Task 2: Linux Command-Line File Operations

Objective:
The objective of this task is to practice basic Linux command-line operations related to directory creation, file handling, permission management, and verification using standard Linux utilities. This document is prepared as a professional submission suitable for upload to a training GitLab repository.

1. Task Requirements

The following operations are performed in this task:
• Create a directory structure /tmp/myfiles
• Navigate into the directory
• Create three text files and add sample content
• List files in the directory
• Change permissions of one file to read-only for the group

2. Creating Directory Structure

Command Used:
mkdir -p /tmp/myfiles

Explanation:
The mkdir command is used to create directories. The -p option ensures that parent directories are created if they do not already exist and prevents errors if the directory already exists.

Command Used:
cd /tmp/myfiles

Explanation:
The cd (change directory) command is used to navigate into the newly created directory.

3. Creating Files and Writing Content

Command Used:
touch file1.txt file2.txt file3.txt

```
student@student-virtual-machine:~/25SUB4508_LSP$ mkdir -p /tmp/myfiles
student@student-virtual-machine:~/25SUB4508_LSP$ cd /tmp/myfiles/
student@student-virtual-machine:/tmp/myfiles$ pwd
/tmp/myfiles
student@student-virtual-machine:/tmp/myfiles$
```

Explanation:
The touch command is used to create empty files. In this case, three text files are created simultaneously.

Command Used:
echo "This is sample content for file1" > file1.txt
echo "This is sample content for file2" > file2.txt
echo "This is sample content for file3" > file3.txt

```
student@student-virtual-machine:/tmp/myfiles$ touch file1.txt file2.txt file3.txt
student@student-virtual-machine:/tmp/myfiles$ nano file1.txt
student@student-virtual-machine:/tmp/myfiles$ nano file2.txt
student@student-virtual-machine:/tmp/myfiles$ nano file3.txt
student@student-virtual-machine:/tmp/myfiles$
```

Explanation:
The echo command writes text to standard output, and the > operator redirects this output into the specified files, replacing any existing content.

4. Listing Files in the Directory

Command Used:
ls -l

```
student@student-virtual-machine:/tmp/myfiles$ ls -l
total 12
-rw-rw-r-- 1 student student 33 Jan 15 11:10 file1.txt
-rw-rw-r-- 1 student student 33 Jan 15 11:10 file2.txt
-rw-rw-r-- 1 student student 34 Jan 15 11:10 file3.txt
student@student-virtual-machine:/tmp/myfiles$
```

Explanation:
The ls command lists directory contents. The -l option displays detailed information including file permissions, ownership, size, and modification time.

5. Changing File Permissions

Command Used:
chmod g=r file2.txt

Explanation:
The chmod command modifies file permissions. The g=r option sets the group permission to read-only, removing write and execute permissions for the group.

Verification Command:
ls -l file2.txt

```
student@student-virtual-machine:/tmp/myfiles$ chmod g=r file2.txt
student@student-virtual-machine:/tmp/myfiles$ ls -l file2.txt
-rw-r--r-- 1 student student 33 Jan 15 11:10 file2.txt
student@student-virtual-machine:/tmp/myfiles$
```

6. Conclusion

This task demonstrates essential Linux command-line skills including directory management, file creation, content writing, permission handling, and verification. These operations are fundamental for Linux system administration and development workflows.