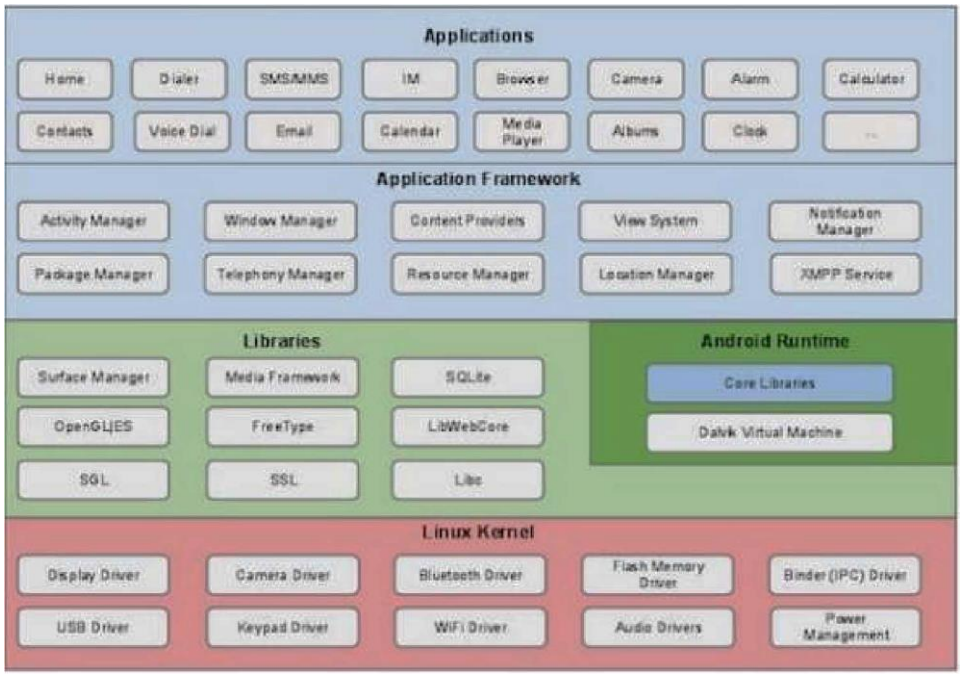
The Android OS is designed for phones. Its many features include:

- Integrated browser, based on the open-sourceWeb Kit engine
- Optimized 2D and 3D graphics, multimedia, and GSM connectivity
- Bluetooth
- EDGE
- 3G
- Wi-FiSQLite Camera GPSCompass Accelerometer

Software developers who want to create applications for the Android OS can download the Android Software Development Kit (SDK) for a specific version. The SDK includes a debugger, libraries, an emulator, some documentation, sample code and tutorials. For faster development, interested parties can use graphical integrated development environments (IDEs) such as Eclipse to write applications in Java.

Android story

- Android Inc was founded in Palo Alto, California,United States by Andy Rubin,RichMiner,Nick Sears & Chris White -- Oct 2003
- Google acquired Android Inc -Aug2005
- The Open Handset Alliance, a consortium of several companies was formed -5 thNov2007
- Android beta SDK released -12thNov2007



Q14. Explain Linux kernal in detail?

Ans:- Many computer users run a modified version of the GNU system every day, without realizing it. Through a typical turn of events, the version of GNU which is widely used today is often called "Linux". Linux is the kernel: the program in the system that allocates the machine's resources to the other programs that you run. The kernel is an essential part of an operating system, but useless by itself; it can only function in the context of a complete operating system. Linux is normally used in combination with the GNU operating system. The whole system is basically GNU with Linux added, or GNU/Linux. All the so-called "Linux" distributions are really distributions of GNU/Linux. By the early 90s put the whole system aside from the kernel. It also started a kernel, the GNU Hurd, which runs on top of Mach. Developing this kernel has been a lot harder than expected; the GNU Hurd started working reliably in 2001, but it is a long way from being ready for people to use in general. Many users do not understand the difference between the kernel, which is Linux, and the whole system, which they also call "Linux". The ambiguous use of the name doesn't help people understand. These users often think that Linus Torvalds developed the whole operating system in 1991, with a bit of help.Programmers generally know that Linux is a kernel. But since they have generally heard the whole system called "Linux". Most free software projects have the goal of developing a particular program for a particular job. For example, Linus Torvalds set out to write a Unix-like kernel (Linux); Donald Knuth set out to write a text formatter (TeX); Bob Scheifler set out to develop a window system (the X Window System). It's natural to measure the contribution of this kind of project by specific programs that came from the project.The GNU Project was not develop specific software packages. This Project develop a complete free Unix-like system: GNU.

At the bottom of the layers is Linux - Linux 3.6 with approximately 115 patches. This provides a level of abstraction between the device hardware, and it contains all the essential hardware drivers like the camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

Q15. Explain Social and Financial impacts of open-source technology.

