**“Київський фаховий коледж зв’язку”**

**Циклова комісія Комп’ютерної та програмної інженерії**

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №8**

з дисципліни: «Операційні системи»

**Тема:** **“ Збереження службових даних системи та її мережева конфігурація”**

**Виконали**

**студенти**

**групи КСМ-13Б**

**Міньков Ілля**

**Колотуша Микола**

**Київ 2023**

**Мета роботи:**

**1. Отримання практичних навиків роботи з командною оболонкою Bash.**

**2. Знайомство з базовими структурами для збереження системних даних - процеси, память, лог-файли та**

**повідомлення про стан ядра.**

**3. Знайомство зі стандартом FHS.**

**4. Знайомство з діями при налаштуванні мережі.**

**Матеріальне забезпечення занять**

**1. ЕОМ типу IBM PC.**

**2. ОС сімейства Windows (Windows 7).**

**3. Віртуальна машина – Virtual Box (Oracle).**

**4. Операційна система GNU/Linux – CentOS.**

**5. Сайт мережевої академії Cisco netacad.com та його онлайн курси по Linux**

**Хід роботи**

**Виконував студент Міньков Ілля**

**4.1 Pseudo File System is a virtual structure that allows accessing various resources or information in the form of files or directories, even though these resources may not be physically part of the file system on disk. It provides an interface, similar to a file system, for interacting with various system resources or information.**

**A notable example of a pseudo file system is /proc in UNIX-like systems (including Linux). In /proc, you can find information about processes, memory, devices, kernel configurations, and other system parameters in the form of files or directories. This information is dynamically updated in real-time and allows system administrators and programs to access dynamic information about the system.**

**Pseudo file systems enable interaction with different parts of the system through a unified file system interface, simplifying the retrieval and utilization of system information.**

**4.2Users may not directly interact with the /proc directory frequently due to its nature as a virtual filesystem that exposes kernel and process information. Here are some reasons why users might not directly use /proc and how information can be obtained from it:**

**Complexity: The /proc directory contains a vast amount of information about processes, kernel parameters, and hardware. Navigating through this information directly using commands or scripts can be complex, and users may prefer higher-level commands or utilities that abstract this complexity.**

**Specialized Tools: Many specialized tools and utilities exist for extracting specific types of information from /proc. For example, commands like ps, top, and htop provide a more user-friendly interface to process-related information.**

**Readability: The information in /proc is often presented in a raw format and may not be as human-readable as the output from dedicated system monitoring tools.**

**If users still want to access information from /proc, they can use various command-line tools and utilities. For example:**

**To view detailed information about a process, users can use cat /proc/[pid]/status or cat /proc/[pid]/cmdline.**

**Information about the CPU can be obtained with commands like cat /proc/cpuinfo.**

**System and kernel information can be found in files like /proc/version and /proc/sys/.**

**Keep in mind that interacting directly with /proc requires an understanding of the information available and its format. Specialized tools and commands often provide a more user-friendly way to access and interpret this information.**

**4.3/proc/cmdline:**

**Purpose: This file contains the command-line arguments that were passed to the kernel during the system's boot. It provides information about the kernel parameters and options used during the boot process.**

**Usage: System administrators and users can examine /proc/cmdline to understand the kernel's configuration and parameters that were set during the boot.**

**/proc/meminfo:**

**Purpose: This file provides detailed information about the system's memory usage, including both physical and virtual memory. It includes information about total memory, free memory, used memory, and various memory-related statistics.**

**Usage: System administrators, monitoring tools, and users can use /proc/meminfo to monitor the system's memory usage and diagnose performance issues related to memory.**

**/proc/modules:**

**Purpose: This file lists the currently loaded kernel modules (device drivers) on the system. Each line in the file corresponds to a loaded module, providing information such as the module name, size, and memory address.**

**Usage: System administrators and users can inspect /proc/modules to see which kernel modules are currently loaded. This information is useful for troubleshooting, checking hardware support, or understanding the configuration of the running kernel.**

**4.4 The free command in the command line is used to display information about the system's memory usage. It provides various statistics, including the total, used, and free memory, as well as swap memory.**

**4.5Log files are text files that record information about various events, errors, user actions, and other occurrences in a computer system or software application. These files serve as important tools for tracking and analyzing events within a system. Here are some examples of the applications of log files:**

**System Logs:**

**Purpose: Record events and errors related to the operation of the operating system.**

**Example: /var/log/messages or /var/log/syslog in UNIX-based systems, Event Viewer in Windows.**

**Web Server Logs:**

**Purpose: Capture requests and responses of a web server, as well as information about access and errors.**

**Example: /var/log/nginx/access.log for Nginx, /var/log/apache2/access.log for Apache.**

**System Service Logs:**

**Purpose: Record information about the operation of services and daemons in the system.**

**Example: /var/log/auth.log for authentication logging in the system.**

**Application Logs:**

**Purpose: Document events and errors specific to an application.**

**Example: Application-specific log files that detail the activities and issues within the application.**

**Security Logs:**

**Purpose: Capture security-related events, such as login attempts, authentication failures, and other security-related activities.**

**Example: /var/log/secure in some UNIX-based systems.**

**Database Server Logs:**

**Purpose: Record events and queries related to a database server's operation.**

**Example: MySQL server logs, PostgreSQL logs.**

**These log files play a crucial role in system administration, troubleshooting, and security analysis by providing a historical record of system and application activities.**

