"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 |

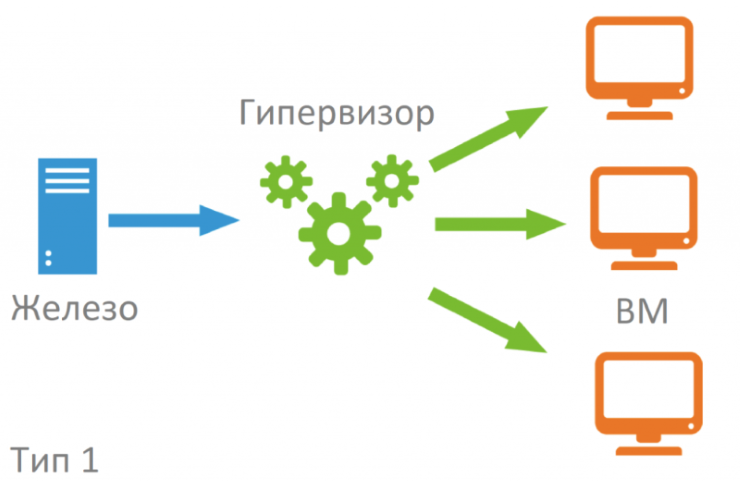
2. After reading the material from short theoretical information, answer the following questions:

**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"**

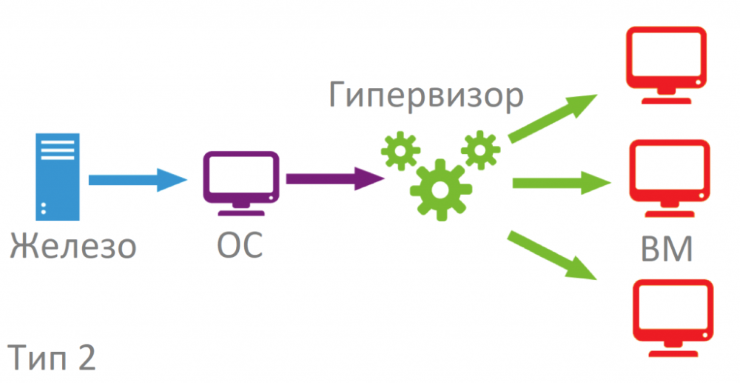


Guest operating systems that use the first type of hypervisor and function normally on it get the opportunity to run virtual machine services. This creates the illusion that with such an OS you can fully control all computer resources. By analogy, if a physical machine with supervisor mode access were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

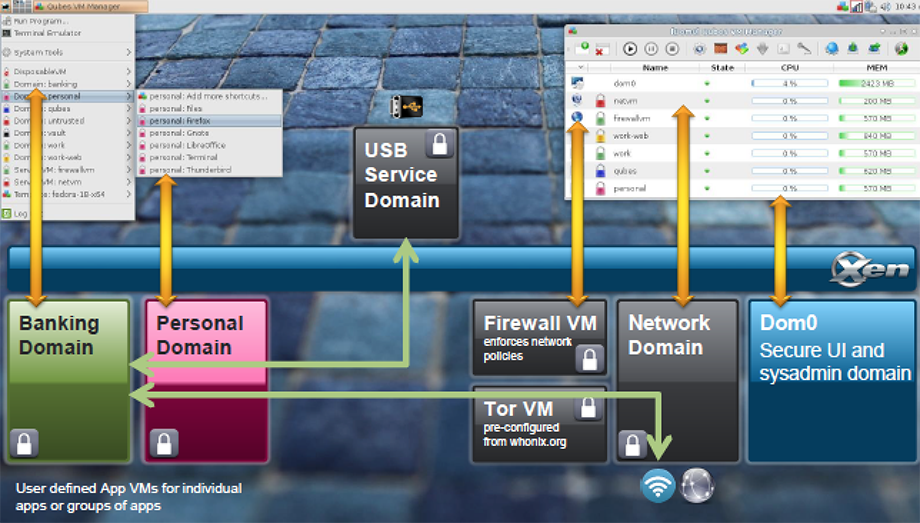
You can use fixed size disks and dynamically allocated disks in VirtualBox, similarly to as you can in Hyper-V.