Ans:-Today, mass collaboration is changing the essential structure of organizations and reshaping how these companies work in our exceptionally serious conditions. The collaborative effort, energized by open philosophies and companion generation, is constraining the administration to reconsider their methodologies. Collaborations that have emerged are breaking the obstructions and making open spaces where all can develop and add to push forward the limits of their organizations just as the limits of ventures, they work in the incomparable British economics specialist, makes a solid contention that one reason for the structure of vertically incorporated collaborations is the "cost of the exchange." perform an exchange inside firm in particular on the off chance that it is less expensive than performing it remotely or in the commercial centre. The Internet blast and the advancement of open-source software and pooled frameworks have made it feasible for online organizations to keep these exchange costs low. Wear Tapscott, the creator of Wikinomics, analyzes this thought: "Exchange costs despite everything exist, except now they're regularly graver in companies than in the commercial centre." Despite the considerable number of advantages (as far as quality, speed, and riches) that open source and the shared method of undertaking ventures have produced, there is still some misconception and hole in the valuation for these huge changes. An organization despite everything confines their appreciation of open source as free programming that sucks up the abundance of a solid entrepreneur society.

Q16. What is Open Solaris & Explain in detail?

Ans:- OpenSolaris is an open-source operating system, similar in scope to GNU/Linux and BSD, but descended from the proprietary Solaris operating system from Sun Microsystems. The OpenSolaris operating system is divided into three distinct but related aspects: the code, the distributions, and the community.

Features of Solaris

Security: Solaris includes some of the world's most advanced security features, such as Process and User Rights

Management: Trusted Extensions for Mandatory Access Control, the Cryptographic Framework and Secure by Default Networking that allow you to safely deliver new Solutions consolidate with security and protect mission-critical data.

Performance: Solaris delivers indisputable performance advantages for database, Web, and Java technology-based services, as well as massive scalability, shattering world records by delivering unbeatable price/performance advantages.

Networking: With its optimized network stack and support for today's advanced network computing protocols, Solaris delivers high-performance networking to most applications without modification.

Data Management: Solaris offers dramatic advances in file system and volume management by delivering virtually unlimited capacity and near- zero administration.

Interoperability: Understanding that businesses today rely on a mix of technologies from a variety of vendors, Solaris provides tools to enable seamless interoperability with hundreds of heterogeneous hardware and software platforms.

Observability: The Solaris release gives you Observability into your system with tools such as Solaris Dynamic Tracing (DTrace), which enables real-time application debugging and optimization.

Platform Choice: Solaris is fully supported on more than 900 SPARC- based and x64/x86-based systems from top manufacturers, including systems from Sun, Dell, HP, and IBM.

Q17. Explain Shared Source.

Ans:- Shared source is a software licensing concept that is more open than the proprietary approach to licensing but more restricted than the open source model. Under a shared source program license, authorized parties are granted full or partial access to source code. Typically, those granted access can view source code but cannot alter it for any commercial purpose. Some shared source programs allow only viewing of code; others allow non-commercial alteration and/or redistribution. Microsoft originated the shared source approach, which has since been adopted by other major industry players, including Hewlett- Packard and Sun Microsystems.

The shared source model offers fewer benefits than the open- source model. It lacks, for example, the collaborative improvement process promoted by the open-source approach. However, even the ability to view source code can be helpful. Shared source code can help developers ensure compatibility with existing programs and can make it easier to review source code for security purposes.

Critics have described shared source as a marketing ploy and suggested that the approach could pose a threat to the purity of the open- source model. In a paper called "Shared Source: A Dangerous Virus," the Open-Source Initiative called Microsoft's shared source program "a trap for the unwary" and warned that developers who'd been exposed to it should be considered "contaminated" and not assigned to projects that were competitive with Microsoft products.

Q18. What is open-source cloud & its use.

Ans:- Open-source cloud is any cloud service or solution that is built using open-source software and technologies. This includes any public, private or hybrid cloud model providing SaaS (Software as a Service), IaaS(Infrastructure as a Service) , PaaS(Platform as a Service) or XaaS(Anything as a Service) built and operated entirely on open-source technologies. The term “open cloud” in IT is a broad-based term for open-source or open-design technologies applied to cloud computing. Open source gives you the flexibility to deploy and, if necessary, migrate critical workloads across or off public cloud platforms. Some uses of the phrase “open cloud” involve specific company products. There are also agencies like the Open Cloud Consortium that use the term “open cloud” in relation to their overall goals and objectives. When using the term “open cloud,” many IT professionals refer to the objective of having transparent, versatile, scalable and easily controlled cloud systems. Cloud computing has become central to many enterprise IT models and many IT experts have been designing cloud systems to be as effective and beneficial as possible. Open cloud is part of the design philosophy that drives innovation in cloud-delivered services.

An open-source cloud is designed and developed using open-source technologies and software such as:

- Open-source operating system, DBMS and software development frameworks
- Open-source workflow and business applications
- Virtualization stack (Hypervisor, virtualization management)
- Hardware with open-source firmware

Moreover, open-source cloud may also refer to any cloud service that provides open-source software or service to end users or businesses. Businesses/cloud providers have the option to customize open-source cloud solutions to a greater extent, which is generally prohibited in closed-source cloud models. Open-source cloud solutions are interoperable with any back-end platform and can easily be migrated to a different IT infrastructure or environment. Examples of open-source cloud are Open Nebula, Open Stack and Virtual Box.

The use of an open-source cloud

The rapid growth of cloud computing has empowered hyperscaler cloud providers to market various technologies to feed the growing demand.Hyperscalers are now providing full-stack capabilities to increase their footprint and further lock-in customers, making the cloud seem more like a threat than an open communal space.

- Reduced operating costs.
- Avoidance of large capital investments in technology.
- Fulfilled security and compliance requirements.
- A solution to the shortage of skilled personnel.