**"Kyiv Vocational College of Communication"**

**Cyclic Commission of Computer Engineering**

**EXECUTION REPORT**

**LABORATORY WORK No. 1**

from the discipline: "Operating systems"

**Topic: "Getting to know the working environment virtual machines and operating systems of different families"**

**Performed by students of the group:**

Барабаш Матвій

Погребняк Ілля

Целуйко Станіслав

**Checked by the teacher**

Sushanova V.S.

**Work of group students КСМ-13Б Team:** **PMC wolf group**

**Kyiv 2023**

**The goal of the work:**

1. Obtaining practical skills for working with virtual machine environments and operating systems of various types and families - their graphics shell, entry and exit from the system, familiarization with the structure desktop, studying basic actions and settings when working in the system.

**Material provision of classes**

1. IBM PC type computer.

2. OS family Windows (Windows 7).

3. Virtual machine - Virtual Box (Oracle).

4. GNU/Linux operating system - CentOS.

**Tasks for preliminary preparation**

The student prepared the material: Барабаш Матвій

1. Read the short theoretical information for the laboratory work and do it a small dictionary of basic English terms on OS classification issues.

|  |  |
| --- | --- |
| **Термін англійською** | **Термін українською** |
| shared hosting | спільний хостинг |
| type 1 hypervisor | гіпервізор типу 1 |
| machine simulators | машинні тренажери |
| binary translation | двійковий переклад |
| type 2 hypervisors | гіпервізори 2 типу |
| host operating system | операційна система хоста |
| guest operating system | гостьова операційна система |
| The Java Virtual Machine | Віртуальна машина Java |
| JVM (Java Virtual Machine) | JVM (Віртуальна машина Java) |
|  |  |

2. After reading the material from short theoretical information, give the answers to the following questions

**The material was prepared by student Погребняк Ілля**

The term "hypervisor" (sometimes known as a virtualizer or virtualization hypervisor) refers to the software or hardware used to create and manage virtual machines (VMs) on physical hardware. Hypervisors allow you to share computer resources, such as CPU, memory, and storage, between different virtual environments running on the same physical server. This allows efficient use of hardware and isolation of different VMs from each other.

**There are two main types of hypervisors:**

**1. Hypervisor of the 1st level (Type 1):**

- Also known as "native" or "direct" hypervisor.

- Runs directly on the server's physical hardware.

- Has direct access to hardware resources and usually runs at a lower level than the host operating system.

- Usually used in large data centers for server virtualization.

**2. Hypervisor of the 2nd level (Type 2):**

- Also known as "operating system layer hypervisor" or "software layer hypervisor".

- Installs on a host operating system that is already running on a physical server.

- Virtual machines are created and run in the context of the host operating system.

- Typically used for development and testing or on desktop computers.

**Progress**

The material was prepared by student Погребняк Ілля

1. Work in graphical mode in the OS of the Linux family.

1.1. Start the **VirtualBox** virtual machine, familiarize yourself with its main features capabilities, read the help for working with it.

*VirtualBox is a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 3. See "About VirtualBox" for an introduction.*

*Presently, VirtualBox runs on Windows, Linux, macOS, and Solaris hosts and supports a large number of guest operating systems including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7, Windows 8, Windows 10), DOS/Windows 3.x, Linux (2.4, 2.6, 3.x and 4.x), Solaris and OpenSolaris, OS/2, and OpenBSD.*

*VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria.*

VirtualBox is a free hypervisor that allows you to create and manage virtual machines on your computer. The main capabilities of VirtualBox include:

1. Support for different operating systems: You can install different operating systems in virtual machines, including Windows, Linux, macOS and others.

2. Resource sharing: You can configure file and folder sharing between the host operating system and virtual machines.

3. Network settings: VirtualBox allows you to use virtual networks, Internet connections, network filters and routing rules.

4. You can create and store images of virtual machines on your computer.

5. Snapshots: VirtualBox allows you to create snapshots of virtual machines, which allows you to save the state of the system at a certain point in time and restore it when needed.

6. Resource management: You can finally separate resources (processors, memory, video cards) for each virtual machine.

**Answers to control questions**

**The student prepared the material: Целуйко Станіслав**

The GNU GPL (GNU General Public License), or simply GPL, is a free software license developed by the Free Software Foundation (FSF). The core concept of the GNU GPL is to provide freedom to users, developers, and the community with broad standards that guarantee free operation and access to software code.