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**

Xen - is a cross-platform hypervisor developed in the computer laboratory of the University of Cambridge and distributed under the terms of the GPL license. Main features: support for the paravirtualization mode in addition to hardware virtualization, the minimality of the hypervisor code due to the removal of the maximum number of components outside the hypervisor.



Xen components used by the Qubes operating system are superimposed on top of the user interface: hypervisor, Dom0, network domain (Network Domain), firewall domain (Firewall Domain), USB domain.

The technology of virtual machines allows you to expand the functionality of the equipment in the following ways:

A virtual machine has performance comparable to a real one.

Ability to migrate a running virtual machine between physical machines.

Good hardware support (most Linux device drivers are supported)

Ability to create a sandbox, reloadable device drivers.

**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:** 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.

VirtualBox supports a long list of host and guest operating systems. A host OS is the operating system installed on a physical machine, on which VirtualBox is installed. A guest OS is an operating system installed on a virtual machine running inside VirtualBox. VirtualBox can be installed on Windows, Linux, macOS, Solaris, and FreeBSD. On VirtualBox you can run VMs with Windows, Linux, macOS, Solaris, FreeBSD, Novell Netware, and other operating systems.

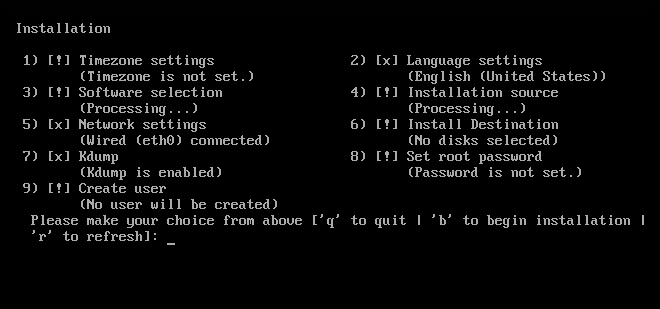
First, you need to enable hardware virtualization features on your CPU (Central Processor Unit) such as Intel VT-X or AMD-V in UEFI/BIOS of your physical computer. Otherwise, if you run 64-bit guest operating systems, you can get the error: VT-x is not available. If Hyper-V is installed on your Windows machine, uninstall Hyper-V before installing VirtualBox (otherwise Hyper-V will block hardware virtualization extensions needed by VirtualBox to run VMs). The majority of modern processors support hardware virtualization.

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

Yes. 64-bit drivers must be used to ensure that all hardware devices work on a 64-bit version of Windows.

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

Text mode installation offers an interactive, non-graphical interface for installing CentOS. This can be useful on systems with no graphical capabilities; however, you should always consider the available alternatives before starting a text-based installation. Text mode is limited in the amount of choices you can make during the installation.



Installations using the graphical interface are preferred. If you are installing CentOS on a system that lacks a graphical display, consider performing the installation over a VNC connection - see Using VNC. The text mode installation program will prompt you to confirm the use of text mode if it detects that a VNC-based installation is possible.

If your system has a graphical display, but graphical installation fails, try booting with the inst.xdriver=vesa option - see Boot Options.

Alternatively, consider a Kickstart installation. See Kickstart Installations for more information.

The main menu in during a text-based installation.

Figure 1. Text Mode Installation

Installation in text mode follows a pattern similar to the graphical installation: There is no single fixed progression; you can configure many settings in any order you want using the main status screen. Screens which have already been configured, either automatically or by you, are marked as [x], and screens which require your attention before the installation can begin are marked with [!]. Available commands are displayed below the list of available options.

When related background tasks are being run, certain menu items can be temporarily unavailable or display the Processing…​ label. To refresh to the current status of text menu items, use the r option at the text mode prompt.

