**This part was completed by Богдан Раєв**

**Lab 18: Special directories and files**

Example 1: Inspecting a specific directory

List the contents of /dev/disk/by-id/:

ls /dev/disk/by-id/

Example 2: Displaying details of a specific file

View the information within /proc/cpuinfo:

cat /proc/cpuinfo

Example 3: Monitoring system status

Check system performance using top:

top

Example 4: Observing active processes

Display all running processes:

ps -ef

Example 5: Initiating a program

Launch the program Vim with administrative privileges:

sudo /usr/bin/vim

Example 6: Exiting a program

Exit Vim without saving changes:

:q!

Example 7: Restarting the system

Reboot the system using administrative privileges:

sudo reboot

Example 8: Powering off the system

Shutdown the system using administrative privileges:

sudo poweroff

**Control questions:**

**1. Give examples of changing access rights by symbolic method (Symbolic Method)?**

\* Access privileges are indicated by symbols when modifying access rights to files and directories using the symbolic approach.

- Illustrations:

\* Give the file owner read permission:

chmod u+r file

\* Give the file group read and write permissions:

chmod g+rw in the file

\* Give every user the ability to read, write, and execute:

chmod a+rwx in the file

\* Take away other users' privileges to record:

chmod o-w in the file

\* Take away each user's rights:

a-rwx file modification

**2. Give examples of changing access rights by numerical method (numeric method, octal method)?**

\* Octal numbers are used in the numeric approach to change the permissions of files and directories.

Examples include: \* Grant the file owner read permission.

chmod 644 file

\* Give the file group read and write permissions:

chmod 664 file

\* Give every user the ability to read, write, and execute:

chmod 774 in the file

\* Take away other users' privileges to record:

640 file chmod

\* Take away each user's rights:

chmod 000 file

**3. Is it possible to execute a file that has execute rights but not set read rights (--x)?**

No, a file with execute privileges but no read rights set cannot be executed. You need read permissions to access a file's contents before you can run it.  
**4. What is the purpose of the umask command?**

To set the default permission pattern for newly created files and directories, use the umask command.

**5. If we change access rights and permissions in the current session, will they be saved in the next session?**

Yes, the permissions and access levels that we modify in this session will remain in effect in the following one. Nevertheless, if we use the umask command to modify permissions and access rights, those modifications will only take effect for newly created files and folders.

**6. Is there any pattern that the system uses regarding rights and accesses when creating new files. As is it possible to change the default permissions?**

Yes, the system employs the 0666 and 0777 default permission patterns for newly created files and folders, respectively. This indicates that all users have the ability to read, write, and execute permissions to newly created files and folders.

The umask command can be used to modify the default permissions. For instance, you can use the following command to specify 0644 as the default permission pattern for newly created files:

**umask 0644**

**7. Imagine that a program needs to create a one-time temporary file that is never created again will be needed after closing the program. What is the correct directory to create this file?**

After the program is closed, you should create a one-time temporary file in the /tmp directory that will never be needed again. After a system reboot, temporary files that don't need to be saved go in this directory.

**8. How can you create a hard link? In what situations is it advisable to use them?**

A hard link is established using the ln command, often with the -s option to create symbolic links. For instance, to generate a hard link pointing to the /etc/passwd file, execute the following command:

ln -s /etc/passwd /tmp/passwd

**9. How can you create a symbolic link? In what situations is it advisable to use them?**

A symbolic link is generated via the ln command with the -s option. For instance, to create a symbolic link pointing to the /etc/passwd file, execute the following command:

ln -s /etc/passwd /tmp/passwd

Symbolic links represent file paths; when accessed, the system redirects the request to the file referenced by the link.

Utilize symbolic links in the following scenarios:

Facilitating access to a file without requiring file duplication.

Crafting aliases for files.

Granting access to files situated in alternate directories.

**10. Compare hard and symbolic links?**

**Hard and symbolic links have the following differences:**

|  |  |  |
| --- | --- | --- |
| **Characteristic** | **Hard link** | **Symbolic link** |
| **Type** | **File** | **File** |
| **Contents** | **The linked file's index number** | **The file path that the link points to** |
| **Behavior when deleting a file** | **The hard link will be removed together with the original file when it is deleted.** | **The symbolic link will stay active but become invalid if the original file is removed.** |
| **Can be used for catalogs** | **Yes** | **No** |
|  |  |  |

**11. There is an original file and two links - symbolic and hard - have been created for it. What will happen with other files, if you delete:**

\* Original file: - There will be no hard connection left. - Though it will no longer be valid, the symlink will stay.

\* Symbolic connection:

- Nothing is going to occur.

\* Hard connection:

- Nothing is going to occur.

**Conclusion:**

I gained knowledge of new commands and their meanings during the LB's execution, but sadly, because to terminal issues, it did not function as intended.