“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №1**

з дисципліни: «Операційні системи»

**Тема: «Ознайомлення з робочим середовищем віртуальних машин та операційних систем різних сімейств»**

Виконав студент

групи КСМ-03Б

Федоренко Артем Олександрович

Перевірив викладач

Сушанова В.С.

Київ 2022

**Мета роботи:**

1. Отримання практичних навиків роботи з середовищами віртуальних машин та операційними системами різних типів та сімейств – їх графічною оболонкою, входом і виходом з системи, ознайомлення зі структурою робочого столу, вивчення основних дій та налаштувань при роботі в системі.

**Матеріальне забезпечення занять**

1. ЕОМ типу IBM PC.

2. ОС сімейства Windows (Windows 10).

3. Віртуальна машина – Virtual Box (Oracle).

4. Операційна система GNU/Linux – CentOS.

**Завдання для попередньої підготовки**

1. Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеличкий словник базових англійських термінів з питань класифікації ОС.

|  |  |
| --- | --- |
| Термін англійською | Термін українською |
| **Operating System** | Операційна система |
| **Shared hosting** | Віртуальний хостинг |
| **Type 1 hypervisor** | Гіпервізор 1 типу |
| **Machine simulators** | Машинні симулятори |
| **Binary translation** | Двійковий переклад |
| **Type 2 hypervisors** | Гіпервізори 2 типу |
| **Host operating system** | Операційна система хоста |
| **Guest operating system** | Гостьова операційна система |
| **Graphical user interface** | Графічний інтерфейс користувача |
| **Сommand line interface** | Інтерфейс командного рядка |

1. Прочитавши матеріал з коротких теоретичних відомостей дайте відповіді на наступні питання:

2.1 A hypervisor is a form of virtualization software used in Cloud hosting to divide and allocate the resources on various pieces of hardware. The program which provides partitioning, isolation, or abstraction is called a virtualization hypervisor. The hypervisor is a hardware virtualization technique that allows multiple guest operating systems (OS) to run on a single host system at the same time. A hypervisor is sometimes also called a virtual machine manager (VMM). There are two types of hypervisors:

• TYPE-1 — The hypervisor runs directly on the underlying host system. It is also known as a “Native Hypervisor” or “Bare metal hypervisor”. It does not require any base server operating system. It has direct access to hardware resources.

• TYPE-2 — A Host operating system runs on the underlying host system. It is also known as ‘Hosted Hypervisor”. Such kind of hypervisors doesn’t run directly over the underlying hardware rather they run as an application in a Host system (physical machine). Basically, the software is installed on an operating system.

2.2

**VirtualBox** is a cross-platform virtualization application. What does that mean? For one thing, it installs on your existing Intel or AMD-based computers, whether they are running Windows, Mac OS X, Linux, or Oracle Solaris operating systems (OSes). Secondly, it extends the capabilities of your existing computer so that it can run multiple OSes, inside multiple virtual machines, at the same time. As an example, you can run Windows and Linux on your Mac, run Windows Server 2016 on your Linux server, run Linux on your Windows PC, and so on, all alongside your existing applications. You can install and run as many virtual machines as you like. The only practical limits are disk space and memory.

VirtualBox is deceptively simple yet also very powerful. It can run everywhere from small embedded systems or desktop class machines all the way up to datacenter deployments and even Cloud environments.

**The techniques and features that Oracle VM VirtualBox provides are useful in the following scenarios:**

* **Running multiple operating systems simultaneously.** VirtualBox enables you to run more than one OS at a time. This way, you can run software written for one OS on another, such as Windows software on Linux or a Mac, without having to reboot to use it. Since you can configure what kinds of virtual hardware should be presented to each such OS, you can install an old OS such as DOS or OS/2 even if your real computer's hardware is no longer supported by that OS.
* **Easier software installations.** Software vendors can use virtual machines to ship entire software configurations. For example, installing a complete mail server solution on a real machine can be a tedious task. With VirtualBox, such a complex setup, often called an appliance, can be packed into a virtual machine. Installing and running a mail server becomes as easy as importing such an appliance into VirtualBox.
* **Testing and disaster recovery.** Once installed, a virtual machine and its virtual hard disks can be considered a container that ca be arbitrarily frozen, woken up, copied, backed up, and transported between hosts.
* **Infrastructure consolidation.** Virtualization can significantly reduce hardware and electricity costs. Most of the time, computers today only use a fraction of their potential power and run with low average system loads. A lot of hardware resources as well as electricity is thereby wasted. So, instead of running many such physical computers that are only partially used, one can pack many virtual machines onto a few powerful hosts and balance the loads between them.

