

# Assignment 1

## Task 1: Basic Linux Commands in a Real-World Scenario

### 1. Check current logged-in user and system information:

- **Command:**
  - `whoami` - Displays the username of the current user.
  - `uname -a` - Shows detailed system information, including kernel version, system architecture, and OS name.
- **Explanation:**

These commands are useful to verify the logged-in account and check the system details before performing administrative tasks.

```
ubuntu@ip-172-31-22-130:~$ whoami
ubuntu
ubuntu@ip-172-31-22-130:~$ uname -a
Linux ip-172-31-22-130 6.8.0-1024-aws #26-Ubuntu SMP Tue Feb 18 17:22:37 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
```

---

### 2. Navigate to the /projects directory and list contents:

- **Command:**
  - `cd /projects` - Changes the current directory to /projects.
  - `ls -l` - Lists files and directories with detailed information (permissions, owner, size, modification date).
- **Explanation:**

Navigating to the project directory allows for organization, and listing contents helps ensure that the desired folder is accessible.

```
ubuntu@ip-172-31-22-130:~$ cd projects
ubuntu@ip-172-31-22-130:~/projects$ ls -l
total 0
```

---

### 3. Create a new project directory and verify it:

- **Command:**
  - `mkdir projectB` - Creates a new directory named projectB.
  - `ls -l` - Confirms that the new directory has been created.
- **Explanation:**

This step sets up the project folder structure, crucial for organizing files related to a specific project.

```
ubuntu@ip-172-31-22-130:~/projects$ ls -l
total 4
drwxrwxr-x 2 ubuntu ubuntu 4096 May  7 10:28 projectB
```

---

#### 4. Create a sample file inside projectB:

- **Command:**
  - `touch projectB/README.txt` - Creates an empty file.
  - `echo "Welcome to Project B" > projectB/README.txt` - Writes a welcome message into the file.
  - `cat projectB/README.txt` - Displays the file's contents.
- **Explanation:**

Creating a readme file provides documentation within the project directory, and verifying the file content ensures that the message was saved correctly.

```
ubuntu@ip-172-31-22-130:~$ touch projectB/README.txt
ubuntu@ip-172-31-22-130:~$ echo "Welcome to Project B" > projectB/README.txt
ubuntu@ip-172-31-22-130:~$ cat projectB/README.txt
Welcome to Project B
```

---

## Task 2: User and Group Permissions Management

### 1. Create a new user and assign to a group:

- **Command:**
  - `sudo useradd -m -G developers john` - Creates a user `john` and adds them to the `developers` group.
  - `sudo passwd john` - Sets the password for the new user.
- **Explanation:**

Managing users and groups is essential to restrict access and assign roles within the system.

```
ubuntu@ip-172-31-22-130:~$ sudo useradd -m -G developers john
ubuntu@ip-172-31-22-130:~$ sudo passwd john
New password:
Retype new password:
passwd: password updated successfully
```

---

### 2. Verify user and group information:

- **Command:**
  - `id john` - Displays user ID, group ID, and associated groups for john.
- **Explanation:**

Checking user information ensures that the account has been correctly created and assigned to the intended group.

```
ubuntu@ip-172-31-22-130:~$ id john
uid=1001(john) gid=1002(john) groups=1002(john),1001(developers)
```

---

### 3. Change group ownership of projectB:

- **Command:**
  - `sudo chown :developers /projects/projectB` - Changes the group ownership of projectB to developers.
- **Explanation:**

Changing the group ownership ensures that only members of the `developers` group can manage the directory.

```
ubuntu@ip-172-31-22-130:~/projects$ cd ..
ubuntu@ip-172-31-22-130:~$ sudo chown :developers projects/projectB
ubuntu@ip-172-31-22-130:~$ l
```

---

### 4. Modify permissions to allow group write access:

- **Command:**
  - `sudo chmod 770 /projects/projectB` - Grants read, write, and execute permissions to the owner and group, but no access to others.
- **Explanation:**

This ensures secure access control, allowing only the owner and group to modify files.

```
ubuntu@ip-172-31-22-130:~$ sudo chmod 770 projects/projectB
ubuntu@ip-172-31-22-130:~$ ls -ld projects/projectB/
drwxrwx--- 2 ubuntu developers 4096 May  7 10:28 projects/projectB/
```

---

## Task 3: Changing File Ownership

### 1. Change ownership of projectB to john:

- **Command:**
  - `sudo chown john:developers /projects/projectB` - Sets john as the owner and developers as the group.

- **Explanation:**  
Assigning ownership to the lead developer allows for better file management by the responsible person.

## 2. Verify the ownership change:

- **Command:**
  - `ls -ld /projects/projectB` - Displays ownership and permissions for the directory.
- **Explanation:**  
Verifying ownership changes ensures that the correct user now has control over the directory.

```
ubuntu@ip-172-31-22-130:~$ sudo chown john:developers projects/projectB
ubuntu@ip-172-31-22-130:~$ ls -ld projects/projectB
drwxrwx--- 2 john developers 4096 May  7 10:28 projects/projectB
```

---

## Task 4: System-Level Monitoring Commands

### 1. Check system uptime:

- **Command:**
  - `uptime` - Displays how long the system has been running and the load averages.
- **Explanation:**  
Monitoring uptime helps assess the stability and availability of the system.

```
ubuntu@ip-172-31-22-130:~$ uptime
10:40:43 up 23 min,  1 user,  load average: 0.03, 0.01, 0.00
```

### 2. Monitor disk usage:

- **Command:**
  - `df -h` - Shows disk space usage in a human-readable format.
- **Explanation:**  
Checking disk space helps prevent issues caused by low storage, especially before installing large applications.

```
ubuntu@ip-172-31-22-130:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        6.8G  1.7G  5.1G  26% /
tmpfs            458M   0  458M   0% /dev/shm
tmpfs            183M  880K  182M   1% /run
tmpfs            5.0M   0   5.0M   0% /run/lock
efivarfs         128K   3.6K  120K   3% /sys/firmware/efi/efivars
/dev/nvme0n1p16  881M   79M  741M  10% /boot
/dev/nvme0n1p15  105M   6.1M   99M   6% /boot/efi
tmpfs            92M   12K   92M   1% /run/user/1000
```

### 3. Check memory usage:

- **Command:**
  - `free -m` - Displays available and used memory in megabytes.
- **Explanation:**  
Monitoring memory usage ensures that there is enough RAM available for new processes.

```
ubuntu@ip-172-31-22-130:~$ free -m
              total        used         free       shared    buff/cache   available
Mem:           914          352           354            2           361          562
Swap:            0            0            0
```

### 4. Monitor running processes:

- **Command:**
  - `ps aux --sort=-%mem | head -5` - Lists the top 5 processes consuming the most memory.
- **Explanation:**  
Identifying resource-hungry processes helps optimize system performance by allowing the administrator to take action.

```
ubuntu@ip-172-31-22-130:~$ ps aux --sort=-%mem | head -5
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root        618  0.0  3.4 1468392 31916 ?        Ss1  10:16   0:01 /usr/lib/snapd/snapd
root       187  0.0  2.8 288952 27136 ?        Ss1  10:16   0:00 /sbin/multipathd -d -s
root       730  0.0  2.4 110000 22912 ?        Ss1  10:16   0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
root       611  0.0  2.1  32408 20480 ?        Ss   10:16   0:00 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
```