

Práctica 3

GoT: Gestión de Personajes de Game of Thrones

Algoritmos y Estructuras de Datos - Grado en Ingeniería Informática (USC)

7 de octubre de 2024

Objetivos

En esta práctica trataremos de:

- Reutilizar los TADs conocidos (árbol binario de búsqueda, lista, cola).
- Valorar qué TAD es el más adecuado para el diseño de las diferentes tareas de un programa.

1. Versión 1 de la práctica

En esta primera versión de la práctica utilizaremos el TAD árbol binario de búsqueda para guardar los datos de un conjunto de personajes. Los datos que tendremos para cada personaje son: name, house, royal y varias listas: parents, siblings y killed.

El programa se iniciará creando la base de datos de personajes vacía. Para la base de datos se utilizará un TAD árbol binario de búsqueda que pueda almacenar tantos personajes como sea necesario. A continuación se ofrecerá el menú de selección de opciones que se repetirá continuamente hasta que el usuario elija la opción de salir. Por último, se destruirá la base de datos. En el menú se mostrarán las siguientes opciones:

- **A/a. Añadir personaje:** esta opción permitirá añadir un nuevo personaje a la base de datos. Al añadir un nuevo personaje se preguntará su nombre, su casa, si es de familia real y se rellenarán varias listas: padres, hermanos, y personas que ha asesinado. Para cada lista se preguntará el nombre del personaje, finalizando el proceso de rellenarla cuando el nombre del personaje sea "fin".
- **L/l. Listado alfabético de personajes:** Se imprimirán los datos de los personajes de la base de datos en orden alfabético. También se imprimirá de forma detallada la información de **todas** sus características.
- **E/e. Eliminar personaje:** esta opción permitirá eliminar un personaje de la base de datos. Se preguntará su nombre y, si está en la base de datos, se informará de su baja. Si no está en la base de datos, también se informará por pantalla.
- **S/s. Salir:** esta opción sale del menú, finalizando el programa.

Temporización recomendada

A continuación se presenta una temporización recomendada para realizar cada una de las fases del proyecto. Tened en cuenta que esto es sólo una estrategia de implementación: tenéis completa libertad para implementar las funcionalidades que se piden en el enunciado, siempre y cuando utilicéis el TAD abb y el TAD lista.

Fase 1: Preparación del proyecto.

En primer lugar, debéis crear un proyecto con el nombre `juegodetronos` y crear `main.c`. Todas las funciones que se llaman desde el menú principal deben estar implementadas en un archivo `juegodetronos.c` con su correspondiente interfaz `juegodetronos.h`, que iremos ampliando en las distintas versiones del programa.

Fase 2: Preparación de los TAD

El programa deberá implementar obligatoriamente el árbol y las listas indicadas con los TAD correspondientes, cuya descripción y funcionalidades se han explicado en clase de teoría. Antes de iniciar el desarrollo de la práctica debéis repasar detalladamente las diapositivas y notas de teoría de los temas correspondientes y, de forma especial, los ejemplos, para recordar el funcionamiento de los TAD.

NO PODRÁ REALIZARSE NINGUNA MODIFICACIÓN SOBRE LOS FICHEROS DE LOS TAD PROPORCIONADOS excepto las indicadas a continuación, además de las que necesitéis para añadir los `#include` necesarios para modificar los tipos de elemento de los TAD, así como definir constantes para las longitudes de las cadenas de caracteres.

Preparación del TAD abb (de personajes)

Para la base de datos de personajes se utilizará el TAD abb implementado en los ficheros adjuntos (`abb.c` y `abb.h`) en el que `TIPOELEMENTOABB` debe ser una estructura con la información del personaje: campos `name` y `house` (podéis usar un array estático definiendo su longitud como una constante), campo `royal` (tendrá un entero con valor 0 o 1), y varias `TLISTA` para almacenar los campos `parents`, `siblings` y `killed`. También debéis modificar `TIPOCLAVE` para indicar cuál va a ser vuestro campo de ordenación en el abb, si ordenamos por orden alfabético será un `char *`.