A **hypervisor** is an important piece of software that makes virtualization possible. It abstracts the guest machines and the OS they run on from the real hardware. Hypervisors create a virtualization layer that separates CPU/processors, RAM, and other physical resources from the virtual machines you create.

**KVM**is built into Linux as an optional feature. It allows you to turn the Linux kernel into a hypervisor. It has direct access to the hardware as well as the virtual machines it hosts. KVM is an open source hypervisor that contains all the features of Linux with many more added. This makes it one of the ***best choices*** for enterprise environments. Some of the highlights include dynamic migration, resource scheduling and control, and higher priorities.

For some time now, KVM has been part of the Linux kernel, so it evolves along with it. Works only in systems ***with hardware support for virtualization*** - on Intel and AMD processors.

***Advantages and disadvantages of KVM***

***The main advantages of a hypervisor are:***

***Independently distributed resources.*** Each virtual machine running under the control of KVM receives its own amount of RAM and permanent memory and cannot "climb" into other areas, which increases the stability of work;

***Broad guest OS support***. In addition to full support for UNIX distributions, including \*BSD, Solaris, Linux, it is possible to install Windows and even MacOS;

interaction with the kernel allows you to directly access the workstation hardware, which makes the work ***faster;***

***support of the giants of the software market*** allows the project to develop rapidly, covering an increasing number of hardware and OS, including the latest;

***simple administration*** - the ability to remotely manage via VNC and a large number of third-party software and add-ons.

***There were also some drawbacks:***

the relative youth of the hypervisor and the corresponding explosive growth lead to various problems, especially when adding support for new hardware and software environments;

the complexity of the settings, especially for an inexperienced user. True, most of the options can not be changed - they are configured optimally.

***Safety***

In KVM, each machine is a Linux process, so standard security policies are automatically applied to it, as well as isolation from other processes.

***Performance and scalability***

Scalability and performance of the complex due to tight integration with Linux. Thus, the hypervisor supports up to 16 processors and up to 256 GB of RAM in each virtual machine.

***Stability***

The software package is constantly being improved - if initially it supported only the Linux x86 platform, today the number of different platforms is in the tens. And thanks to cooperation with leading software manufacturers, the KVM hypervisor can be called the most stable and reliable on the market.

***XEN hypervisor***

XEN is a hypervisor that enables the simultaneous creation, execution and management of multiple virtual machines on one physical computer. Xen is primarily a bare-metal, type-1 hypervisor that can be directly installed on computer hardware without the need for a host operating system. Because it's a type-1 hypervisor, Xen controls, monitors and manages the hardware, peripheral and I/O resources directly. Xen supports multiple instances of the same or different operating systems with native support for most operating systems, including Windows and Linux. Moreover, Xen can be used on x86, IA-32 and ARM processor architecture.

XEN has two the most significant features: paravirtualization and the minimum code of the hypervisor itself. The basic principle of paravirtualization is to prepare guest operating systems by slightly modifying their kernel before running in a virtualized environment. The second factor that relates to the features of XEN is the insignificant amount of code of the hypervisor itself. This is achieved by moving more management functions out of the hypervisor itself.

The Xen hypervisor allows a company to consolidate multiple virtual machines into one hardware platform. It also allows you to share resources so that instead of having say 6 physical systems that each need 20% free CPU to handle spikes, the 6 virtual machines can run closer to the capacity of the larger host system since they won’t all be experiencing CPU spikes at the same time. If you have some spare capacity you could even spin up additional machines during peak times. The virtual machines can be more easily monitored and controlled using XenCenter. As for the disadvantages, you users might have have to pay more attention to capacity planning to avoid performance issues on the bank of virtual systems.