- The basic principles and concepts of the GNU GPL include the following:

1. Freedom to use the software: Users have the right to freely use the software for any purpose without restriction.

2. Freedom to explore and modify the program: The GNU GPL guarantees the right of users to access the source code of the program and to modify it in order to improve, adapt, or correct errors.

3. Freedom to distribute the program: The main idea behind the GPL is that if you distribute a program (or a modified version), you must also make the source code of that program available to users under the same license.

4. No restrictions on use: The GPL does not allow you to add restrictions or conditions that contradict the basic principles of free software. This means that users cannot be restricted in their right to use the program or its modified version for any purpose.

5. Openness and transparency: The main characteristics of the GPL are the openness of the source code and the transparency of the license terms. It promotes the development of the development community and supports the ideas of free software.

The GNU GPL is sometimes used to license a large amount of free and open source software, including operating systems, text editors, web servers, and other projects. This license plays an important role in the development of free software and promotes the spread of free and open technologies.

**The student prepared the material: Барабаш Матвій**

**2. The following control questions and their answers**

**1.** What is open source software?

The essence of Open Source Software (OSS) is that the source code of the program is available for public viewing, use, modification and distribution. Key aspects and principles of open source software include:

1. Free access to source code: Open source software is available to any user for free. You can view the source code and use it for your own purposes.

2. Right to Modification: Users have the right to modify the source code of the program according to their needs or to solve problems. This allows you to adapt the software to the specific requirements of the user.

3. Free distribution: You can freely distribute open source software, share it with other users, and even incorporate it into your own projects.

4. Free license: Most open source projects use specific licenses that define the rules for using, modifying and distributing the source code. These licenses often ensure that the source code remains open and available.

5. Collaborative development: Open source software is often developed by a community of volunteers or professional developers working together on a project. This can lead to rapid development and problem solving because many people have access to the source code.

6. Open discussion and community: Open source projects often support open discussions, forums, and user communities that facilitate sharing of experiences, problem solving, and collaboration between developers and users.

7. Code security and auditing: Open access to the source code helps identify and fix potential security vulnerabilities. Many people can audit the code for bugs and security threats.

Open source software promotes the development and distribution of free and open source software. It has a positive impact on innovation, reduces dependence on a single software manufacturer, and contributes to the development of safer and more stable products. Many well-known projects, such as Linux, Apache, Mozilla Firefox and others, are based on the open source model.

**2.** What is a distribution?

In the context of software and operating systems, the term "distribution" (or "Linux distribution") refers to the set of programs and other components that comprise the Linux operating system and various additional software, stored as packages. Linux distributions are ready-to-install and ready-to-use versions of the Linux operating system that can be installed on computers or servers.

The main characteristics of Linux distributions include:

1. Operating system: Distributions contain the Linux kernel and the main components of the operating system, such as the command line shell, system libraries and utilities.

2. Software packages: Additional programs and desktop environments are included in the distribution, which may include web browsers, text editors, graphics games, servers, development tools, and much more. These programs are usually packaged as packages for easy installation and updating.

3. Package Management Systems: Linux distributions use package management systems such as APT (for Debian and Ubuntu), YUM and DNF (for Fedora and CentOS), Pacman (for Arch Linux) and many others to install and manage software packages. .

4. Setup and configuration: Distributions usually include tools for system setup, network setup, user and access rights management, network setup, and many other administrative tasks.

Popular Linux distributions include Debian, Ubuntu, Fedora, CentOS, Arch Linux, openSUSE, Slackware, and many others. Each of them can have its own specifics, target audience and package distribution methods, which makes them suitable for different types of tasks and users. Users can choose the distribution that best suits their needs and requirements.

**3.** What tasks of system administration can be implemented on the basis of the Linux OS?

A system administrator, working with the Linux operating system, can perform various tasks related to the management and support of the infrastructure of information systems. Here are some tasks that can be implemented on the basis of the Linux OS:

1. Installation and configuration of operating systems: A system administrator can install and configure various Linux distributions on servers and computers.

2. Management of users and access rights: The administrator can create, block and delete users, as well as configure access rights to files and resources.