At the bottom of the screen in text mode, a green bar is displayed showing five menu options. These options represent different screens in the tmux terminal multiplexer; by default you start in screen 1, and you can use keyboard shortcuts to switch to other screens which contain logs and an interactive command prompt. For information about available screens and shortcuts to switch to them, see Accessing Consoles (x86).

Limits of interactive text mode installation include:

The installer will always use the English language and the US English keyboard layout. You can configure your language and keyboard settings, but these settings will only apply to the installed system, not to the installation.

You cannot configure any advanced storage methods (LVM, software RAID, FCoE, zFCP and iSCSI).

It is not possible to configure custom partitioning; you must use one of the automatic partitioning settings. You also cannot configure where the boot loader will be installed.

You cannot select any package add-ons to be installed; they must be added after the installation finishes using the Yum package manager.

To start a text mode installation, boot the installation with the inst.text boot option used either at the boot command line in the boot menu, or in your PXE server configuration. See Booting the Installation on 64-bit AMD, Intel, and ARM systems for information about booting and using boot options.

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)?

1. Install the GNOME Desktop package group with yum groupinstall:
2. yum groupinstall "GNOME Desktop"

Copy

1. Manually start the desktop environment using startx:
2. startx

Copy

1. If more than one environment is installed, you'll need to specify the path to gnome-session:
2. startx /usr/bin/gnome-session

Copy

1. When you're done using GNOME and log out of the desktop, you'll be returned to the console.
2. To configure the system to automatically start the graphical environment when it boots, set the default start up target to graphical.target:

systemctl set-default graphical.target

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 1.

**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**

GTK+ is the interface which GNOME applications use to interact with the user. It is similar to the role of Motif in CDE, Qt in KDE or the-widget-set-which-has-no-name in Win32. JWM is a free window manager for the X Window System written by Joe Wingbermuehle. The author distributes only in the form of source code, compiles and runs on any UNIX system running X11.

In the minimal version, only the Xlib library is required. It is possible to use smoothed and TrueType fonts, icons in PNG and JPEG formats, and bidirectional writing if the appropriate libraries are available, which must be connected at the compilation stage. Supports GNOME, Motif, and WM-Spec[2] specifications for window managers.

Ported to many operating systems, including FreeBSD[3], NetBSD[4], OpenBSD, Minix[5], OpenSolaris (used by default in the MilaX distribution based on it[6]) and used as the default window manager in some versions of Linux distributions: Damn Small Linux[7], Tiny Core Linux (up to version 2.0), Puppy Linux[8], SystemRescueCD[9].

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

**Answers to control questions: Prepeared by Brytiuk Boghdan**

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

The main difference between Type 1 vs. Type 2 hypervisors is that Type 1 runs on bare metal and Type 2 runs on top of an operating system. Each hypervisor type also has its own pros and cons and specific use cases.

Virtualization works by [abstracting physical hardware](https://searchservervirtualization.techtarget.com/feature/4-types-of-virtualization-IT-admins-should-know) and devices from the applications running on that hardware. The process of virtualization manages and provisions the system's resources, including processor, memory, storage and network resources. This enables the system to host more than one workload simultaneously, making more cost- and energy-efficient use of the available servers and systems across the organization.

Both Type 1 and Type 2 hypervisors use hardware acceleration support, but to varying degrees. Type 1 hypervisors rely on hardware acceleration technologies and typically don't function without those technologies available and enabled through the system's BIOS. Type 2 hypervisors are generally capable of using hardware acceleration technologies if those features are available, but they can typically fall back on software emulation in the absence of native hardware support.

When choosing between a [Type 1 and Type 2 hypervisor](https://www.idkrtm.com/history-of-virtualization/), admins must consider the type and size of their workloads. If admins primarily work in an enterprise or large organization and must deploy hundreds of VMs, a Type 1 hypervisor will suit their needs.

But if admins have a smaller deployment or require a testing environment, Type 2 hypervisors are less complex and have a smaller price tag. And enterprises and organizations can use Type 2 hypervisors as needed for workloads that suit the technology.