**Хід роботи**

1. Подивіться ознайомчі відео та демонстраційні матеріали з наступних напрямків:

1.1 GNU/Linux. Базові відомості.

1.2 Встановлення CentOS у VirtualBox.

1.3 Встановлення CentOS в текстовому режимі.

1.4 Встановлення оточення робочого столу Gnome в CentOS.

1.5 Встановлення оточення робочого столу KDE в CentOS.

1.6 The Shell (Linux)

1.7 Огляд графічних оболонок Linux

2. Після перегляду відео дайте відповіді на питання:

2.1 Creating a virtual machine for Windows is very simple. After installation, launch VirtualBox, press the "Create" button and answer the wizard's questions:

1. specify the name, family and specific operating system that will be installed on this machine (for example, Windows XP), press next;
2. then choose the amount of RAM of the future virtual computer;
3. at the next stage, we create a virtual hard disk - select the "boot disk" option, specify its size in GB (for Windows XP at least 7 GB), file type - VDI. If the size of the real hard disk allows, create a fixed virtual disk, if not, select the "dynamic virtual disk" option, specify the storage location of the virtual disk file (a specific logical partition of the real hard disk), press the "create" button and wait for the process to complete.
4. After creating a virtual hard disk, a new virtual machine will appear in the main window of the VirtualBox program.
5. We select it, press the "Properties" button and get to the section, which is a kind of BIOS of the virtual machine. Here you can change the parameters, turn on and off the various devices included in it. The default setting is a universal option, so we will not change anything here. You only need to go to the "Media" section and specify the source (presence) of the operating system installation. In the "Media" section, select the device with the image of the disk and the inscription "Empty", then in the right part of the VirtualBox window, specify the real optical drive (if the installation will be carried out from the real installation disk located in this drive) or the image of the optical disk (if the installation disk is saved on the hard disk in the form of an image), press the "OK" button.
6. After that, we return to the main window of the VirtualBox program and press the "Start" button. A virtual machine will start, on which the operating system will be installed from the source (real or virtual drive) specified at the previous stage.

2.2 32-bit systems, limited to 3.2 GB of RAM, 32-bit Windows. It refers to restrictions that do not allow the use of 4 GB of physical memory. 64-bit systems will allow you to store up to 17 billion GB of RAM.

2.3

1. After booting from the netinstall ISO file, you will see the first welcome screen with various drop-down menus. Press ESC at this point.
2. Then you will see the blank screen with boot: prompt as shown below. Type **linux text** at the prompt and enter. The rest of the installation steps will proceed in text mode.
3. If you are using LiveCD of CentOS, installing CentOS in text mode is straightforward. At the initial welcome screen, simply choose Install (Text Mode).

2.4

1. Update your system. Type in the command line “yum update”.
2. Install GNOME packeges. Command “sudo yum -y groups install "GNOME Desktop"”.
3. Set up the Windows X System by typing “echo "exec gnome-session" >> ~/.xinitrc”
4. Test Windows X System with command
5. Now we need to make it start automatically: systemctl set-default graphical.target

2.5

***XFCE***

***XFCE*** gives you a stable desktop experience, primarily due to its three-year LTS release cycle. Periodic updates are also provided in the interim to keep you updated with the latest innovations in the environment. The frequent releases are an admirable perk since XFCE’s community and support contribution is not as per KDE’s scale.

XFCE uses the GTK toolkit and continues to be an independent alternative to the GTK-built GNOME environment. It has been in the game for over 25 years and offers you more reliability for periodic and stable releases.

XFCE uses reusable components so that you can pick and choose what you need to drive stable performance on your existing hardware.

XFCE offers you a lean and responsive front-end experience as a desktop environment, regardless of the host distribution. Even if you are running XFCE on a system with low hardware, you can expect a decent performance from the desktop environment. If agile performance is what you are after, you might be biased about XFCE. Having said this, it has veritably less RAM usage than KDE plasma, using only 531MB idly, regardless of VM execution or native installation.

XFCE defines productivity and performance with its bare minimalism. As highlighted before, XFCE allows you to pick and choose components on your use case basis.

The environment is hardware friendly and works well on low-end system configurations without compromising the quality and delivery of user experiences.

