Revision Notes: Linux Commands and Concepts

Welcome to the revision notes for our Linux class. This document will cover all the key concepts discussed in the class, including Linux commands, navigation, file handling, and the comparison of hardlinks and softlinks. These notes should serve as a comprehensive guide to understanding Linux commands with tips, analogies, and practical usage scenarios.

Basic Linux Navigation Commands

1. PWD (Present Working Directory)

- **Purpose:** Displays the current directory you're operating in【4:8†source】.
- **Usage:** Simply type `pwd`, and it will return the absolute path of the current working directory.

2. CD (Change Directory)

- **Purpose:** Used to change the current directory【4:4†source】【4:8†source】.
- **Usage Patterns:**
  - `cd <directory-path>` - Moves you to the specified directory.
  - `cd ..` - Moves you up one directory to the parent directory.
  - `cd ../..` - Moves you up two directories (you can combine `..` with slashes to navigate up multiple levels)【4:8†source】.
  - **Analogy:** Using CD is like setting a destination in GPS; you can specify either a relative or absolute path to navigate【4:15†source】.

File and Directory Handling

1. LS (List Directory Contents)

- **Advanced Usage:**
  - `ls <file-name>` - Checks if the file exists and displays its name.
  - `ls *.txt` - Lists all files with a `.txt` extension using the wildcard `*` 【4:15†source】.
  - `ls a*.txt` - Lists files starting with 'a'【4:15†source】.
  - **Concept of Wildcards:** Using `*` for multiple characters and `?` for a single character can help in listing files based on patterns【4:6†source】.

2. MKDIR and RMDIR

- **MKDIR (Make Directory):** Creates a new directory【4:0†source】.
- **RMDIR (Remove Directory):** Deletes an empty directory【4:1†source】.

3. RM and RMRF

- **Purpose:** Removes files or directories.
- **Usage:**
  - `rm <file>` - Deletes a file.
  - `rm -rf <directory>` - Forcibly removes a directory and its contents recursively【4:0†source】【4:1†source】.
  - **Warning:** RMRF is dangerous in the wrong hands as it cannot be undone 【4:0†source】【4:1†source】.

4. TOUCH

- **Purpose:** Creates an empty file【4:0†source】.

File Copy and Move Commands

1. CP (Copy)

- **Purpose:** Copies files from one location to another【4:0†source】.
- **Usage:**
  - `cp <source-file> <destination-file>` - Creates a copy of the source file with a different name or location.

2. MV (Move)

- **Purpose:** Moves or renames files【4:10†source】.
- **Usage:**
  - `mv <source-file> <destination-file>` - Moves the file to the new destination with a new name, effectively deleting the original file.
  - **Analogous to Renaming:** Although called moving, it resembles renaming as it deletes the original【4:10†source】.

Links: Hardlinks and Softlinks

1. Hardlink

- **Purpose:** References another file on the same filesystem【4:3†source】【4:10†source】.
- **Characteristics:**
  - Multiple filenames can point to the same inode (storage location), ensuring data consistency across different names【4:7†source】【4:12†source】.
  - Deletion of one link does not affect the data, provided another link remains【4:12†source】【4:14†source】.
  - **Analogy:** Like one city known by two different names; deletes affect the filesystem entry but not the data【4:19†source】.

2. Softlink (Symbolic Link)

- **Purpose:** Creates a link to another file or directory.
- **Characteristics:**
  - Acts as a shortcut and can span filesystems【4:19†source】.
  - Deleting the original file breaks the link, leaving the soft link without data access【4:19†source】.

Additional Commands

1. CAT

2. ECHO

- **Purpose:** Prints a string to the terminal.
- **Usage:** `echo "Hello World"` prints "Hello World"【4:9†source】.

3. CLEAR

- **Purpose:** Clears the terminal screen, offering an unobstructed view for continued operations【4:9†source】.

4. HISTORY

- **Purpose:** Views the list of previously executed commands in chronological order, which is helpful for auditing and troubleshooting【4:9†source】.

By understanding and mastering these commands and concepts, you can effectively navigate and manage Linux systems. Always exercise caution with powerful commands like `rm -rf`, and leverage the wealth of syntax variations for efficient command-line operations.