kobisbell.com
lab 1: organizing a file system

Overview

This lab was designed for you learn how to navigate and organize a RHEL file system. In this lab, I will have you step into the shoes of a System Administrator at Gourmet, a company that specializes in smart kitchen appliances and recipe apps. Your mission is to organize the company's file system effectively.

A well-structured file system is crucial for facilitating collaboration among various departments and I wanted to make a small-scale version of emulating such an environment. Throughout this lab, you will engage in practical tasks that simulate real-world scenarios faced by System Administrators. By organizing directories, managing files, and implementing version control, you will gain valuable skills that are essential for any IT professional.

Objectives

By the end of this lab, you will be able to:

- Create a structured directory system.
- Manage files and directories using command-line tools.
- Implement symbolic and hard links.
- Practice basic version control concepts.

Instructions

Now that we have an understanding on what we will be working with today, we can get more into organizing the file system. Make sure you read and follow the steps carefully to ensure a cohesive setup.

Part One: Setting Up the Main Directory

So, we can begin by making the main directory that will hold all the content that we are going to need for our lab. Just to keep this separate from other files on our system.

1. **Create the main directory**:
    - Create a directory named Gourmet in your home directory.
2. **Create subdirectories**:
    - Now, inside Gourmet, you should create the following subdirectories:
        - Research
        - Development
        - Marketing
        - Support

Part Two: Organizing Research Files

Now that we have our main structure, let's focus on organizing the Research department.

1. **Navigate to the Research directory**.
2. **Create subdirectories**:

    - In Research, create three subdirectories:
        - Prototypes
        - UserStudies
        - Competitors
3. **Add files**:
    - In the Prototypes directory, create the following files to document your projects:
        - smart_oven_v1.txt
        - smart_oven_v2.txt
        - smart_fridge_v1.txt

Part Three: Managing Development Files

Next, we'll turn our attention to the Development department.

1. **Change to the Development directory**.
2. **Create multiple files**:
    - Use a single command to create files named app_version_1.0.py through app_version_5.0.py. This will help you manage different versions of your applications efficiently.
3. **Create a symbolic link**:
    - Let's create a symbolic link in your home directory called current_app that points to app_version_5.0.py. This will make accessing the latest version easier.

Part Four: Structuring Marketing Assets

Now, let's organize some assets for Marketing.

1. **In the Marketing directory**, create the following subdirectories:
    - Images
    - Videos
    - Presentations
2. **In the Images directory**, create 10 empty files with names formatted as product_image_XX.png (where XX is a two-digit number from 01 to 10). This is for keeping your images organized.
3. **Organize files**:
    - Move all files with even numbers in their names to a new subdirectory called EventPhotos. This will help you keep track of specific events or promotions.

Part Five: Handling Support Documentation

Next up is organizing support documentation.

1. **Navigate to the Support directory**.
2. **Create a FAQ file**:
    - Create a file named faq.txt and add the text "Frequently Asked Questions." This will be useful for customer inquiries.
3. **Create hard links**:
    - Create three hard links to faq.txt named:
        - user_manual.txt

        - troubleshooting.txt
        - contact_info.txt

Part Six: Implementing Basic Version Control

Let’s implement some basic version control practices.

1. **Create a version control directory**:
    - In the Development directory, create a new subdirectory called Versions.
2. **Copy Python files**:
    - Copy all .py files from the Development directory to the Versions directory for easy access and management.
3. **Create a version log**:
    - In the Versions directory, create a new text file named version_log.txt.
    - Add entries for each Python file you copied, including filenames and dates to keep track of changes over time.
4. **Update symbolic link**:
    - Navigate back to your home directory and update the symbolic link to point to the latest version file in the Versions directory.

Part Seven: Cleaning Up

Finally, let’s clean up after our work and remove all the content for our next lab:

1. **Remove the Gourmet directory using the rm command with the recursive flag.**