***FVWM***

FVWM is an ICCCM-compliant multiple virtual desktop window manager for the X Window system. It is configured by editing text-based configuration files. Although using FVWM does not require any knowledge of programming languages, it is possible to extend FVWM with M4, C, and Perl preprocessing. FVWM also has a Perl library which allows one to create modules. FVWM stands for F Virtual Window Manager with the preferred interpretation being that the F does not stand for anything in particular

FVWM provides a number of functions to start modules or applications when initialising, restarting or exiting the window manager.

Menus in FVWM can be dynamic, meaning that their content is refreshed every time the menu is opened. This can be useful when using a menu generator, such as one that constructs a menu of applications from XDG desktop entries, and you want the content to always be up to date.

Dynamic menus can be created in FVWM by using the *DynamicPopUpAction* and *DynamicPopDownAction* keywords. The former can be used to create/recreate the menu when it is opened whilst the latter can be used to clear the menu when it is closed. It is important to note that for a submenu to be dynamic, the parent menu that it is included in must also be created dynamically even if it does not contain any dynamic content.

***KDE*** is an international free software development community. As a central development hub, it provides the tools and resources that enable collaborative work on this kind of software. Notable products include KDE Plasma 5, KDE Frameworks, and many cross-platform applications such as Krita or digiKam designed to run on Unix and Unix-like desktops, and on Android.

Plasma is the default desktop environment on many Linux distributions, such as openSUSE, Manjaro Linux, Kubuntu, PCLinuxOS. KDE is a working shell with an integrated set of cross-platform applications for Linux, Windows, Solaris and Mac. In many ways, KDE (K Desktop Environment) is similar to Windows, so users feel at home when using it. As in Win OS, they access the menu by clicking on the left side of the panel. The software has one menu bar at the bottom of the screen, it can be changed at the user's request. With plasma desktops, KDE is the most attractive of all Linux desktops. Although KDE is sleeker in appearance and has more options than GNOME 2.x, XFCE, or LXDE, it is also more resource intensive. On the other hand, it requires less system resources than Unity and less RAM than GNOME

***Fluxbox*** is a simple and minimalist window manager for the X Window System. Fluxbox is based on Blackbox v.0.61.1 and is similar to it - same colors, window layout and full compatibility of themes and styles. It has minimal requirements for system resources and provides all the basic possibilities for organizing comfortable work. Fluxbox supports such functions as combining several windows in the form of tabs, editing menus, collapsing applications to the system tray, virtual desktops, flexible tools for grouping windows, customizing the appearance, binding hot keys. The fluxbox code is written in C++ and distributed under the MIT license.

On the outside, Fluxbox is a clean desktop with a toolbar containing the desktop title, clock, and list of running applications. By default, Fluxbox does not have its own icons on the desktop, but you can add them using idesk.

Programs are launched by selecting the appropriate item from the drop-down menu that appears after clicking the right mouse button anywhere on the desktop, or by typing the name of the program in the pop-up input panel, which is called either from the above-mentioned drop-down menu, or by pressing hot keys.

***GNOME GUI***

GNOME ([GNU](https://www.techtarget.com/searchdatacenter/definition/GNU-Linux) Network Object Model Environment, pronounced gah-NOHM) is a graphical user interface and set of computer [desktop](https://www.techtarget.com/searchenterprisedesktop/definition/desktop) [applications](https://www.techtarget.com/searchsoftwarequality/definition/application) for users of the [Linux operating system](https://www.techtarget.com/searchdatacenter/definition/Linux-operating-system).

• **Minimalist Distraction-Free Design**

On GNOME, the panel at the top does not contain any app launchers. This panel is small and black, like on a phone or tablet, and is lately static. It contains the date and time, a few system indicators in the top right, an Activities button in the top left, and the name of the currently running app next to that.

• **Focused, Consistent, and Intuitive Apps**

GNOME’s minimalist design extends from the desktop to the apps themselves. You don’t have various menus to navigate or numerous preferences to toggle. Most features are often found right at the top of the app in what’s called the headerbar.

• **Desktop, Tablet, and Mobile-Friendly**

These GNOME apps not only fit well on your desktop, but if you shrink them down, you find that most now adjust to fit a mobile device as well. Adaptive design is great on desktops because that means you can tuck an app away at the side of your screen and find that the interface is still usable.

