Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №10**

з дисципліни: «Операційні системи»

**Тема: “Спеціальні каталоги та файли в Linux”**

Виконали:

студенти групи РПЗ-93б

Філіпено А.Д.,

Афанасьєв І.В.

Перевірив викладач

Сушанова В.С.

Київ 2022

**The purpose of the work:**

1. Gaining practical skills in working with the Bash command shell.

2. Familiarity with special directories and files in Linux.

**Material support:**

1. Computer type IBM PC.

2. Windows 7.

3. Virtual machine - Virtual Box (Oracle).

4. GNU / Linux operating system - CentOS.

5. Cisco Network Academy website netacad.com and its online Linux courses.

**Tasks for preliminary preparation(Andrii)**

1. Read brief theoretical information for laboratory work and make a small dictionary of basic English terms on the purpose of team assignments and their parameters.

|  |  |
| --- | --- |
| **Term** | **Purpose** |
| chmod u+s file | To add the setuid permission symbolically |
| chmod u-s file | To remove the setuid permission symbolically |
| setgidpermission | The setgid permission is similar to setuid, but it makes use of the group owner permissions. |
| chmod g+s <file|directory> | To add the setgid permission symbolically |
| chmod g-s <file|directory> | To remove the setgid permission symbolically |
| sticky bit | Sticky bit permission is used to prevent other users from deleting files that they do not own in a shared directory. |
| links | посилання |
| inode number*.* | A unique identification number of file. |
| symbolic link | A soft link, is simply a file that points to another file. |

2. Based on the considered material give answers to the following questions:

2.1. The Unix access rights flags **setuid** and **setgid** allow users to run an executable with the file system permissions of the executable's owner or group respectively and to change behaviour in directories. They are often used to allow users on a computer system to run programs with temporarily elevated privileges in order to perform a specific task. While the assumed user id or group id privileges provided are not always elevated, at a minimum they are specific.

Main difference:

* setuid: a bit that makes an executable run with the privileges of the *owner* of the file
* setgid: a bit that makes an executable run with the privileges of the *group* of the file

2.2. The **sticky bit** restricts who can delete files in a directory on Linux systems. Specifically, when the sticky bit is set, only the user that owns, the user that owns the directory, or the root user can delete files within the directory. In some cases, the sticky bit is more intuitively referred to as a "restricted deletion flag" or "restricted deletion bit".

**Progress(Andrii)**

* 1. Create a table of commands studied in paragraph 2 in the following form:

|  |  |
| --- | --- |
| **Command name** | **Its purpose and functionality** |
| ls -ld /tmp  ls -ld /var/tmp | Using the -d option for the ls command lists directory information; combined with the -l option it shows ownership and permissions for the directory files. List the details of the /tmp and /var/tmp directories. |
| ln | create a hard link |
| rm | remove files |
| ln -s source softlink | The -s option for the ln command creates a symbolic link instead of a hard link. |

**Answers to control questions(Ilya)**

1. To create hard links, the ln command is used with two arguments. The first argument is an existing file name to link to, called a target, and the second argument is the new file name to link to the target.

ln target name

The peculiarity of hard links is that they directly point to the index descriptor number and, accordingly, such names can only point to files within the same file system, ie on the same media on which is the directory containing this name.

1. The -s option for the ln command creates a symbolic link instead of a hard link. Create a symbolic link to the source file and view the details of both files:

ln -s source softlink

Unlike hard links, soft links can be linked to directories and can cross devices and partitions to their targets.

1. **Hard links:**

* **Hard links don’t have a single point of failure. I**f you have five files hard linked together, then deleting any four of these files would not result in deleting the actual file contents.
* Since each file system (partition) has a separate set of inodes, hard links cannot be created that attempt to cross file systems
* Hard links cannot be created on directories. The reason for this limitation is that the operating system itself uses hard links to define the hierarchy of the directory structure.

**Soft links:**

* Symbolic links, have a single point of failure: the original file. If the original file is removed, then any files linked to it fail.
* **Soft links are easier to see.** Soft links are much more visual, not requiring any extra commands beyond the ls command to determine the link.
* **Soft links can link to any file and directory.**

1. **-** Deleting any four of these files would not result in deleting the actual file contents;

* If the original file is removed, then any files linked to it fail;
* If you see a regular file with a link count that is greater than one, you can use the find command with the -inum search criteria to locate the other files that have the same inode number.

**Conclusion:**

We gained practical skills in working with the Bash shell and got acquainted with special directories and files in Linux.