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

The **GNU General Public License** (**GNU GPL** or simply **GPL**) is a series of widely used [free software licenses](https://en.wikipedia.org/wiki/Free_software_license) that guarantee [end users](https://en.wikipedia.org/wiki/End_user) the [four freedoms](https://en.wikipedia.org/wiki/Four_Freedoms_(Free_software)) to run, study, share, and modify the software The license was the first [copyleft](https://en.wikipedia.org/wiki/Copyleft) for general use and was originally written by the founder of the [Free Software Foundation](https://en.wikipedia.org/wiki/Free_Software_Foundation) (FSF), [Richard Stallman](https://en.wikipedia.org/wiki/Richard_Stallman), for the [GNU Project](https://en.wikipedia.org/wiki/GNU_Project). The license grants the recipients of a [computer program](https://en.wikipedia.org/wiki/Computer_program) the rights of [the Free Software Definition](https://en.wikipedia.org/wiki/The_Free_Software_Definition). These GPL series are all copyleft licenses, which means that any [derivative work](https://en.wikipedia.org/wiki/Derivative_work) must be distributed under the same or equivalent license terms. It is more restrictive than the [Lesser General Public License](https://en.wikipedia.org/wiki/GNU_Lesser_General_Public_License) and even further distinct from the more widely used [permissive software licenses](https://en.wikipedia.org/wiki/Permissive_software_license) [BSD](https://en.wikipedia.org/wiki/BSD_licenses), [MIT](https://en.wikipedia.org/wiki/MIT_License), and [Apache](https://en.wikipedia.org/wiki/Apache_License).

Historically, the GPL license family has been one of the most popular software licenses in the [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) domain Prominent free software programs licensed under the GPL include the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) and the [GNU Compiler Collection](https://en.wikipedia.org/wiki/GNU_Compiler_Collection) (GCC). David A. Wheeler argues that the copyleft provided by the GPL was crucial to the success of [Linux](https://en.wikipedia.org/wiki/Linux)-based systems, giving the programmers who contributed to the kernel the assurance that their work would benefit the whole world and remain free, rather than being exploited by software companies that would not have to give anything back to the community

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

Open source is a term that originally referred to open source software (OSS). Open source software is code that is designed to be publicly accessible—anyone can see, modify, and distribute the code as they see fit.

Open source software is developed in a decentralized and collaborative way, relying on peer review and community production. Open source software is often cheaper, more flexible, and has more longevity than its proprietary peers because it is developed by communities rather than a single author or company.

Open source has become a movement and a way of working that reaches beyond software production. The open source movement uses the values and decentralized production model of open source software to find new ways to solve problems in their communities and industries.

**4. What is a distribution?**

Distribution (distro) is a process of delivering software from a developer to the end user. Software distribution ranges from OS server distribution to interpreter distribution.

Software is distributed in bundles containing required files, instructions, configuration settings and management settings used to deploy a system application.

There are four types of bundles, as follows:

* Directive bundle: Performs multiple system actions.
* File bundle: Copies or installs system files.
* Imaging bundle: Performs actions prior to an OS boot.
* Windows bundle: Created for distributing Microsoft Windows software patch (MSP) packages, Microsoft Windows Installer (MSI) packages or other Windows based applications to Windows systems.

A bundle is assigned to a system or users. When a bundle is assigned to a user, it is available for access by all users, regardless of user system. When a bundle is assigned to a system, it is accessible to all users that log into the assigned system.

The distribution phase is followed by unpacking and installation, and package management tools are available for software package installation.

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

## **Documentation**

Documentation is how sysadmins keep records of assets, including hardware and software types, counts, and licenses. Should there be any issues in the production environment, documentation helps identify the hardware, virtual machine, appliance, software, etc., that may be involved.

### **Hardware inventory**

Maintain lists of all your physical and virtual servers with the following details:

* **OS:** Linux or Windows, hypervisor with versions
* **RAM:** DIMM slots in physical servers
* **CPU:** Logical and virtual CPUs
* **HDD:** Type and size of hard disks
* **External storage (SAN/NAS):** Make and model of storage with management IP address and interface IP address
* **Open ports:** Ports opened at the server end for incoming traffic
* **IP address:** Management and interface IP address with VLANs
* **Engineering appliances:** e.g., Exalogic, PureApp, etc.