3. Network settings: Administrators can manage network settings, set IP addresses, configure DNS, routing and firewalls.

4. Package and Program Management: Install, update, and uninstall software using package management systems such as APT, YUM, DNF, or Pacman.

5. Monitoring and diagnostics: A system administrator can use monitoring tools such as Nagios, Zabbix or Prometheus to track the health of servers and resources.

6. Data backup and recovery: Ensuring data security by creating backup copies and restoring information in case of accidents or data loss.

7. Ensuring security: Configuring firewalls, access control, system updates to ensure protection against potential security threats.

8. Server administration: Management of servers, such as web servers (Apache, Nginx), databases (MySQL, PostgreSQL), e-mail (Postfix, Sendmail), file servers and many others.

9. Automate Tasks: Using automation tools like Ansible, Puppet, Chef, or Shell scripts to automate everyday tasks and processes.

10. Scaling and Optimizing Performance: Expanding and optimizing infrastructure to ensure high performance and performance.

This is just a small list of tasks that can be implemented on the basis of the Linux operating system. The work of a system administrator with Linux can be very diverse and depends on the specific needs and infrastructure of the organization.

**4.** How are Android and Linux OS related?

The Android operating system and the Linux kernel have a complex relationship, and Android can be considered a variant of the operating system built on top of the Linux kernel. Here's how they're related:

1. Linux Kernel: Android uses the Linux kernel as the basis for handling the device's hardware resources such as CPU, memory, peripherals and others. The Linux kernel provides a basic level of abstraction for interacting with the hardware and enabling the device to operate.

2. Android-specific runtime stack: On top of the Linux kernel, Android has its own runtime stack, which includes libraries, frameworks, and a Java environment for running applications. One of the key components of this stack is the Android Runtime (ART), which executes Java programs written for Android.

3. Applications and software: Android works with applications and software created for mobile devices. Android applications use Java for their development and execution, and can use native code written in C/C++ to optimize performance.

4. Common principles and libraries: Android inherits certain common principles and libraries from Linux, such as network protocols, file systems, security, and many others. This makes it possible to use many common tools and resources.

5. Open source: The Linux kernel and many components of the Android operating system are based on free and open source software, which means that their source codes are available for review and modification.

In general, Android and Linux are similar in that they are based on the Linux kernel and shared open source software concepts, but they also have their own unique components and functionality specific to mobile devices. Android is one of the most popular operating systems for smartphones and tablets, and it is intensively developed and adapted for different devices and user requirements.

**5.** Main capabilities and scope of use of Embedded Linux?

Embedded Linux is a version of the Linux operating system that is designed for embedded systems and devices such as embedded computers, microcontrollers, Internet of Things (IoT) devices, automotive systems, and many others. The main capabilities and scope of use of Embedded Linux include the following:

1. Low-level access to hardware: Embedded Linux enables developers to access hardware, including processors, memory, peripherals, and other components, and control them with their own code.

2. Support for different architectures: Embedded Linux supports many different processor architectures such as ARM, MIPS, PowerPC and many others, allowing it to be used on different devices.

3. System minimization: Developers can create specialized and minimalist versions of Embedded Linux that have only those components and functions that are necessary for a specific device or application. This reduces the requirements for memory and computing resources.

4. Support for real-time (Real-Time): Embedded Linux can be configured to support real-time, which makes it suitable for use in systems where the device's response to real-time events is important.

5. Open Source: Many Embedded Linux implementations are based on free and open source software, allowing developers to modify and customize the system to their needs.

6. Network Support: Embedded Linux has a rich set of networking tools and protocols, making it suitable for use in networking devices, IoT devices, and communication systems.

7. Application Development: Embedded Linux provides tools for developing applications and software that can run on embedded devices, including libraries and frameworks for creating applications.

8. Update and Remote Management: Embedded Linux allows you to update device software remotely, which simplifies the management of a large number of devices in a distributed network.

The field of use of Embedded Linux is very diverse and covers industries such as automotive, medical devices, IoT devices, robotics, network devices, industrial control and many others. Its flexibility, broad developer community, and abundance of available software make Embedded Linux an essential tool for building embedded systems.

**Conclusions**

In the course of the laboratory work, I studied the VirtualBox program, more the issue of: Virtual machine - Virtual Box (Oracle), GNU/Linux operating system - CentOS is studied in detail.