“Київський фаховий коледж зв’язку”

ЗВІТ ПО ВИКОНАННЮ

ЛАБОРАТОРНОЇ РОБОТИ №1

з дисципліни: «Операційні системи»

Виконали

студенти

групи КСМ-13Б

Міньков Ілля

Колотуша Микола

**Готував матеріал студент Міньков Ілля**

**1. Порівняйте гіпервізори типу 1 та типу 2, яка між ними відмінність та сфера їх застосування?**

Type 1 Hypervisor vs. Type 2 Hypervisor: Comparison and Use Cases

Type 1 Hypervisor (Bare Metal Hypervisor):

Installation: A Type 1 hypervisor is installed directly on the physical server, allowing it to operate without a host operating system.

Performance: It typically provides better performance and lower latency as there is no need to process a host operating system.

Security: Type 1 hypervisors are generally more secure as they have a smaller attack surface compared to a host operating system.

Use Cases: This type of hypervisor is most commonly used in large data centers and server farms for server consolidation, resource allocation, and ensuring isolation between virtual machines. Examples include VMware vSphere/ESXi, Microsoft Hyper-V (when used without a host operating system), and KVM.

Type 2 Hypervisor (Hosted Hypervisor):

Installation: A Type 2 hypervisor is installed as an application on the host operating system, meaning it requires a functioning host OS to operate.

Performance: It may have more overhead compared to Type 1 due to the need to process the host operating system.

Security: Type 2 hypervisors can be less secure as the host operating system can be vulnerable, potentially affecting the virtual machines.

Use Cases: Type 2 hypervisors are more commonly used on workstations and testing environments where performance is less critical or when you need to quickly spin up a virtual machine on an existing operating system. Examples include Oracle VirtualBox, VMware Workstation, and Parallels Desktop for Mac.

So, the primary difference between them lies in their installation method and performance level. When choosing between them, it's important to consider the specific needs and requirements of your project.

**Готував матеріал студент Міньков Ілля**

**2. Розкрийте поняття «GNU GPL», яка його основна концепція?**

"GNU GPL" stands for "GNU General Public License," or simply "GPL." It is one of the most well-known and widely used open-source licenses in the world of software. Its primary concept is to ensure the freedom and openness of software code.

Key concepts and principles of GNU GPL:

Freedom of Use: GNU GPL allows users to freely use, modify, copy, and distribute the software it covers. This means that users have the right to use the program for any purpose, including commercial use.

Open Source: The license requires that all source code of the software be made available to users. This allows other developers to review, modify, and improve the program.

Requirement of Source Code Disclosure: If users make changes to the program and distribute it, they must also provide free access to the source code of those changes. This ensures openness and the dissemination of improvements.

Community: GNU GPL fosters the development of a large community of free software developers and users. This community collaboratively works on enhancing the software, making it available to everyone.

Protection from Patents: GNU GPL contains provisions that protect users from patent claims related to the software. If a developer patents technology used in the program, that patent cannot be used to restrict the free use of the software.

GNU GPL helps preserve the openness and freedom of software code, encourages collaboration and improvement of programs, and supports the concept of free software. Many open-source projects, such as Linux and the GNU Compiler Collection (GCC), use the GNU GPL license to ensure the freedom and accessibility of their source code.

**Готував матеріал студент Міньков Ілля**

**3. В чому суть програмного забезпечення з відкритим кодом?**The essence of open-source software lies in making the entire source code of a program available for public viewing, modification, use, and distribution. The core idea of open source is to promote freedom, collaboration, and accessibility of software. Here are some key aspects of the essence of open-source software:

Access to Source Code: In open-source software, the entire source code (instructions that make up the program) is open and accessible for viewing through specific licenses that guarantee this accessibility.

Freedom of Use: Users have the right to freely use open-source software for any purpose, from personal use to commercial applications, without restrictions or licensing costs.

Ability to Modify: Users can make changes to the source code of the software and create their own versions of the program to meet their needs or fix bugs.

Distribution and Redistribution: Users have the right to distribute open-source software and redistribute their modifications with the source code. This promotes the spread and development of the software.

Developer and User Community: Open-source software typically has an active community of developers and users who collaborate on development, improvement, and support of the software.

Transparency and Security: Open-source software is subject to public scrutiny, leading to the discovery and correction of errors and enhancing overall security.

Open-source software includes projects such as Linux (an operating system), Apache HTTP Server (a web server), Mozilla Firefox (a web browser), and many others. This approach to software development fosters innovation, collaborative development, and openness in the IT industry.

4. **Що таке дистрибутив?**In the context of software and operating systems, the term "distribution" (or "distro") refers to a package or version of software or an operating system that includes the software itself, as well as various additional components and configurations to provide full functionality and readiness for use.

Key characteristics of a distribution:

Operating System: A distribution includes an operating system (e.g., Linux, Windows, or another) that serves as the foundation for computer or server operation.

Software Packages: A distribution may include certain pre-installed or available-for-installation programs and applications.

Additional Libraries and Components: Often, a distribution contains additional libraries, drivers, configurations, and other components that tailor the operating system for specific tasks or types of hardware.

Utilities and System Administration Tools: A distribution may come with a set of tools for system management, network configuration, remote access, and more.

License and Documentation: Each distribution typically comes with an appropriate license that defines usage rules and documentation that provides instructions for installation and system setup.

Distributions are commonly used to simplify the process of installing and configuring operating systems, ensure compatibility of software and components, and distribute operating systems and software to consumers and users. They can be officially supported by organizations (e.g., Canonical for Ubuntu Linux) or created by developer communities (e.g., Linux distributions like Debian or Fedora).