Además, debéis modificar las funciones `_clave_elem()`, `_comparar_claves()` y `_destruir_elem()` en `abb.c` para adaptarlas a vuestro `TIPOELEMENTOABB`. Recordad que para usar `strcmp()` debéis incluir la librería `string.h`.

Preparación del TAD lista (de parents, siblings y killed)

Para las listas de `parents`, `siblings` y `killed` se utilizará el TAD Lista implementado en los ficheros adjuntos (`lista.c` y `lista.h`) en el que el `TIPOELEMENTOLISTA` debe definirse como una estructura con el campo `nameP` (podéis usar un array estático, pero entonces debéis definir la longitud como una constante).

Fase 3: Implementación del menú

Por último, deberéis implementar el menú del programa. El menú debe funcionar con letras mayúsculas o minúsculas. Las opciones del menú que debería haber implementadas son las siguientes:

- **A/a. Añadir personaje:** tendréis que llamar a la función `anhadirPersonaje()` que realiza la toma de datos por teclado de la información del personaje a añadir a la base

de datos. Tenéis que tener en cuenta que debéis implementar un bucle para poder tomar información sobre las personas de las distintas listas mientras el nombre de la misma sea distinta de “fin” (puede resultar útil aquí la función `strcmp()` incluida en la librería `string.h`).

- **L/l. Listado alfabético de personajes:** Tenéis que tener en cuenta que las funciones para imprimir podéis implementarlas en cadena, es decir, `listadoPersonajes()` llamaría a `_imprimirPersonaje()` e `_imprimirPersonaje()` llamaría varias veces a `_imprimirLista()`.
- **E/e. Eliminar personaje:** tendréis que llamar a la función `eliminarPersonaje()` que solicitará el nombre del personaje a eliminar y, si está en la base de datos, procederá a eliminarlo, informando por pantalla del resultado de la acción.
- **S/s. Salir:** tendrá como efecto salir del bucle del menú.

Una vez salimos del menú, habrá que proceder con la destrucción de la base de datos de memoria.

Ejemplo de ejecución versión 1

A continuación se muestra un ejemplo de ejecución del programa. Tened en cuenta que el formato de vuestro menú puede diferir de lo aquí expuesto.

```
1 > ./juegodetronos
2
3 Bienvenid@ al programa de gestion de personajes GoT
4 A. Anhadir personaje
5 L. Listar personajes
6 E. Eliminar personaje
7 S. Salir
8 Seleccione una opcion: A
9
10 Name: Aegon Targaryen
11   House (? si desconocido): Targaryen
12   Royal (0/1): 1
13   Parents (fin para terminar): Elia Martell
14   Parents (fin para terminar): Rhaegar Targaryen
15   Parents (fin para terminar): fin
16   Siblings (fin para terminar): Rhaenys Targaryen
17   Siblings (fin para terminar): Jon Snow
18   Siblings (fin para terminar): fin
19   killed (fin para terminar): fin
20
21 A. Anhadir personaje
22 L. Listar personajes
23 E. Eliminar personaje
24 S. Salir
25 Seleccione una opcion: a
26
27 Name: Aeron Greyjoy
28   House (? si desconocido): Greyjoy
29   Royal (0/1): 0
30   Parents (fin para terminar): fin
31   Siblings (fin para terminar): Balon Greyjoy
32   Siblings (fin para terminar): Euron Greyjoy
33   Siblings (fin para terminar): fin
34   killed (fin para terminar): fin
35
```

```

36 A. Anhadir personaje
37 L. Listar personajes
38 E. Eliminar personaje
39 S. Salir
40 Seleccione una opcion: A
41
42 Name: Aerys II Targaryen
43   House (? si desconocido): Targaryen
44   Royal (0/1): 1
45   Parents (fin para terminar): fin
46   Siblings (fin para terminar): Rhaella Targaryen
47   Siblings (fin para terminar): fin
48   killed (fin para terminar): Brandon Stark
49   killed (fin para terminar): Rickard Stark
50   killed (fin para terminar): fin
51
52 A. Anhadir personaje
53 L. Listar personajes
54 E. Eliminar personaje
55 S. Salir
56 Seleccione una opcion: L
57
58 Aegon Targaryen
59   House: Targaryen
60   Royal: Yes
61   Parents: Elia Martell, Rhaegar Targaryan
62   Siblings: Rhaenys Targaryan, Jon Snow
63
64 Aeron Greyjoy
65   House: Greyjoy
66   Siblings: Balon Greyjoy, Euron Greyjoy
67
68 Aerys III Targaryen
69   House: Targaryen
70   Royal: Yes
71   Siblings: Rhaella Targaryen
72   killed: Brandon Stark, Rickard Stark
73
74 A. Anhadir personaje
75 L. Listar personajes
76 E. Eliminar personaje
77 S. Salir
78 Seleccione una opcion: E
79
80 Nombre del personaje a eliminar: Aeron Greyjoy
81 El personaje Aeron Greyjoy ha sido eliminado de la base de datos
82
83 A. Anhadir personaje
84 L. Listar personajes
85 E. Eliminar personaje
86 S. Salir
87 Seleccione una opcion: E
88
89 Nombre del personaje a eliminar: Daenerys Targaryen
90 Este personaje no esta registrado en la base de datos
91
92 A. Anhadir personaje
93 L. Listar personajes
94 E. Eliminar personaje
95 S. Salir
96 Seleccione una opcion: 1
97
98 Aegon Targaryen

