import tkinter as tk

from tkinter import messagebox, simpledialog

class Inode:

    def \_\_init\_\_(self, name, is\_dir=False):

        self.name = name

        self.is\_dir = is\_dir

        self.contents = {}

        self.permissions = {'user': 'rwx', 'group': 'r-x', 'others': 'r--'}

class FileSystemGUI:

    def \_\_init\_\_(self, root):

        self.fs = Inode('/')

        self.current\_dir = self.fs

        self.root = root

        self.root.title("File System GUI")

        self.label = tk.Label(root, text="File System GUI", font=("Helvetica", 16))

        self.label.pack(pady=10)

        self.create\_button = tk.Button(root, text="Create", command=self.create)

        self.create\_button.pack(pady=5)

        self.delete\_button = tk.Button(root, text="Delete", command=self.delete)

        self.delete\_button.pack(pady=5)

        self.list\_button = tk.Button(root, text="List", command=self.list)

        self.list\_button.pack(pady=5)

        self.change\_dir\_button = tk.Button(root, text="Change Dir", command=self.change\_dir)

        self.change\_dir\_button.pack(pady=5)

        self.read\_button = tk.Button(root, text="Read", command=self.read)

        self.read\_button.pack(pady=5)

        self.write\_button = tk.Button(root, text="Write", command=self.write)

        self.write\_button.pack(pady=5)

        self.text\_area = tk.Text(root, width=40, height=10, wrap=tk.WORD)

        self.text\_area.pack(pady=10)

    def create(self):

        name = simpledialog.askstring("Create", "Enter name:")

        if name is not None:

            is\_dir = messagebox.askyesno("Create", "Is it a directory?")

            new\_inode = Inode(name, is\_dir)

            self.current\_dir.contents[name] = new\_inode

            messagebox.showinfo("Result", f"{'Directory' if is\_dir else 'File'} '{name}' created.")

    def delete(self):

        name = simpledialog.askstring("Delete", "Enter name:")

        if name in self.current\_dir.contents:

            del self.current\_dir.contents[name]

            messagebox.showinfo("Result", f"Deleted '{name}'.")

        else:

            messagebox.showerror("Error", f"'{name}' not found.")

    def list(self):

        contents = ', '.join(self.current\_dir.contents.keys())

        self.text\_area.delete('1.0', tk.END)

        self.text\_area.insert(tk.END, contents)

    def change\_dir(self):

        name = simpledialog.askstring("Change Dir", "Enter directory name:")

        if name in self.current\_dir.contents and self.current\_dir.contents[name].is\_dir:

            self.current\_dir = self.current\_dir.contents[name]

            messagebox.showinfo("Result", f"Changed directory to '{name}'.")

        else:

            messagebox.showerror("Error", f"'{name}' is not a valid directory.")

    def read(self):

        name = simpledialog.askstring("Read", "Enter name:")

        if name in self.current\_dir.contents:

            contents = self.current\_dir.contents[name].contents

            self.text\_area.delete('1.0', tk.END)

            self.text\_area.insert(tk.END, contents)

        else:

            messagebox.showerror("Error", f"'{name}' not found.")

    def write(self):

        name = simpledialog.askstring("Write", "Enter name:")

        if name in self.current\_dir.contents:

            content = simpledialog.askstring("Write", f"Enter content for '{name}':")

            if content is not None:

                self.current\_dir.contents[name].contents = content

                messagebox.showinfo("Result", f"Content written to '{name}'.")

        else:

            messagebox.showerror("Error", f"'{name}' not found.")

if \_\_name\_\_ == "\_\_main\_\_":

    root = tk.Tk()

    app = FileSystemGUI(root)

    root.mainloop()