• **Support for the Latest Technologies**

There’s an audio server managing sound. There are package formats that apps come in. On Linux, there are multiple versions of each of these system components, with newer ones hitting the scenes every few years. These system components are agnostic about which desktop environment you use, but GNOME is often the first or among the first to integrate these new technologies.

**• All the Programs Needed to Do the Essentials**

The GNOME Project has been around for decades, and in that time a sizable number of programs have popped into existence and matured. Everything is in place for a fully-functional desktop operating system.

***JWM GUI***

• JWM Kit is a new set of software design to simplify usage of JWM (Joe’s Window Manager). In order to maximize personal customization JWM Kit focus on providing a graphical interface for editing menus, trays, groups, etc. No part of the kit is designed to run in the background. It only runs when the user starts it, and properly quits when “closed”, or for notifications will close when timed out in a few seconds.

• The quick and easy setup of a taskbar and application menu makes for a quick, sensible default configuration for all the users who want that common setup. For technical users, the XML configuration may be a refreshing change from the custom configuration formats in other environments. Joe's Window Manager, in a way, demonstrates the rich diversity of open source

**Відповіді на контрольні запитання**

1. ***Type 1 hypervisors*** themselves can also be forgiven by the OS, on top of which you can run virtual machines. The physical machine, depracing the hypervisor, will serve only as a method of virtualization. You can't win for anything else. Type 1 hypervisors are more important than ever in corporate environments. ***Type 2 hypervisor*** is placed in the middle of the operating system of the physical host computer. At the top of the type 1 hypervisors, they work without intermediary on the owner, the location of the hypervisors and may have one program run. Type 2 hypervisors ring vicorist in the middle with a small number of servers.

2. The ***GNU General Public License*** is one of the most popular free software licenses. created by Richard Stallman for the GNU project. It is often abbreviated as GNU GPL or simply GPL. The purpose of the GNU GPL is to give a user the right to copy, modify, and distribute a program and an obligation that users of all programs derived from it will also receive those rights. The principle of "inheritance" of such rights is called "copyleft", a term proposed by Richard Stallman. Unlike the GPL, proprietary software licenses very rarely grant the user such rights, and mostly try to limit them, for example, by prohibiting the restoration of the source code.

3. Open source software (OSS) is a type of software in which the source code is released under a license in which the copyright holder grants users the right to use, study, modify and distribute the software to anyone and for any purpose. Open source software is a prime example of open collaboration.

Open source software development can offer different perspectives beyond that of a single company. A 2008 report by the Standish Group stated that the adoption of open source software models has resulted in savings for consumers of approximately $60 billion

4. A distribution is a set of files that are required to install utilities or software packages. An example of a distribution can be an installation disk with an operating system.

5. The main tasks of a system administrator (superuser) in Linux include:

installation (installation) of the OS; management of the OS loading process; setting operating modes of the OS; editing configuration files; mounting and dismounting of file systems; introduction and removal of OS users; software updates; OS kernel configuration; ensuring the reliable functioning of the OS; computer network configuration.

6. Android OS is a Linux-based mobile operating system that primarily runs on smartphones and tablets. The Android platform includes an operating system based on the Linux kernel, a GUI, a web browser and end-user applications that can be downloaded.

7. The term "embedded Linux®" includes the description of any variant of the open source Linux® operating system running on an embedded computer system, a specialized device or platform integrated into an overall larger product, such as a device or consumer electronics item . equipment.

8. They mainly differ in the graphics used in the operating system. To perform an operation on the CLI system, you need to write a command. On the other hand, GUI users are provided with visual aids (graphics) that include images and icons, making it easier for users to perform a task directly.

CLI systems require knowledge of commands to perform tasks, while GUI does not require knowledge, it can also work with beginners.

**Висновки**

В ході виконання лабораторної роботи нами було досліджено гіпервізори: VirtualBox, XEN, KVM. Також дослідили графічні інтерфейси Gnome, KDE, Fluxbox, XFCE та Fvwm. Більш детально теоретично досліджено питання налаштувань і можливостей ОС Linux. Отримано практичні навики роботи з командами Linux, налаштуваннями графічних інтерфейсів та гіпервізорів.