"Kyiv Vocational College of Communication"

Cyclic Commission of Computer Engineering

EXECUTION REPORT

**EXECUTION REPORT**

**LABORATORY WORK No. 1**

**Topic: "Getting to know the working environment of virtual machines and the features of the Linux operating system"**

It was performed by students of the RPZ group - 03B

Team 6: Sichkar Maxim,

Brytyuk Bohdan

Kyiv 2023

**The goal of the work:**

1. Familiarity with various types of hypervisors, virtualization when working with operating systems.

2. Familiarity with the main types of modern operating systems, a brief overview of their capabilities.

**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.

5. Cisco network academy site netacad.com and its online Linux courses

**Tasks for preliminary preparation.**

**The student prepared the material:**  Brytyuk Bohdan.

1. Read the short theoretical information for the laboratory work and make a small dictionary basic English terms on classification of virtual environments.

|  |  |
| --- | --- |
| **The term is in English** | **The term is in Ukrainian** |
| physical machine | фізична машина |
| software | програмне забезпечення |
| hypervisor technology | технологія гіпервізора |
| virtual devices | віртуальні пристрої |
| high bandwidth | висока пропускна здатність |
| virtualization | віртуалізація |
| user mode | режим користувача |

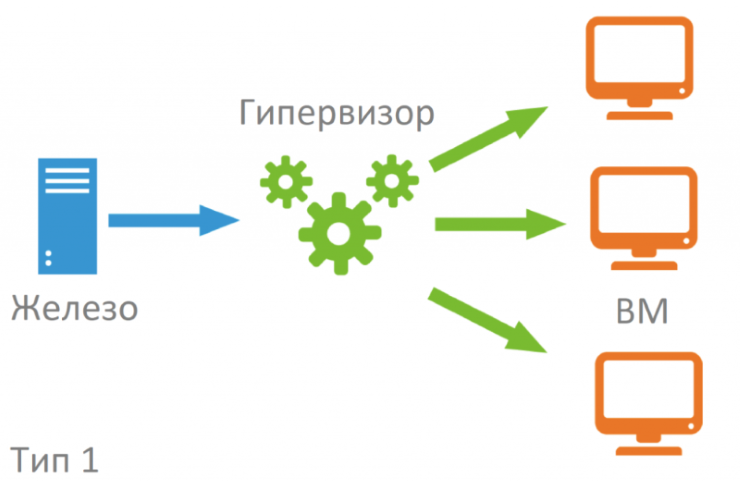
2. After reading the material from short theoretical information, answer the following questions:

**The student prepared the material:**  Brytyuk Bohdan

**2.1. Describe the concept of "hypervisor". What are their types?**

* A hypervisor is software that allows you to separate physical hardware from programs running in a computer's operating system. You can't "touch" it, it's all just unique software.
* When using hypervisor technology, it is possible to manage virtual devices from one host computer with one key. They act as guest operating systems. This kind of work makes the most efficient use of computing resources. The equipment receives high bandwidth. In short, the hypervisor is the real disruptive force that enables virtualization and VPS to function properly.
* Virtualization is a technology that allows you to create several virtual machines from one physical computer or server. The real hardware is called the host. It is characterized by the presence of powerful processors, disk and RAM, etc. Installation of special software (software) allows you to divide server resources into several virtual computers.
* It is accepted to divide hypervisors into two types - "Type 1" and "Type 2". Despite this, there is a third variety (hybrid), which combines the properties of both species.

**Hypervisor "Type 1"**

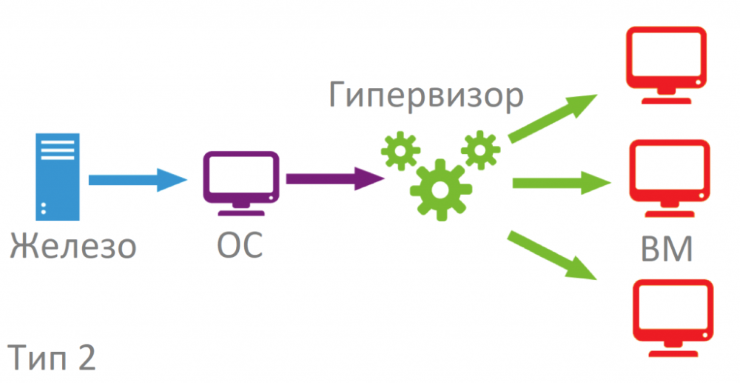


Guest operating systems that use the first type of hypervisor and function normally on it get the opportunity to run virtual machine services. This creates the illusion that with such an OS you can fully control all computer resources. By analogy, if a physical machine with supervisor mode access were used.

The first type (hardware hypervisor) has an important feature – its code takes up hundreds of times less space than modern OSes. It follows that the possibility of a critical error is reduced several times. The security of the system, which is responsible for virtualization, is achieved thanks to the complete transfer of the resources of the physical machine under the control of the virtual machine.

