Escuela de Ingeniería Informática

Dpto. de Lógica y Programación

Algoritmos y Estructuras de Datos

NRC 26308

Proyecto 2: Terminal Unix

Este proyecto consiste en el desarrollo de una aplicación que simula una **terminal Unix** básica. La aplicación deberá **cargar y gestionar un sistema de archivos** utilizando una estructura de datos dinámica y jerárquica. La interacción con el usuario se realizará a través de un conjunto de **comandos Unix** fundamentales.

Requisitos Funcionales

1. **Carga del Sistema de Archivos Inicial:**
    - La aplicación debe **leer un archivo de texto plano** que contendrá la definición inicial de carpetas y archivos.
    - Este contenido se almacenará en una **estructura dinámica de datos jerárquica** (por ejemplo, árboles, listas enlazadas anidadas) que represente el sistema de archivos.

2. **Interfaz de Usuario y Comandos:**
    - Al iniciar, la terminal imprimirá la **dirección raíz (`/`)** y esperará la interacción del usuario.
    - Se deben implementar los siguientes comandos Unix:
        - `cd <ruta>`: Permite **cambiar el directorio actual**. Al cambiar de directorio, se debe cambiar la ruta que se muestra por consola. Ejemplo:

        - `ls`: Permite **ver por consola** los nombres de los directorios y archivos que están contenidos en el directorio actual.
        - `mkdir <nombre_carpeta>`: Permite **crear una nueva carpeta** en el directorio actual.
        - `rm <nombre_archivo_o_carpeta>`: Permite **eliminar un archivo o una carpeta**. Si es una carpeta, se eliminará junto con su contenido.
        - `touch <nombre_archivo>`: Permite **crear un nuevo archivo de texto plano** en el directorio actual.
        - `exit`: Finaliza la ejecución de la terminal.

    - Se debe permitir el cambio de nombre a las carpetas y archivos de texto plano. Además de implementar un pequeño editor de texto para mostrar por consola y editar el contenido de los archivos de texto.
    - Todos los comandos a implementar en la terminal deben soportar **rutas absolutas** y **rutas relativas**

3. **Manejo de Archivos y Carpetas:**
    - Solo se considerará la **creación de carpetas y archivos de texto plano**.
    - Las **carpetas pueden contener otras carpetas y archivos de texto plano**.

Consideraciones Técnicas

- La estructura de datos elegida debe permitir una **navegación y manipulación eficiente** del sistema de archivos.
- La implementación de los comandos debe **validar las entradas del usuario** y manejar casos de error (por ejemplo, ruta no encontrada, archivo ya existente).

Evaluación del proyecto

Para la evaluación del proyecto se realizará una revisión durante la semana 15 (a partir del 07/07/2025), donde se les pide lo siguiente:

- Diseñar la estructura del archivo de texto donde se almacenarán las carpetas y archivos que se verán en la terminal
- Implementación de una estructura de datos jerárquica para operar sobre el sistema archivos a través de la terminal según lo establecido previamente.
- Al ejecutar el comando `exit`, se deben guardar los cambios realizados a la estructura en el mismo archivo de texto que se leyó al principio de la ejecución de la terminal. De esta forma, los cambios realizados tendrán persistencia en el tiempo.

Detalles a tomar en cuenta

- El lenguaje de programación que debe utilizar para desarrollar la aplicación es C++.
- En las revisiones del proyecto, el mismo deberá ser compilado usando el editor de código Visual Studio Code. Se va a evitar el uso de cualquier otro entorno.
- La aplicación debe ser desarrollada para ser usada por consola.
- Se deben utilizar las estructuras de datos dinámicas vistas en clases para el manejo de los datos. En caso de no utilizar estas estructuras o usar bibliotecas predefinidas por el lenguaje de programación para el uso de estructuras dinámicas, no será corregido el proyecto.
- Se deberá hacer uso del sistema de control de versiones Git y alojar su código en GitHub en un repositorio privado. Se tomará en cuenta para la evaluación el uso de buenas prácticas en los *commits* y el manejo de las ramas del repositorio. DE NO HACER USO GIT, NO SE REVISARÁ EL PROYECTO.
- Para las revisiones del proyecto, solo se tomará en cuenta el código que se encuentre en la rama principal del repositorio y que haya sido subido a la misma antes del día de la corrección. NO SE REVISARÁ CÓDIGO EN OTRA RAMA FUERA DE LA PRINCIPAL.

- También se tomará en cuenta las buenas prácticas aplicadas al código desarrollado, como la documentación del mismo, el estándar utilizado para nombrar variables y funciones y el buen uso de las funciones y librerías.