Tarea 4 - Análisis de Algoritmos y Estructura de Datos

Leonardo Espinoza Ortiz
Universidad de Santiago de Chile

2-2022

1-Introducción

El objetivo de esta tarea es poner en práctica los conocimientos adquiridos sobre organización de datos basada en árboles para un problema común: implementación de sistemas de archivos. Así, en esta tarea se deberá crear un pequeño sistema de archivos (filesystem), al que se llamará treeSO, que permita organizar carpetas y archivos a través de un conjunto de comandos, muy similar al sistema de archivos Linux.

La implementación de treeSO debe utilizar una estructura de árbol, y el sistema de archivos deberá soportar los siguientes comandos:

- cd: comando para ubicarse en una carpeta

```
Sintaxis

    cd <carpeta>

Ejemplo

    cd mis_documentos
```

- ls: comando para listar el contenido de una carpeta

```
Sintaxis

    ls <carpeta>

Ejemplo

    ls
    /mis_documentos/tareas
```

- mkdir: comando para crear una carpeta

```
Sintaxis

    mkdir <carpeta>

Ejemplo

    mkdir tareas
```

- mkfile: comando para crear un archivo en cierta carpeta

```
Sintaxis

    mkfile <carpeta> <file>

Ejemplo

    mkfile
    /mis_documentos/tareas
    file1
```

- rm: comando para eliminar un archivo o una carpeta. En este último caso, elimina la carpeta con todo su contenido

```
Sintaxis

    rm <carpeta>

Ejemplo

    rm tareas
```

- tree: comando para mostrar el contenido de una cierta carpeta, en formato de árbol

```
Sintaxis

    tree <carpeta>

Ejemplo

    tree tareas
```

- find: comando para buscar carpetas o archivos desde cierta carpeta. El resultado es una lista de las carpetas y archivos encontrados

```
Sintaxis

    find  <carpeta>  <nombre
    a buscar>
```

Ejemplo

```
find     /mis_documentos
file1
```

- exit: comando para salir de treeSO

2-Solución propuesta
La solución propuesta será crear un árbol genérico que sea capaz de soportar todos los comandos y pueda representar de forma correcta el sistema de archivos solicitado.

Se creará un TDA item que tendrá los siguientes atributos:
-Nombre de la carpeta o archivo.
-Tipo, 1 si es carpeta, 0 si es archivo.
no.
-Archivo no debe poseer un nodo hijo, por lo tanto es equivalente a un nodo hoja.
-Nombre del programa principal: “TreeSo”

-Para la implementación de este:
En cada función se verificará el tipo de ítem que se ingresa para verificar que la función se implementa al tipo de ítem correcto

Funciones a representar:

-cd: ubicarse en una carpeta.
Lo primero será utilizar un verificador de si la carpeta destino es una carpeta, luego se recorre el árbol y se apunta a esa carpeta.

-ls: listar contenido de una carpeta.
Lo primero, verificar si el nodo que se revisará es una carpeta, luego se recorre desde esa carpeta hacia los nodos hijos siguientes.

-mkdir:crear carpeta
Se crea la carpeta en el nodo en que se encuentre apuntando el buscador.
Se crea un nodo de tipo carpeta.

-mkfile: crear archivo en cierta carpeta.
Se busca la carpeta bajo su nombre y se verifica si el item es una carpeta.
Luego, se crea un nodo archivo que será el “hijo” de la carpeta.

-rm: eliminar archivo o carpeta
Se elimina el archivo por su nombre, y se verifica si el nodo es tipo archivo.
En caso de ser carpeta, se elimina la carpeta bajo su nombre y todos los nodos hijos que posea esta carpeta.
También se verifica si el nodo es tipo carpeta.

-tree: mostrar el contenido de la carpeta en formato árbol.
Se muestra el contenido de la carpeta representada como un árbol, y se verifica que este ítem sea carpeta.

-find: buscar carpetas y archivos desde cierta carpeta.
Se buscan los archivos desde cierta carpeta(nodo tipo carpeta) y desde ese nodo, se recorre sus hijos buscando una coincidencia a lo que se quiere buscar.

-exit: terminar el programa.

3-Resultados y análisis

Si bien no se completó en su totalidad la tarea propuesta, se logró representar de forma clara la solución propuesta y su representación.
Resultados no pueden ser expresados en código debido a que no se implementó de forma completa.

4-Conclusiones:
Se puede concluir que la estructura de los árboles se encuentra presente en muchos análisis de algoritmos e implementaciones en la vida real.
Es una herramienta muy útil a la hora de crear representaciones aptas para este tipo de problemas.

Anexo manual de usuario