```

```
99      House: Targaryen
100     Royal: Yes
101     Parents: Elia Martell, Rhaegar Targaryan
102     Siblings: Rhaenys Targaryan, Jon Snow
103
104 Aerys III Targaryen
105     House: Targaryen
106     Royal: Yes
107     Siblings: Rhaella Targaryen,
108     killed: Brandon Stark, Rickard Stark
109
110 A. Anhadir personaje
111 L. Listar personajes
112 E. Eliminar personaje
113 S. Salir
114 Seleccione una opcion: s
115
116 Gracias por usar nuestro sistema GoT
117
118 >
```

2. Versión 2: Lectura/Grabación de la base de datos de personajes

En esta versión leeremos los datos de los personajes de un archivo que se introducirá por línea de comandos y también actualizaremos la base de datos guardando los datos actualizados en el mismo archivo.

La lectura del archivo se realizará antes del menú y la grabación se realizará al finalizar el menú, antes de destruir el árbol.

El programa debe funcionar aunque no se introduzca un archivo inicial.

Lectura de datos desde un fichero

Deberá permitirse la lectura de un fichero de personajes con sus datos y la inicialización de la base de datos a partir de este fichero. El fichero deberá tener el siguiente formato: *NAME/HOUSE/ROYAL/PARENT_1,PARENT_2,PARENT_3/SIBLING_1,SIBLING_2,SIBLING_3/KILLED_1,KILLED_2,KILLED_3/*

Un ejemplo se muestra en el siguiente fragmento (disponible para descargar en la tarea del campus virtual):

```
1 Aegon Targaryen|Targaryen|1|Elia Martell,Rhaegar Targaryen|Rhaenys
  Targaryen,Jon Snow||
2 Aeron Greyjoy|Greyjoy|0|Balon Greyjoy,Euron Greyjoy|||
3 Aerys II Targaryen|Targaryen|1|Rhaella Targaryen|Brandon Stark,Rickard
  Stark||
4 Arya Stark|Stark|0|Eddard Stark,Catelyn Stark|Robb Stark,Sansa Stark,Bran
  Stark,Rickon Stark|Polliver,Rorge,Ghita,Meryn Trant,Lothar Frey,Walder
  Frey,Petyr Baelish,The Night King,White Walker,Viserion|
```

Se asume que el fichero es siempre correcto, para no sobrecargar vuestro proyecto con comprobaciones defensivas.

Pautas sobre la función `cargar_archivo()`

Esta función debe leer cada línea de un archivo de entrada y almacenar los datos de ese personaje en el árbol con el que representáis la base de datos.