Hypervisors of the first type are VMware ESXi, Xen and others.

**Hypervisor "Type 2"**



It has another name - host. Such technology is installed as a "second layer" on top of the installed operating system. In essence, this type is an application of the main operating system. Linux is often chosen for stable operation. All physical resources and emulation are managed from the host operating system. The second type of hypervisor has less authority. These include KVM, Oracle VM VirtualBox and others.

KVM

Hypervisor kvm stands for Kernel-based Virtual Machine. It was created in 2006. Almost immediately, it was used in the Linux 2.6.20 kernel. The kvm.ko module is responsible for virtualization. KVM is open source. It contains QEMU components for working in user mode and modules for Intel, AMD.

Oracle VM VirtualBox

The hypervisor can be used on any operating system. It was created in 2007 by specialists from Sun Microsystems. Like many similar products, it is distinguished by its open source code, which allows you to modify it an unlimited number of times. If VirtualBox is installed on a 32-bit OS, it is able to work with 64-bit guest systems.

**2.2. List the main components and capabilities of hypervisors according to your option (serial number according to the journal), table 1.**

**Brytyuk Bohdan – 1 version**

Oracle VM VirtualBox is a modular, cross-platform hypervisor for Linux, macOS, Microsoft Windows, FreeBSD, Solaris/OpenSolaris, ReactOS, DOS and other operating systems from Oracle Corporation. It was created in 2007 at Sun Microsystems Corporation, then, after its takeover by Oracle, work on the hypervisor continued. The source code of the base version is open under the GNU GPL, so the hypervisor is popular and available for unlimited modification. Interestingly, VirtualBox is able to support 64-bit guest systems even if the host OS is 32-bit.

VirtualBox is a type 2 hypervisor that is sometimes called a hosted hypervisor. A type 2 hypervisor is an application that runs on the operating system (OS) and is already installed on a host. When a physical computer starts, the operating system installed on the host loads and takes control. A user starts the hypervisor application (VirtualBox in this case) and then starts the needed virtual machines. VM hosted processes are created.

Virtual Box can run on a higher number of operating systems such as Linux, Windows, Solaris, and Mac OS. If Windows is used on the physical machines in your environment, you could preference Hyper-V. If your environment is multiplatform, then you can take advantage of VirtualBox and run your virtual machines on different computers with different operating systems.

VirtualBox supports virtual disks of the following formats: VDI, VMDK, VHD, HDD.

* VDI is the native VirtualBox format
* VMDK is VMware virtual disk format
* VHD is Hyper-V format (VHDX is not supported by VirtualBox)
* HDD is Parallels Desktop format

You can use fixed size disks and dynamically allocated disks in VirtualBox, similarly to as you can in Hyper-V.

You can convert fixed disks to dynamic disks for both Hyper-V and VirtualBox. Fixed virtual disks need more time for creation and consume more storage space right after creation, but then provide higher performance. Dynamic virtual disks are created near instantly and allow you to save storage space, but their performance is lower than the performance of fixed virtual disks.

VirtualBox includes a built-in Shared Folders feature that can be easily enabled from the VirtualBox user interface. Transferring files between VirtualBox host machine and guest machine by using Shared Folders is convenient. You should open VM Settings, and then select Shared Folders. Add Share: set a folder path (the folder is located on the host machine, for example, C:\temp), enter a folder name that would be displayed in the guest OS. If needed enable or disable the following options: Read-only, Auto Mount, and Make Permanent by checking the boxes. Notice that VirtualBox Guest Additions must be installed on a guest OS.

Sichkar Maxim – 23 version

**Progress:**

**The student prepared the material:**  Brytyuk Bohdan, Sichkar Maxim

***1. Watch introductory videos and demonstration materials from the following areas:***

*1.1. GNU/Linux. Basic information.*

*Access: https://www.youtube.com/watch?v=k4AKMLS2Ac8*

*1.2. Installing CentOS in VirtualBox.*

*Access: https://www.youtube.com/watch?v=W3XTYYoHe9A*

*1.3. Installing CentOS in text mode.*

*Access: https://www.youtube.com/watch?v=gOR-1o3K18Q*

*1.4. Installing the Gnome desktop environment on CentOS.*

*Access: https://www.youtube.com/watch?v=gcEiIH3KF4Y*

*1.5. Installing the KDE desktop environment on CentOS.*

*Access: https://www.youtube.com/watch?v=\_ruIWLExaOY*

*1.6. The Shell (Linux)*

*Access: https://drive.google.com/open?id=0B0PV0\_SM0LoDSVNPWUVRdUxaN2s*

*1.7. An overview of Linux graphical shells*

*Access:* [*https://www.youtube.com/watch?v=lEGplwLXZ78*](https://www.youtube.com/watch?v=lEGplwLXZ78)

**The student prepared the material:**  Brytyuk Bohdan, Sichkar Maxim

***2. After watching the video, answer the following questions.***

2.1. List the steps for deploying an operating system based on a VirtualBox virtual machine.

2.2. Are there any hardware limitations when installing 32-bit and 64-bit OS?

2.3. What are the main steps when installing CentOS in text mode?

2.4. How can you install Gnome and KDE graphical shells on CentOS, if it is already there

installed in text mode (specify necessary commands and packages)?

