**"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:  
Греков Данііл  
Богдан Раєв**

**Checked by the teacher**

Sushanova V.S.

**Work of group students КСМ-13Б Team:** **bed0lagi**

**Kyiv 2023**

**The goal of the work:**

1. Acquiring practical knowledge for using virtual machine environments and operating systems of different kinds and families, including their graphics shell, methods for entering and leaving the system, knowledge of the desktop's layout, and basic actions and settings to study when using 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: Богдан Раєв

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

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

**The material was prepared by student Богдан Раєв**The software or hardware used to create and manage virtual machines (VMs) on physical hardware is referred to as a "hypervisor" (also known as a virtualizer or virtualization hypervisor). With the aid of hypervisors, you are able to distribute CPU, memory, and storage between various virtual environments running on the same physical server. This enables the separation of various VMs from one another and the effective use of hardware.  
  
**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.

- Runs more slowly than the host operating system and has direct access to hardware resources. .

- Typically used for server virtualization in large data centers.

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

- Also referred to as "software layer hypervisor" or "operating system layer hypervisor".

- Installs an operating system on a physical server that is already in use..

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

- Usually applied to desktop computers or for use in development and testing.

**Progress**

The material was prepared by student Греков Данііл

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.

Powerful x86 and AMD64/Intel64 virtualization software for business and home use is called VirtualBox. In addition to being the only business-oriented solution that is freely accessible as Open Source Software under the terms of the GNU General Public License (GPL) version 3, VirtualBox is also a very feature-rich, high-performance product. For an introduction, see "About VirtualBox".

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

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.

You can build and control virtual machines on your computer using VirtualBox, a free hypervisor. The primary functions of VirtualBox are as follows:

1. Support for a variety of operating systems: Windows, Linux, macOS, and other operating systems can be installed in virtual machines.

2. Resource sharing can be set up to share files and folders between virtual machines and the host operating system.   
3. Network configurations: With VirtualBox, you can use virtual networks, Internet connections, network filters, and routing rules.

4. Your computer can be used to create and store virtual machine images.   
5. Snapshots: You can take snapshots of your virtual machines using VirtualBox, which enables you to save the state of the system at a specific time and restore it later.

6. The ability to manage resources (processors, memory, and video cards) for each virtual machine is finally available.  
**Answers to control questions**

**The student prepared the material: Греков Данііл** 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. Software use rights: Users have the unrestricted freedom to use the software for any purpose.

2. Freedom to investigate and alter the program: The GNU GPL guarantees users' access to the program's source code and their ability to alter it to enhance, adapt, or fix bugs.

3. Freedom to distribute the program: The GPL's central tenet is that anyone who distributes software (or a modified version) must also make the source code of that software available to users under the same license.

4. No usage restrictions: The GPL forbids the addition of conditions or restrictions that go against the fundamental tenets 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. Transparency and openness: The GPL's main attributes are the transparency of the license terms and the openness of the source code. It supports the principles of free software and encourages the growth of the development community.

Many different types of free and open source software, such as operating systems, text editors, web servers, and other projects, are occasionally licensed under the GNU GPL. This license is crucial to the creation of free software and encourages the adoption of open source and free technologies.  
**The student prepared the material: Богдан Раєв   
2. The following control questions and their answers**

**1.** What is open source software?  
The fact that the program's source code is accessible to the general public for viewing, use, modification, and distribution defines Open Source Software (OSS). Open source software's fundamental features and tenets include:

1. Free source code access: Anyone can use open source software for nothing. You can look at the source code and use it however you like.

2. Right to Modification: Users have the right to change the program's source code to suit their needs or to fix issues. This enables you to modify the software to meet the unique needs of the user.   
3. Free distribution: Open source software can be shared with others, used in your own projects, and distributed at no cost.

4. Free license: The majority of open source projects use particular licenses that specify the limitations on how the source code may be used, altered, and distributed. The source code is frequently kept open and accessible thanks to these licenses.

5. Collaborative development: Open source software is frequently created by a group of professionals or volunteers who collaborate on a project. Given that so many people have access to the source code, this can result in rapid development and problem solving.   
6. Open discussion and community: Open source projects frequently support public forums, user communities, and discussions that encourage experience sharing, collaboration, and problem solving between developers and users.

7. Code security and auditing: Having free access to the source code enables the detection and correction of potential security flaws. The code can be audited by numerous people for errors and security risks.

The creation and dissemination of free and open source software is encouraged by open source software. It promotes innovation, lessens reliance on a single software vendor, and helps to create products that are more reliable and stable. The open source model is the foundation for many well-known projects, including Linux, Apache, Mozilla Firefox, and others.  
**2.** What is a distribution?

The term "distribution" (or "Linux distribution") in the context of software and operating systems refers to the collection of applications and other elements that make up the Linux operating system as well as other additional software that is stored as packages. On computers or servers, Linux distributions can be installed as ready-to-use, ready-to-install versions of the Linux operating system.

The primary traits of Linux distributions are as follows:

1. Operating system: Distributions come with the Linux kernel and all of the essential utilities, system libraries, and command-line shells.

