

30 de octubre de 2024

Introducción

Este taller práctico está diseñado para reforzar los conceptos aprendidos en la clase de introducción a la línea de comandos. A través de una serie de ejercicios, aplicarás los comandos y técnicas que hemos discutido, tanto en Linux como en Windows. ¡Manos a la obra!

1. Navegación y manipulación de archivos

Ejercicio 1

Sigue estos pasos:

1. Crea un directorio llamado `proyecto_web`
2. Dentro de `proyecto_web`, crea los siguientes subdirectorios:
 - `css`
 - `js`
 - `img`
3. En el directorio raíz de `proyecto_web`, crea un archivo llamado `index.html`
4. En el directorio `css`, crea un archivo llamado `estilos.css`
5. En el directorio `js`, crea un archivo llamado `script.js`
6. Muestra el árbol de directorios de `proyecto_web`

Código base (Linux):

```
# Paso 1
mkdir -----

# Paso 2
cd -----
mkdir --- --- ---
```

```
# Paso 3
touch -----
```

```
# Paso 4
cd ---
touch -----
```

```
# Paso 5
cd ../ ---
touch -----
```

```
# Paso 6
cd ..
tree
```

Código base (Windows):

```
# Paso 1
mkdir -----
```

```
# Paso 2
cd -----
mkdir --- --- ---
```

```
# Paso 3
type nul > -----
```

```
# Paso 4
cd ---
type nul > -----
```

```
# Paso 5
cd ..\ ---
type nul > -----
```

```
# Paso 6
cd ..
tree
```

2. Visualización y edición de contenido

Ejercicio 2

Realiza las siguientes tareas:

1. Agrega el siguiente contenido al archivo `index.html`:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Mi Proyecto Web</title>
  <link rel="stylesheet" href="css/estilos.css">
</head>
<body>
  <h1>Bienvenido a mi proyecto</h1>
  <script src="js/script.js"></script>
</body>
</html>
```

2. Muestra el contenido de `index.html`

3. Agrega el siguiente contenido al archivo `estilos.css`:

```
body {
  font-family: Arial, sans-serif;
  background-color: #f0f0f0;
}
h1 {
  color: #333;
}
```

4. Muestra las primeras 3 líneas de `estilos.css`

Código base:

```
# Paso 1
echo "<!DOCTYPE html>
<html lang=\"es\">
<head>
  <meta charset=\"UTF-8\">
  <title>Mi Proyecto Web</title>
  <link rel=\"stylesheet\" href=\"css/estilos.css\">
</head>
```

```
<body>
  <h1>Bienvenido a mi proyecto</h1>
  <script src=\"js/script.js\"></script>
</body>
</html>\" > index.html
```

```
# Paso 2
--- index.html
```

```
# Paso 3
echo \"body {
  font-family: Arial, sans-serif;
  background-color: #f0f0f0;
}
h1 {
  color: #333;
}\" > css/estilos.css
```

```
# Paso 4
--- -n 3 css/estilos.css
```

3. Manipulación de archivos y directorios

Ejercicio 3

Realiza las siguientes tareas:

1. Crea una copia de `index.html` llamada `index.backup.html`
2. Mueve `index.backup.html` al directorio `img`
3. Elimina el archivo `script.js` del directorio `js`
4. Renombra el directorio `img` a `imagenes`
5. Muestra el nuevo árbol de directorios de `proyecto_web`

Código base (Linux):

```
# Paso 1
--- index.html index.backup.html
```

```
# Paso 2
--- index.backup.html img/
```

```
# Paso 3
--- js/script.js
```

```
# Paso 4
--- img imagenes

# Paso 5
tree

Código base (Windows):

# Paso 1
---- index.html index_backup.html

# Paso 2
---- index_backup.html img\

# Paso 3
---- js\script.js

# Paso 4
ren img imagenes

# Paso 5
tree
```

4. Búsqueda y filtrado

Ejercicio 4

Realiza las siguientes tareas:

1. Busca todos los archivos con extensión `.html` en el directorio `proyecto_web` y sus subdirectorios
2. Muestra
3. Muestra todas las líneas que contengan la palabra `color`.^{en} el archivo `estilos.css`
4. Cuenta el número de archivos en el directorio `proyecto_web` y sus subdirectorios

Código base (Linux):

```
# Paso 1
---- -name "*.html"

# Paso 2
---- "color" css/estilos.css
```

```
# Paso 3
---- -R | wc -l
```

Código base (Windows):

```
# Paso 1
dir /s /b *.html
```

```
# Paso 2
findstr "color" css\estilos.css
```

```
# Paso 3
dir /s /b /a-d | find /c /v ""
```

5. Redirección y tuberías

Ejercicio 5

Realiza las siguientes tareas:

1. Crea un archivo llamado `proyecto_info.txt` que contenga una lista de todos los archivos en `proyecto_web` y sus subdirectorios
2. Agrega una línea al final de `proyecto_info.txt` que diga "Fecha de creación: "seguida de la fecha actual
3. Muestra el contenido de `proyecto_info.txt`, pero solo las primeras 5 líneas
4. Cuenta cuántas veces aparece la palabra "proyecto." en `proyecto_info.txt`

Código base (Linux):

```
# Paso 1
---- -R > proyecto_info.txt
```

```
# Paso 2
echo "Fecha de creación: $(date)" >> proyecto_info.txt
```

```
# Paso 3
---- -n 5 proyecto_info.txt
```

```
# Paso 4
---- -i proyecto_info.txt | wc -l
```

Código base (Windows):

```
# Paso 1
dir /s /b > proyecto_info.txt

# Paso 2
echo Fecha de creaci n: %date% >> proyecto_info.txt

# Paso 3
---- /h proyecto_info.txt

# Paso 4
findstr /i /c:"proyecto" proyecto_info.txt | find /c /v ""
```

6. Conclusión

¡Felicidades por completar este taller práctico! Has aplicado una variedad de comandos y técnicas de línea de comandos tanto en Linux como en Windows. Recuerda que la práctica constante es clave para dominar estas habilidades. A medida que continúes con tu aprendizaje en desarrollo web, encontrarás que estas habilidades de línea de comandos son invaluable para tu trabajo diario.

Desafío extra

Para practicar aún más, intenta lo siguiente:

- Crea un script (bash para Linux o batch para Windows) que automatice la creación de la estructura de directorios de un proyecto web básico (con carpetas para HTML, CSS, JavaScript e imágenes).
- Investiga cómo usar el comando `sed` (Linux) o `powershell` (Windows) para realizar reemplazos de texto en archivos.
- Explora cómo usar variables de entorno en la línea de comandos y cómo pueden ser útiles en el desarrollo web.