**4.6** **The file /var/log/dmesg contains kernel ring buffer messages and information about events that occurred during the system boot process. The name "dmesg" stands for "display messages" or "driver messages." The primary purpose of this file is to provide information about detected devices and events happening at the kernel level.**

**Some of the key information found in the /var/log/dmesg file includes:**

**Device Initialization: Information about devices that were recognized and initialized during the system boot.**

**Errors and Issues: Messages about errors encountered during boot, such as problems with reading devices or drivers.**

**Memory Information: Details about memory allocation and usage during the system boot.**

**Kernel Version: Information about the version of the kernel that was loaded.**

**Timestamps: Timestamps for various events occurring during the boot process.**

**The /var/log/dmesg file is often used by system administrators or technicians for troubleshooting issues related to the system boot or device detection.**

**4.7The Filesystem Hierarchy Standard (FHS) is a set of guidelines and standards that define the structure of the file system on Unix-like operating systems. Its purpose is to promote consistency and interoperability among different Unix-like systems. The FHS specifies the layout of directories and the arrangement of files within those directories, ensuring that software developers, system administrators, and users can have a common understanding of where to find and place files.**

**Key objectives and purposes of FHS include:**

**Consistency Across Systems:**

**FHS aims to establish a common and consistent directory structure across various Unix-like systems. This consistency simplifies the development and porting of software across different platforms.**

**Ease of Maintenance:**

**By providing a standardized structure, FHS makes it easier for system administrators to understand and maintain the file system. This facilitates tasks such as backups, upgrades, and system management.**

**Interoperability:**

**FHS enhances interoperability between different Unix-like systems. Software applications designed to follow FHS guidelines can be installed and run on any system that adheres to the standard.**

**Portability:**

**FHS promotes portability by defining common directory paths for system files, configuration files, libraries, and other essential components. This makes it easier to create software that works seamlessly on various Unix-like platforms.**

**Simplifying Software Development:**

**Software developers benefit from FHS by having a consistent and predictable file system structure. This simplifies the development and packaging of software for Unix-like systems.**

**Enhancing User Experience:**

**Users benefit from FHS as it provides a standardized layout that simplifies navigation and access to system files and user-related data.**

**4.8 In Linux and other Unix-like operating systems, access to commands for viewing and configuring the network can be achieved through the command line or various user interfaces. Here are some basic commands that can be used:**

**ifconfig:**

**Purpose: Displays information about network interfaces, sets the status, and configures network parameters.**

**Example:**

**ifconfig**

**ip:**

**Purpose: Provides advanced network configuration capabilities, including information about interfaces, routes, and more.**

**Example:**

**ip addr show**

**ip route show**

**iwconfig:**

**Purpose: Displays information about wireless network interfaces.**

**Example:**

**iwconfig**

**route:**

**Purpose: Displays and configures the routing table.**

**Example:**

**route -n**

**netstat:**

**Purpose: Displays various information about network connections, routing, and more.**

**Example:**

**netstat -i**

**netstat -nr**

**ss:**

**Purpose: Displays information about sockets, and it can serve as an alternative to netstat.**

**Example:**

**ss -a**

|  |  |  |
| --- | --- | --- |
| df -h | | Lists all file systems on the system, including their size, free space, and usage. |
| du -sh | | Returns the size of files or directories using gigabytes (GB). |
| lsblk | | Displays a list of all block devices on the system. |
| cat /etc/fstab | | Lists all file systems that are available when the system is booted. |
| mount | | Mounts the file system. |
| umount | | Unmounts the file system. |
| ip a | | Displays a list of all network interfaces on the system. |
| ifconfig | | Outputs similar data as ip a. |
| route -n | Displays the routing table. | |
| ping [адреса або ім'я хоста] | Sends ICMP packets to the host and displays response times. | |
| traceroute [адреса або ім'я хоста] | Traces the path of ICMP packets to a host address. | |

**КОНТРОЛЬНІ ЗАПИТАННЯ**

**Виконував студент Колотуша Микола**

**1.** **Relationship between cat and tac commands:**

**cat: Displays the contents of files in straight order.**

**tac: Does the same, but displays the contents of the files in reverse order (from end to beginning).**

**2.ss command:**

**Displays information about sockets, i.e. network connections, on the system.**

**3.Difference between ps --forest and pstree:**

**ps --forest: Displays process information in the form of a tree.**

**pstree: Also displays information as a tree, but this is a separate command, different from ps.**

**4.System configuration directories:**

**/etc: General system settings.**

**/var: Some settings that change as the system works.**

**5.Directory with installed applications:**

**/bin, /usr/bin: Executable files for users.**

**/sbin, /usr/sbin: Superuser executable files.**

**6.System and Super User Applications:**

**/sbin, /usr/sbin: System applications.**

**/usr/bin: Applications for users.**

**7.Designation of commands:**

**ping: Check the availability of a network device or server.**

**ifconfig: Output information about network interfaces and their status.**

**traceroute: Tracking the path to the specified server or domain.**

**8.Network Interfaces in Linux:**

**They are called network devices or interfaces, such as eth0, eth1 and so on.**

**9.Output settings of a single network interface using ifconfig:**

**ifconfig eth1**

**Conclusion:**

During the execution of the LB, I learned about new commands and what they mean, but unfortunately it did not work out in practice, because of problems with the terminal