Assignment 1: **Basic Linux and Git command.**

BASIC LINUX COMMANDS

- ★ `ls`: List the contents of a directory.
- ★ `pwd`: Print the working directory (your current location).
- ★ `cd`: Change directory (navigate to a different directory).
- ★ `mkdir`: Create a new directory.
- ★ `cp`: Copy files or directories.
- ★ `mv`: Move or rename files or directories.
- ★ `rm`: Remove files or directories (use with caution!).
- ★ `touch`: Create an empty file.
- ★ `rmdir`: Removes an empty directory.
- ★ `cat`: Display the contents of a file on the terminal.
- ★ `more` / `less`: View files one screen at a time (better for large files).
- ★ `head`: Display the first few lines of a file.
- ★ `tail`: Display the last few lines of a file.
- ★ `chmod`: Change file or directory permissions.
- ★ `chown`: Change file or directory ownership.
- ★ `uname`: Get information about the kernel version.
- ★ `whoami`: Show your current username.
- ★ `df`: Show disk usage.
- ★ `ps`: List processes running on the system.
- ★ `man`: Get help on a specific command (e.g., `man ls`).
- ★ `clear`: Clear the terminal screen.
- ★ `exit`: Exit the terminal session.
- ★ `sudo`: Run a command with superuser privileges (use with caution!).
- ★ `top:` Displays real-time system processes and resource usage.
- ★ `free:` Displays memory usage.
- ★ `Ifconfig:` Configures network interfaces
- ★ `ping:` Tests connectivity to another host.
- ★ `netstat:` Displays network connections, routing tables, and interface statistics.

BASIC GIT COMMANDS

★ `git config`

Description: Sets user-specific configuration settings for Git.

Usage:

- **Set user name:** `git config --global user.name "Your Name"`
- **Set user email:** `git config --global user.email "your.email@example.com"`

★ `git init`

Description: Initializes a new Git repository.

Usage: `git init`

★ `git clone`

Description: Clones an existing repository into a new directory.

Usage: `git clone https://github.com/user/repository.git`

★ `git add`

Description: Adds changes in the working directory to the staging area.

Usage:

- **Add a specific file:** `git add file_name`
- **Add all changes:** `git add .`

★ `git commit`

Description: Records changes to the repository with a message.

Usage: `git commit -m "Commit message"`

★ `git status`

Description: Shows the status of changes as untracked, modified, or staged.

Usage: git status

★ **git diff**

Description: Shows changes between commits, commit and working tree, etc.

Usage: git diff

★ **git branch**

Description: Lists, creates, or deletes branches.

Usage:

- **List branches: git branch**
- **Create a new branch: git branch branch_name**
- **Delete a branch: git branch -d branch_name**

★ **git checkout**

Description: Switches to another branch or restores working tree files.

Usage:

- **Switch to a branch: git checkout branch_name**
- **Create and switch to a new branch: git checkout -b branch_name**

★ **git merge**

Description: Merges changes from one branch into the current branch.

Usage: git merge branch_name

★ **git rebase**

Description: Reapplies commits on top of another base tip.

Usage: git rebase branch_name

★ git remote

Description: Manages set of tracked repositories.

Usage:

- List remote repositories: git remote
- Add a remote repository: git remote add origin https://github.com/user/repository.git

★ git fetch

Description: Downloads objects and refs from another repository.

Usage: git fetch origin

★ git pull

Description: Fetches from and integrates with another repository or a local branch.

Usage: git pull origin branch_name

★ git push

Description: Updates remote refs along with associated objects.

Usage: git push origin branch_name

★ git reset

Description: Resets current HEAD to the specified state.

Usage:

- Soft reset (keeps changes in working directory): git reset --soft HEAD~1
- Hard reset (discards all changes): git reset --hard HEAD~1

★ git revert

Description: Creates a new commit that undoes changes from a previous commit.

Usage: git revert commit_id

★ git stash

Description: Stashes changes in a dirty working directory away.

Usage:

- Save changes: git stash
- Apply stashed changes: git stash apply

★ git log

Description: Shows commit logs.

Usage: git log

★ git show

Description: Shows various types of objects, such as commits.

Usage: git show commit_id

★ git blame

Description: Shows what revision and author last modified each line of a file.

Usage: `git blame file_name`