Linux Basics and Scripting Revision Notes

This revision note covers the key topics discussed in the class, focusing on Linux commands and scripting, as well as providing practical examples.

Overview

The class involves learning about the Linux filesystem, basic commands, and shell scripting. We'll go through various commands that help manage directories, create files, and automate tasks using shell scripts.

Linux Filesystem Basics

Directories and Files

- **Directories and Files:**
  - In Linux, files are the fundamental components much like books in a library, forming a structured hierarchy.
  - Directories are akin to the shelves on which books are organized. They are used to store files in a structured manner.
- **Current Directory Commands:**
  - `pwd` (Print Working Directory): Displays the current directory you are working in【10:0†source】.
  - `ls` : Lists files and directories within the current directory【10:0†source】.

Basic Commands

- **Creating Directories:**
  - `mkdir` : Used to create new directories. For example, `mkdir Geography` creates a directory named "Geography"【10:0†source】.

you inside the "Geography" directory【10:0†source】.
    - `cd ..`: Moves you one directory up in the hierarchy【10:0†source】.
- **Creating Files:**
    - `touch`: Creates an empty file. For example, `touch file.txt` creates a file named "file.txt"【10:0†source】.
- **Removing Directories and Files:**
    - `rm`: Remove files.
    - `rm -rf`: Remove directories and their contents recursively【10:0†source】.

Understanding the Root and Home Directories

- **Root Directory (/):**
    - The top level of the file system hierarchy. All files and directories stem from the root directory【10:0†source】.
- **Home Directory (~):**
    - A user-specific directory, typically used for storing personal files and configurations【10:0†source】.

Shell Scripting Basics

Writing a Simple Shell Script

1. **The Shebang (`#!`):**
    - A special line at the top of a script (e.g., `#!/bin/bash`) that tells the system what interpreter to use to execute the script【10:14†source】.
2. **Creating a Script:**
    - Use `vi` or `vim` editors to create script files (`vi script.sh`).
    - Insert executable commands within the file and save【10:10†source】.
3. **Executing a Script:**
    - Use `chmod +x script.sh` to make the script executable.
    - Execute with `./script.sh`【10:14†source】.

Example Script

A simple script to create directories and files:

```
#!/bin/bash
mkdir B
cd B
mkdir Civics
mkdir Physics
cd Civics
touch book3.txt
cd ..
cd Physics
touch book4.txt
```

Common Script Commands

- **Exiting Vim:**
  - Use `:wq` to write and quit.
  - Use `:q!` to quit without saving【10:15†source】.
- **Permissions:**
  - `chmod 700 script.sh` provides execution permissions【10:15†source】.

CPU, Memory, and Disk Analogy

- **CPU (Central Processing Unit):**
  - Compares to multiple burners on a stove allowing for multitasking .
- **RAM (Random Access Memory):**
  - Acts like the cooking space, holding data temporarily for quick access .

This guide provides a structured overview of the Linux basics and scripting as discussed in class, offering practical command examples and conceptual analogies for enhanced understanding.