2.5. Give a brief description of the graphical interfaces used in different

Linux distributions according to their variant (serial number according to the magazine), table 2.

**Brytyuk Bohdan – 1 version**

KDE stands for K Desktop Environment. It is a desktop environment for Linux based operation system. You can think KDE as a GUI for Linux OS. KDE has proved Linux users to make it use as easy as they use windows. KDE provides Linux users a graphical interface to choose their own customized desktop environment. You can choose your Graphical Interface among various available GUI interfaces that have their own look.

You can imagine Linux without KDE and GNOME just like DOS in windows. KDE and GNOME are much similar with Windows except they are related to Linux through x server rather then operation system. When you install Linux you have a choice to choose your own desktop environment from two or three different desktop environments like KDE and GNOME. Another popular environment same as KDE is GNOME. Both come with variety of features with different distributions. KDE comes with variety of features some of the main among them are listed below:

## **What is Fluxbox?**

[Fluxbox](http://www.fluxbox.org/) is a fast, lightweight and responsive window manager for GNU/Linux. It is not nearly as elaborate as GNOME or KDE, but it uses considerably fewer system resources. This makes it uniquely suited for situations where system resources, especially RAM, are very limited. Its menu and configuration is done by simple files located in the user directory under the name ~/.fluxbox.

**Fluxbox** is for the X Window System based on Blackbox and compatible with it. Fluxbox looks like blackbox and handles styles, colors, window placement and similar things exactly like blackbox. It has support for KDE, [Xfce](http://www.xfce.org/) and [Gnome](http://www.gnome.org/) applications. However, it does not depend on any other window manager.

In accordance with Fluxbox's goal of simplicity, the main menu, the keyboard shortcuts and the basic configuration are all changed by editing text files. Fluxbox's themes are 100% compatible with those of blackbox. Colors, gradients, borders, and several other basic appearance attributes can be specified; recent versions of Fluxbox support rounded corners and graphical elements. Fluxbox also has several features which blackbox lack, including tabbed windows, a feature familiar from PWM, and configurable titlebar.

# Features

Fluxbox has been made to be very light on the resources with a basic interface having only a taskbar and a menu (Root Meun) accessible by right-clicking on the desktop. Customization is important to with Fluxbox, but you will not see over bloated dialog boxes, but simple text files, allowing you to change Fluxbox to suit your preferences.

## **Root Menu**

Unlike GNOME, KDE, and XFCE, Fluxbox has no "start" button. To get to the menu, simply right-click anywhere on the desktop. Submenus will expand if you simply mouse over them. To run a program, highlight its entry in the menu and click on it.

## **Workspace**

Fluxbox has virtual desktops called Workspaces. These allow you to keep projects separate, not cluttering up the desktop with extraneousness windows. For example you can set your music player to another workspace, getting its window out of your way. Turning the mousehweel over any unoccupied space on the desktop will page through the available workspaces. This can be a quick way of flipping from one workspace to the next.

## **Tabbed Windows**

The tabbing mechanism for windows is an unique feature of Fluxbox. You can combine multiple windows into one window with tabs across the top. A control-click on the tab of one windows starts the tab feature, allowing you to then drag it onto the tab of another window. The two windows will appear to merge into a single window, with two title tabs. You can now view each tab within the newly-joined window by clicking on its respective title bar.

Window tabbing is a good way of conserving screen real-estate and reducing clutter. This is particularly useful if you are running an application from a terminal. By tabbing the application with its corresponding terminal window, it is easy to flip back and forth from the debug output in the terminal to the application.

## **Dock Apps**

Another nice feature is Fluxbox's support for docking applications (dockapps). Basically a dockapp runs as sort of an icon with minature display or controls. But not like an icon, more like small controls on a walkman (for example) as opposed to a big dial face of a home stereo. It aims to be lightweight and highly customizable, with only minimal support for graphical icons, and only basic interface style capabilities.

Sichkar Maxim – 23 version

**The student prepared the material:**  Brytyuk Bohdan, Sichkar Maxim

**Answers to control questions:**

1. Compare type 1 and type 2 hypervisors, what is the difference between them and their scope?

2. Explain the concept of "GNU GPL", what is its basic concept?

3. What is open source software?

4. What is a distribution?

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

6. How are Android OS and Linux related?

7. Main capabilities and scope of use of Embedded Linux?

8. How can you change the type of Linux boot: in text mode (level 3) or graphical (level 5)? What is the difference between CLI and GUI modes?

***Conclusion:***