2. Software bundles: Additional applications and desktop environments, such as web browsers, text editors, graphics games, servers, development tools, and much more, are included in the distribution. Typically, these programs are packaged as packages for easy instalation  
  
3. Package Management Systems: To install and manage software packages, Linux distributions use package management systems like APT (for Debian and Ubuntu), YUM and DNF (for Fedora and CentOS), Pacman (for Arch Linux), and many others.

4. Configuration and setup: Distributions typically come with tools for network setup, user and access rights management, system setup, and a variety of other administrative tasks.

Debian, Ubuntu, Fedora, CentOS, Arch Linux, openSUSE, Slackware, and many other popular Linux distributions are available. They can each have unique features, target markets, and package distribution strategies, making them suitable for various tasks and user types. Users have a choice of distributions based on their needs and specifications.  
**3.** What tasks of system administration can be implemented on the basis of the Linux OS?

Working with the Linux operating system, a system administrator is able to manage and support the infrastructure of information systems in a variety of ways. On the basis of the Linux operating system, the following tasks can be carried out:

1. Operating system installation and configuration: A system administrator can set up different Linux distributions on computers and servers.

2. Management of users and access rights: The administrator can set up access rights to files and resources as well as create, block, and delete users.   
3. Network settings: Administrators have control over network settings, including IP addresses, DNS, firewalls, and routing.

4. Package and Program Management: Use package management tools like APT, YUM, DNF, or Pacman to install, update, and uninstall software.

5. Monitoring and diagnostics: To keep tabs on the condition of servers and resources, a system administrator can use monitoring tools like Nagios, Zabbix, or Prometheus.

6. Data backup and recovery: Ensuring data security by creating backup copies of data and restoring it in the event of a disaster or data loss.

7. Using access control, firewall configuration, and system updates to defend against potential security threats. 8. Server administration: Managing servers, including file servers, database servers (MySQL, PostgreSQL), email servers (Postfix, Sendmail), Apache, Nginx, and many others.

9. Automate Tasks: To automate routine tasks and processes, use automation tools like Ansible, Puppet, Chef, or Shell scripts.

10. Scaling and Performance Optimization: Improving and enhancing infrastructure to ensure high performance.  
   
This is merely a short list of the things that can be done using the Linux operating system. With Linux, a system administrator's duties can be extremely varied and are based on the infrastructure and unique requirements of the organization.  
**4.** How are Android and Linux OS related?

The relationship between the Linux kernel and the Android operating system is complicated, and Android can be seen as a different version of the kernel-based OS. These are their connections:

1. Linux Kernel: Android manages hardware resources like the CPU, memory, peripherals, and others by using the Linux kernel as its foundation. A fundamental level of abstraction for interacting with the hardware and enabling the device to function is provided by the Linux kernel.

2. The Android runtime stack, which sits on top of the Linux kernel, consists of libraries, frameworks, and a Java environment for running applications. The Android Runtime (ART), which executes Java programs created for Android, is one of the essential elements of this stack.   
  
3. Software and applications: Android is compatible with software and applications designed for mobile devices. Android applications can use native C/C++ code written in Java for performance optimization while also using Java for development and execution.

4. Common principles and libraries: Linux is the source of many of the common principles and libraries that Android inherits, including security, file systems, and network protocols. The use of numerous common tools and resources is made possible by this.

5. Open source: The Android operating system and the Linux kernel both rely on free and open source software, which makes their source codes accessible for inspection 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?

An operating system called Embedded Linux is specifically made for embedded systems and gadgets like embedded computers, microcontrollers, Internet of Things (IoT) gadgets, automotive systems, and many more. The following are some of Embedded Linux's primary features and areas of application:

1. Low-level hardware access: Embedded Linux gives programmers access to the hardware, such as the processors, memory, peripherals, and other parts, so they can control them with custom code.

2. Support for various processor architectures: Embedded Linux can be used on a variety of devices because it supports a wide range of processor architectures, including ARM, MIPS, PowerPC, and many others.

3. System minimization: Programmers can create customized, stripped-down versions of Embedded Linux that contain only the parts and features required by a particular machine or application. As a result, less memory and computing power is needed.

4. Real-time support: Embedded Linux can be set up to support real-time, making it appropriate for use in systems where the device's reaction to real-time events is crucial.   
5. Open Source: Since many Embedded Linux implementations are built using free and open source software, programmers can change and adapt the system to suit their needs.

6. Network Support: Embedded Linux is suited for use in networking devices, IoT devices, and communication systems thanks to its extensive set of networking tools and protocols.

7. Application Development: Embedded Linux offers libraries and frameworks for building applications, as well as other tools for creating software that can be used on embedded devices.

8. Update and Remote Management: Embedded Linux enables remote software updates for devices, making it easier to manage numerous devices across a distributed network.

Embedded Linux can be used in a wide range of industries, including the automotive, medical, IoT, robotics, network, industrial control, and many more. Embedded Linux is a crucial tool for developing embedded systems because of its adaptability, sizeable developer community, and abundance of readily available software.  
 **Conclusions**

As part of my laboratory work, I studied the VirtualBox software, paying particular attention to the following: Virtual Machine - Virtual Box (Oracle); GNU/Linux Operating System - CentOS.