

Práctica 3- Sistemas Operativos 2

Introducción:

Esta práctica consiste en realizar un menú en que el usuario pueda interactuar con el árbol via terminal. Las opciones que el usuario puede seleccionar son las siguientes:

- 1) El usuario puede crear un árbol dando la dirección de un fichero txt que contenga las direcciones de los ficheros pdf.
- 2) En esta opción el usuario puede salvar el árbol en un fichero binario creado por el propio usuario.
- 3) El usuario puede recuperar un árbol ya guardado en disco dando la dirección del fichero binario que tenga el contenido del árbol.
- 4) Esta opción permite buscar una palabra en el árbol creado y el resultado dado por la aplicación es el número de veces que esta palabra sale en el árbol.
- 5) Permite salir de la aplicación y recuperar la memoria utilizada durante la aplicación.

Funcionamiento de la aplicación:

La aplicación se compila por medio de la función make y su llamada se hace con el siguiente comando:

```
~./main
```

Al iniciar la aplicación el usuario tiene que elegir una de las opciones dadas.

El funcionamiento de la primera opción del menú es básicamente:

- 1) Lectura del fichero txt línea por línea

Si es la primera línea se salva en un variable de tipo string que se usa para leer los demás ficheros pdf.

- 2) Se leen todas las direcciones de los ficheros PDF y se salva en un vector.

3) Hace la llamada pipe con todos los ficheros usando el vector y guarda palabra por palabra en el árbol.

Si es una palabra compuesta se descompone en dos palabras y guarda las dos palabras en el árbol. Si la palabra ya existe en el árbol se incrementa el número de veces que la palabra aparece.

El funcionamiento de la tercera opción se resumen en:

- 1) Leer el nombre del fichero binario
- 2) Abrir el fichero usando la función de `c fopen`.

Si hay problema en leer el fichero se notifica al usuario y se cierra la aplicación.

- 3) Ser ler el número mágico del fichero con la función `fread`.
- 4) En seguida se leer el número de nodos guardados
- 5) Para cada nodo guardado se leer el número la longitud del string key y el número de veces que aparece en el árbol.
- 6) Se inserta en el árbol la información recuperada.

En la cuarta opción se pide al usuario que introduzca una palabra y la aplicación busca en el árbol la palabra dada por el usuario, si no la encuentra se imprime por pantalla que no se ha podido encontrar la palabra y en caso contrario se imprime el número de veces que aparece la palabra en el árbol binario. Esta funcionalidad es muy útil para probar el funcionamiento de la aplicación y detectar posibles errores de formatación de caracteres.

Pruebas realizadas y problemas encontrados:

Las pruebas realizadas fueron muchas al decorrer de la implementacion del codigo ya que el lenguaje c es muy poco informático según lo que está pasando. He probado con los ficheros PDF que está en la carpeta pdfs y algunos de mi propia librería de archivos. Las pruebas fueron des de poner ficheros con pocos caracteres en el árbol hasta ficheros muy largos para comprobar que no se hacía accesos inválidos a la memoria. También se ha realizado pruebas de formatación de strings.

Problemas en la implementación:

- El principal problema fue al escribir el fichero binario. Por algún motivo no se puede leer el fichero que se escribe con la función `writeTree`. Por lo tanto la funcionalidad 2 del menú no funciona. Aunque no este esta funcionalidad la aplicación permite que el usuario lea el fichero binario y ponga los datos en el árbol.
- El problema con el `malloc` al parecer no se reservaba bien el espacio de memoria con el código de la práctica anterior y me saltaba errores de asignación de

espacio de la función malloc. Tras un día de búsqueda intensiva en todos los malloc que tiene la aplicación la solución fue hacer la liberación de espacio usando un tamaño mayor que el del string.

- Otro problema que no se ha podido solucionar ha sido el de doble guión (long Dash en Inglés). palabras que tienen (—) fueron puestas en el árbol por la función ispunct de c no detectar como un carácter de puntuación, además de ser un carácter doble por lo cual en mi código es muy complicado de detectar y remover de la cadena de caracteres.