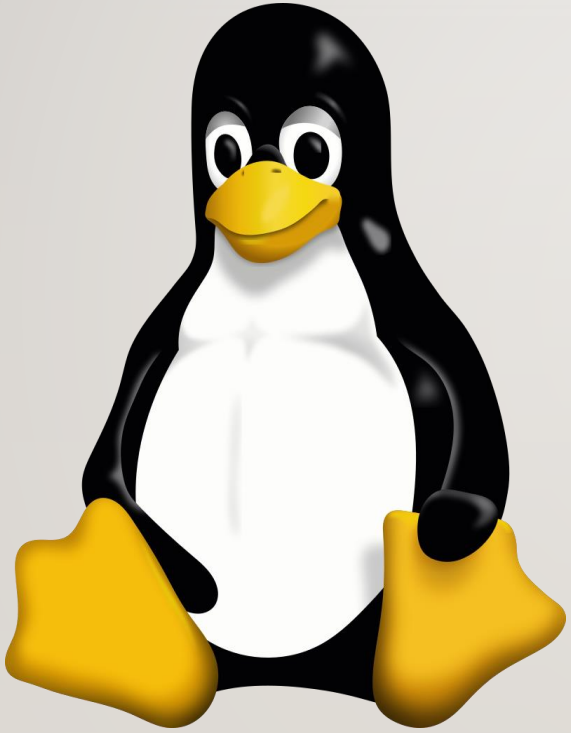
Introduction to Linux & Basic Commands



Ubuntu

```
ls              # Lists files and directories in the current directory
ls -l           # Long listing, shows file permissions, owner, size, and modificat
ls -a           # Show hidden files (those that start with a dot)
ls -lh          # Human-readable file sizes
```

- `cd` (Change directory):

```bash
cd /home/user/Documents    # Navigate to the Documents folder
cd ..                      # Move one level up in the directory structure
cd ~                       # Go to the home directory
```

- `pwd` (Print working directory):

```bash
pwd         # Shows the full path of the current working directory
```

```bash
pwd          # Shows the full path of the current working directory
```

- `mkdir` (Make directory):

```bash
mkdir new_folder          # Create a directory named "new_folder"
mkdir -p project/data     # Create directories including any necessary parent dire
```

- `cp` (Copy files):

```bash
cp file.txt backup.txt       # Copy file.txt to backup.txt
cp -r folder/ new_folder/    # Copy the contents of a directory recursively
```

- `mv` (Move or rename files):

```bash
mv file.txt new_file.txt     # Rename a file
mv file.txt /path/to/dir     # Move file to another directory
```

- `rm` (Remove files or directories):

```bash
rm file.txt              # Remove a file
rm -r directory/         # Remove a directory and its contents
rm -i file.txt           # Prompt before removal
```

3. **File Permissions**:

- `chmod` (Change file mode bits):

```bash
chmod 755 script.sh      # Set read, write, execute for the owner; read and execut
chmod u+x script.sh      # Add execute permission for the user
```

- `chown` (Change file ownership):

```bash
sudo chown user:group file.txt   # Change the owner and group of a file
```

4. Text Editors:

- Nano:

```bash
nano filename.txt       # Open a file in Nano editor
Ctrl+X                  # Exit Nano
```

- Vim:

```bash
vim filename.txt        # Open file in Vim editor
i                       # Insert mode to edit text
:wq                     # Save and quit
```

4. Text Editors:

- Nano:

```bash
nano filename.txt       # Open a file in Nano editor
Ctrl+X                  # Exit Nano
```

- Vim:

```bash
vim filename.txt        # Open file in Vim editor
i                       # Insert mode to edit text
:wq                     # Save and quit
```

Key Topics & Commands:

1. Viewing and Editing Files:

- `cat` (Concatenate files and print to standard output):

```bash
cat file.txt                # Print file contents to the terminal
cat file1.txt file2.txt     # Concatenate multiple files and print them
```

- `less` (View file content page by page):

```bash
less large_file.txt         # View large file one page at a time
```

- `head` and `tail` (View the beginning and end of files):

```bash
head -n 10 file.txt          # View the first 10 lines of the file
tail -n 20 file.txt          # View the last 20 lines of the file
```

. Data Filtering & Processing:

- `grep` (Global Regular Expression Print):

```bash
grep "error" logfile.txt     # Find lines containing "error" in a log file
grep -i "error" logfile.txt  # Case-insensitive search for "error"
```

- `awk` (Text processing language):

- `grep` (Global Regular Expression Print):

```bash
grep "error" logfile.txt     # Find lines containing "error" in a log file
grep -i "error" logfile.txt  # Case-insensitive search for "error"
```

- `awk` (Text processing language):

```bash
awk '{print $1, $3}' data.csv    # Print the 1st and 3rd columns from a CSV file
awk -F"," '{print $2}' data.csv  # Specify comma as the field separator
```

- `sed` (Stream editor):

```bash
sed 's/error/success/g' logfile.txt    # Replace "error" with "success" globally i
```

4. **Downloading Files:**

- `wget`:

```bash
wget http://example.com/data.csv   # Download a file from the internet
```

- `curl`:

```bash
curl -O http://example.com/data.csv  # Download a file with curl
```

1. **Process Management:**

   - `ps` (Process status):

```bash
ps aux                    # View all running processes
```

   - `top` (Real-time system monitoring):

```bash
top                       # Display real-time CPU and memory usage
```

- `kill` (Terminate a process):

```bash
kill 1234                    # Terminate a process with PID 1234
```

Task Scheduling with `cron`:

- `crontab` (Schedule jobs):

```bash
crontab -e                   # Edit cron jobs
# Add this line to schedule a script every day at 6 AM:
0 6 * * * /path/to/script.sh
```