Linux for Data Engineers – Group Practical Activity Brief 2

Searching for Files and Directories

1. Using ‘find’ to search for Files and Directories

`find` (Search for Files and Directories)

- Find all `.txt` files in the current directory and subdirectories.

```bash
find . -name "*.txt"
```

- Find files modified in the last 7 days.

```bash
find . -type f -mtime -7
```

- Find and delete empty directories.

```bash
find . -type d -empty -delete
```

2. Using ‘grep’ to search inside files

`grep` (Search Inside Files)

- Search for a specific string in all `.log` files.

```bash
grep "ERROR" *.log
```

- Search recursively in all files in the current directory.

```bash
grep -r "search_term" .
```

- Find all `.log` files and count the occurrences of "ERROR".

```bash
find . -name "*.log" | xargs grep -c "ERROR"
```

Redirection (`>` and `>>`)

- Redirect output of a command to a file.

```bash
du -sh . > disk_usage.txt
```

- Append output to an existing file.

```bash
echo "New log entry" >> logfile.log
```

3. **Changing file permissions**

Changing File Permissions

- Make a script executable.

```bash
chmod +x script.sh
```

Changing File Ownership

- Change the owner of a file.

```bash
chown new_owner file_name
```

2. Using ‘at’ for one-time tasks

Using `at` for One-Time Tasks

Schedule a one-time task to run at a specific time.

```bash
echo "/path/to/script.sh" | at 02:00 tomorrow
```

3. Using ‘cron’ for regularly scheduled tasks

Using `cron` for Regularly Scheduled Tasks

Scheduling a Task to Run Every Monday at 3 AM

- Open the `crontab` editor.

```bash
crontab -e
```

- Add the following line.

```bash
0 3 * * 1 /path/to/script.sh
```