Assignments: explain absolute and relative path

In Linux (and other operating systems), paths specify the location of files and directories. There are two types of paths: absolute and relative. Understanding the difference between them is crucial for effectively navigating and managing the file system.

Absolute Path:

An absolute path specifies the complete path to a file or directory, starting from the root directory (/). It provides the full address, ensuring that no matter where you currently are in the file system, you can always locate the file or directory using this path.

- **Example: '/home/user/Documents/demo.txt'**
- **Explanation:** This path starts from the root directory (/) and specifies every directory down the hierarchy until it reaches the '**demo.txt'** file. It does not depend on the current working directory.

Relative Path:

A relative path specifies a file or directory location relative to the current working directory. It does not start from the root directory but rather from your current position in the file system.

- **Example: 'Documents/demo.txt'**
- **Explanation:** If your current working directory is '**/home/user,'** this path points to the '**demo.txt'** file inside the '**Documents'** directory, which is a subdirectory of your current directory.

Special Symbols in Relative Paths:

- '.' (dot): Refers to the current directory.
- '..' (double dot): Refers to the parent directory.
- **Example with '..':** If your current working directory is '**/home/user,'** the path '**../admin/logs'** would refer to '**/home/admin/logs'.**

Examples:

1. **Navigating Using Absolute Path:**
    - Command: '**cd /var/log'**
    - Explanation: This command changes the current directory to '**/var/log**', regardless of your current location in the file system.
2. **Navigating Using Relative Path:**
    - Command: '**cd ../user'**
    - Explanation: If your current directory is '**/home/admin'**, this command changes the directory to '**/home/user**'.

When to Use

- **Absolute Path:**
  - Use when you need to specify the location unambiguously.
  - Useful in scripts or commands where you want to ensure the exact file or directory is referenced.
- **Relative Path:**
  - Use for convenience when working interactively in the terminal.
  - Useful when you know the current working directory and want to quickly navigate within nearby directories.

By understanding and using both absolute and relative paths appropriately, you can efficiently manage files and directories in a Linux environment.