### **Software inventory**

* **Configured applications:** e.g., Oracle WebLogic, IBM WebSphere Application Server, Apache Tomcat, Red Hat JBoss, etc.
* **Third-party software:** Any software not shipped with the installed OS

### License details

Maintain license counts and details for physical servers and virtual servers (VMs), including licenses for Windows, subscriptions for Linux OS, and the license limit of hypervisor host.

## **Server health checkup**

* **Running processes:** Check for processes that are consuming more resources than expected, and take action to fine-tune the applications (with the help of the application team).
* **CPU utilization:** Consistently monitor and check the CPU utilization of the critical process like "java", "http", "mysql" etc. to ensure that these are not consuming the CPU resources more than expected. If it is so, then coordinate with the application team to check it at application level  and fine tune the same. Parallely analyse the OS parameters like "Ulimits".
* **Memory utilization:** Check memory utilization and clear the cache, if required.
* **Zombie processes:** Check for processes where the PID still exists in the process table after it is terminated. Zombie processes degrade server performance, so find and kill any that exist.
* **Load average:** If you're having performance issues, check the load average and tune the server for performance.
* **Disk/SAN/NAS utilization:** Check the I/O reports for externally attached storage to track and check the speed of read/write operations. If you find any issues, coordinate with the storage and network teams immediately to correct them.

## **Backup and disaster recovery planning**

Communicate with the backup team and provide them the data and client priorities for backup. The recommended backup criteria for production servers is:

* **Incremental backups:** Daily, Monday to Friday
* **Full backup:** Saturday and Sunday
* **Disaster recovery drills:** Perform restoration mock drills once a month (preferably, or quarterly if necessary) with the backup team to ensure the data can be restored in case of an issue.

## **Patching**

Operating system patches for known vulnerabilities must be implemented promptly. There are many types and levels of patches, including:

* Security
* Critical
* Moderate

When a patch is released, check the bug or vulnerability details to see how it applies to your system (e.g., does the vulnerability affect the hardware in your system?), and take any necessary actions to apply the patches when required. Make sure to cross-verify applications' compatibility with patches or upgrades.

## Application compatibility

Before going live with any application, check its compatibility with your hardware and operating system, and make sure to do load testing (with the support of application team).

## **Server hardening**

### Linux:

* **Set a BIOS password:** This prevents users from altering BIOS settings.
* **Set a GRUB password:** This stops users from altering the GRUB bootloader.
* **Deny root access:** Rejecting root access minimizes the probability of intrusions.
* **Sudo users:** Make sudo users and assign limited privileges to invoke commands.
* **TCP wrappers:** This is the weapon to protect a server from hackers. Apply a rule for the SSH daemon to allow only trusted hosts to access the server, and deny all others. Apply similar rules for other services like FTP, SSH File Transfer Protocol, etc.
* **Firewalld/iptables:** Configure firewalld and iptables rules for incoming traffic to the server. Include the particular port, source IP, and destination IP and allow, reject, deny ICMP requests, etc. for the public zone and private zone.
* **Antivirus:** Install antivirus software and update virus definitions regularly.
* **Secure and audit logs:** Check the logs regularly and when required.
* **Rotate the logs:** Keep the logs for limited period of time like "for 7 days", to keep the sufficient disk space for flawless operation.

## **Use a syslog server**

By configuring a syslog server in the environment to keep records of system and application logs, in the event of an intrusion or issue, the sysadmin can check previous and real-time logs to diagnose and resolve the problem.

## **Automation**

Many sysadmin tasks (such as server health checkups, resource utilization, backup triggers, transfer files and logs, etc.) must be done at specific times. Therefore, the sysadmin must write scripts or use external tools and configure them as cron jobs to do the tasks automatically at the proper time.

## **Monitoring tools**

Install and configure live monitoring tools like Nagios, HP, etc., to monitor your IT infrastructure and issue alerts about potential problems.

