

Implementación de un sistema de archivos jerárquico simplificado (MiniFS)

Se desea implementar un sistema de archivos jerárquico simplificado, representado como un árbol cuyos nodos pueden ser archivos o directorios.

Justificar las invariantes de representación y costos tanto de las funciones como de memoria.

```
struct FSNode
{
    string nombre;
    bool isFile;
    FSNode **children;
    int childCount;
};
```

```
struct FSStr
{
    FSNode *current;
    Stack path;
    int capacidad;
};
```

Funcionalidades

Implementará las siguientes funciones sobre el sistema de archivos:

`FSNode* createNode(string name, bool isFile, int capacidad):`
Crea un nuevo nodo del sistema de archivos (archivo o directorio).

`bool hayNombre(FSNode* node, string name):`
Verifica si existe un hijo con el nombre dado en un nodo directorio.

`void mkdir(string name, MiniFS fs)`
Agrega un nuevo directorio al nodo actual si hay espacio y no hay otro nodo con el mismo nombre.

`void touch(string name, MiniFS fs)`
Agrega un nuevo archivo al nodo actual si hay espacio y no hay otro nodo con el mismo nombre.

`void cd(int index, MiniFS fs)`
Cambia el nodo actual al hijo de índice dado.

`void cdUp(MiniFS fs)`
Vuelve al directorio anterior, si no está en la raíz.

`void ls(MiniFS fs)`
Lista los nombres de los hijos del nodo actual.

`void rm(int index, MiniFS fs)`

Elimina un archivo del nodo actual en el índice dado, reacomodando los elementos.

`int countFiles(MiniFS fs)`

Devuelve la cantidad total de archivos (no directorios) en el subárbol desde el nodo actual.

`void cdPath(MiniFS fs, string* path, int len)`

Cambia el directorio actual siguiendo una secuencia de nombres de subdirectorios.

`string currentName(MiniFS fs)`

Devuelve el nombre del nodo actual.

`bool isFile(MiniFS fs)`

Indica si el nodo actual es un archivo.

`int childCount(MiniFS fs)`

Devuelve cuántos hijos tiene el nodo actual.

`void print(string content)`

Imprime un string por pantalla.