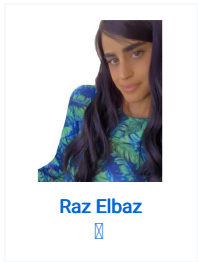


TREE-tool : stree - File Tree Printing Tool

implement the "tree" tool in a long mode, printing the file tree with permissions, user and group, and size information.

written by 



Description

The "stree" tool is a custom implementation of the "tree" command, developed as a student project. It is designed to provide a detailed and visually appealing representation of the file tree structure within a specified directory.

When executed, "stree" recursively traverses the directory and its subdirectories, printing the name of each file and directory encountered. Additionally, it displays various information for each entry, including permissions, user, group, and size.

To enhance the readability of the output, "stree" utilizes special characters to represent the hierarchical relationship between directories. The tool also applies different colors to highlight specific types of files, making it easier to identify directories, executable files, image files, and more.

Task Description

1. Implement the "stree" tool to print the file tree structure in long mode.
2. Ensure that the output includes the following information for each file/directory:
 - Permissions: The permissions of the file/directory.
 - User: The user associated with the file/directory.
 - Group: The group associated with the file/directory.
 - Size: The size of the file/directory.
3. Start printing the tree structure from the current folder and move downward.
4. Copy the special tree characters (-, |, etc.) from the original tree terminal window for accurate representation.
5. Submit the following files:
 - The implementation code for the "stree" tool.
 - A working makefile to build the tool.

Getting Started

To use the "stree" tool, follow these steps:

1. Build the tool using the provided Makefile:

```
make
```

2. Run the "stree" tool with the desired directory path:

```
./stree <directory-path>
```

♥ Replace `directory-path` with the path to the directory for which you want to display the file tree or write: `.`

Examples

Here are some example usages of the "stree" tool:

1. Print the file tree for the current directory:

```
./stree .
```

2. Print the file tree for a specific directory:

```
./stree /path/to/directory
```

♥ Replace `/path/to/directory` with the path to the directory for which you want to display the file tree or write: `.`

Output

The "stree" tool will display the file tree structure, along with the following details for each file or directory:

- Permissions
- Owner
- Group
- Size

The tool uses various colors to distinguish different types of files:

- Blue: Directories
- Bright Yellow: Pipes
- Green: Executable files
- Dark Yellow: Block or character device files
- Purple: JPG, PNG, or GIF files
- Red: .tar.gz, or .zip files

The "stree" tool will display the file tree structure in the following format:

```
.
├─ [permissions  user      group      size]  name
|   └─ [permissions  user      group      size]  name
|       └─ [permissions  user      group      size]  name
├─ [permissions  user      group      size]  name
|   └─ [permissions  user      group      size]  name
└─ [permissions  user      group      size]  name

X directories, Y files
```

- Each line represents a file or directory in the tree.
- The permissions field displays the file's permissions, including the file type and access permissions for the owner, group, and others.
- The user field shows the owner of the file.
- The group field displays the group associated with the file.
- The size field represents the size of the file in bytes.
- The name field shows the name of the file or directory.

Note: The name of directories will be displayed in blue, while special file types will have different colors.

At the end of the output, the tool will display the total number of directories and files found in the specified directory.

Example Output

```
.
├─ [drwxr-xr-x  user1  group1      4096]  directory1
|   └─ [-rw-r--r--  user1  group1      1024]  file1.txt
|       └─ [-rw-r--r--  user1  group1      512]  file2.txt
├─ [drwxr-xr-x  user2  group2      4096]  directory2
|   └─ [-rw-r--r--  user2  group2      256]  file3.txt
└─ [-rw-r--r--  user3  group3      128]  file4.txt

3 directories, 4 files
```