## **Conclusion**

While these are the most important tasks a sysadmin is responsible for, there is much more to the role than the duties on this list.

For example, the sysadmin must coordinate with multiple teams to resolve issues, communicate with and update customers, maintain 100% uptime, hold discussions with the audit team, prepare weekly/monthly/quarterly reports, do continuous monitoring of servers and services using appropriate tools, and maintain the hardware console and respond to any triggered alarms.

The sysadmin is always a single point of content (SPOC) in the data center or network operations center for issues related to web hosting, application and server outages, and other critical IT operations problems.

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

Android OS is a Linux-based mobile operating system that mainly runs on smartphones and tablets. The Android platform includes an operating system based on the Linux kernel, a graphical user interface, a web browser, and downloadable end-user applications.

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

Embedded operating systems are old news in the tech market, and engineers have developed dozens of them to facilitate hardware and firmware development. Because of this, we can now choose the best, utilizing those that went through thick and thin and have proven to be the best for engineering purposes. Since embedded systems are designed for narrowly-targeted functionalities, it's paramount to understand which tools for their development will fit all your requirements.

Embedded Linux includes a Linux kernel operating with the help of open-source software development tools and fundamental GNU utilities. Its baseline advantages for developers are readymade pieces of code they can incorporate into their embedded apps. They no longer have to work with cold and empty hardware, developing everything from scratch.

Thus, embedded development has shifted from the overly complex, which made engineers' blood run cold, to a quick and much less painful process. Also, the fact that Linux supports multiple software variants required for creating embedded apps, such as serial communication protocols, TCP/IP stack, etc., makes embedded Linux impossible to resist.

**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?**

**CLI** is the word form used for Command Line Interface. CLI permits users to put in writing commands associate degree exceedingly in terminal or console window to interact with an operating system. CLI is a platform or medium wherever users answer a visible prompt by writing a command and get the response from the system, for this users have to be compelled to kind command or train of command for performing the task. CLI is suitable for pricey computing wherever input exactitude is the priority.

**GUI** stands for Graphical User Interface. GUI permits users to use the graphics to interact with an operating system. In the graphical user interface, menus are provided such as windows, scrollbars, buttons, wizards, painting pictures, alternative icons, etc. It’s intuitive, simple to find out, and reduces psychological feature load. In GUI, the information is shown or presented to the user in any form such as: plain text, videos, images, etc.

1. A graphical user interface enables users to interact with the operating system or application. On the other hand, a CLI is an interface that allows the user to perform tasks by issuing commands in successive lines of text or command lines.
2. The CLI needs the commands to be memorized, making it difficult for newcomers. Professionals frequently prefer CLI. GUI is more user-friendly than CLI and may be utilized by both beginners and experienced professionals.
3. The CLI is ideal for dealing with difficult tasks. The difficult task may be handled by writing a few commands, whereas GUI requires some steps to be followed in order to function.
4. The CLI may or may not support multitasking. On the other hand, in a GUI, multitasking is simple. For example, MS Word and Chrome both have multitasking capabilities.
5. The GUI is slow than CLI. It took more time to complete tasks. On the other hand, CLI performs better. A script may be written to carry out a set of commands, although the GUI does not have this capability.
6. When interacting, only the keyboard and the system are used. On the other hand, the GUI has additional resources to engage with the user.
7. The interface in Command Line Interface is consistent all of the time. The interface in GUI changes as the software is updated.
8. In the Command Line Interface, there is less color strain on the eyes. The color strain is greater in GUI. CLI is a command-line interface that gives users greater control over the system. There are a few limitations in the GUI when it comes to accessing folders and doing tasks.
9. GUI is more flexible than CLI.
10. The graphical user interface (GUI) provides access to the operating system and files. However, CLI is still required to do complex tasks.

***Conclusion***: We got acquainted with different types of hypervisors, virtualization when working with operating systems. We got acquainted with the main types of modern operating systems, reviewed a brief overview of their capabilities.