La función `cargar_archivo()` se llama desde `main()` y debe utilizar sus argumentos para comprobar que se ha llamado correctamente desde línea de comandos:

En línea de comandos (cambiad `ejecutable` por el nombre de vuestro programa y `archivo.txt` por el nombre de vuestro archivo de texto):

```
1 ./ejecutable -f archivo.txt
```

En `main()`:

```
1 int main(int argc, char** argv) {
2     TABB ArbolPersonajes;
3     ...
4     cargar_archivo(&ArbolPersonajes,argc,argv);
5     ...
6 }
```

Recordatorio de los argumentos de `main()`:

- `argc` tomará el valor 3: es el número de cadenas que se introducen por la línea de comandos: el ejecutable, la opción `-f` y el nombre del archivo.

- `argv[]` es un array donde se almacenan esas 3 cadenas:
 - `argv[0]` almacena la cadena `"./ejecutable"`
 - `argv[1]` almacena la cadena `"-f"`
 - `argv[2]` almacena la cadena `"archivo.txt"`

En `juegodetronos.c`: implementamos la función `cargar_archivo()`, que recibirá el árbol donde va a insertar los datos de los personajes que se lean desde fichero. Debe recibir también los argumentos de `main()`, como vimos en el código anterior de `main.c`, para comprobar si existe o no archivo de datos para cargar.

Para poder procesar este fichero debéis tener en cuenta las siguientes consideraciones:

- En primer lugar, debéis procesar los argumentos por línea de comandos (que el número de argumentos sea 3 como mínimo, que el segundo argumento sea `"-f"` y que el tercer argumento sea un nombre de archivo que al intentar abrirlo no de error.
- En segundo lugar, debéis leer el fichero línea a línea. Para ello podéis utilizar la función `fgets()`, que tiene el problema de que añade el retorno de carro a la cadena. Para eliminarlo, podéis utilizar la siguiente función (necesitaréis la librería `string.h`).

```
1 void _strip_line(char * linea){
2     linea[strcspn(linea, "\r\n")] = 0;
3 }
```

- En tercer lugar, debéis procesar los campos de la línea leída. Para eso, necesitaréis dos tipos de llamadas a la función `strtok()`. En el primer tipo de llamada, partiréis la línea a partir del carácter `"|"`. Para eso, es suficiente con hacer tantas llamadas a `strtok()` como campos tenga vuestro personaje, tal y como sigue:

```
1 char * name = strtok(linea, "|"); //primera llamada, se envia la
   cadena linea
2 char * house = strtok(NULL, "|"); //siguientes llamadas, el primer
   argumento es NULL
3 ...
4 char *killed_list = strtok(NULL, "|"); //ultima llamada
```

Después, como las funciones que usan `strtok()` sobrescriben la cadena original, no podemos usarla sobre las cadenas que tienen la lista de parents, siblings y killed. Esto se puede hacer con la función `strtok_r()` o `strtok_s()` que no están disponibles en todos los entornos (por ejemplo, en Windows con cygwin no están disponibles). Por tanto, escribiremos nuestra función `_splitByComma()` para dividir cada una de estas listas en elementos separados por comas y así insertarlos en la lista que corresponda.

A continuación tenéis un ejemplo de uso de `strtok()`:

```
1 char *cadena="Hola,mundo,soy,una,cadena,separada,por,comas";
2 char *token=strtok(cadena,",");
3 while(token!=NULL){
4     printf("Token:%s\n",token);
5     token=strtok(NULL,",");
6 }
```

A continuación tenéis un ejemplo de una función que hace lo mismo escrita por nosotros:

```

1 char *cadena="Hola ,mundo ,soy ,una ,cadena ,separada ,por ,comas ";
2 char *start=cadena;
3 char *end=cadena;
4
5 while(*end!='\0'){
6     if (*end==',' ){
7         *end='\0';
8         printf("Token: %s\n",start);
9         start=end+1;
10    }
11    end++;
12 }
13 printf("Token: %s\n",start); //Print the last token

```

Tenéis total libertad para utilizar alguna de estas funciones o generar vosotros una función que realice la lectura de los personajes, sólo tened en cuenta que strtok() no se puede usar sobre dos cadenas iniciales simultáneamente, es decir, no la podemos usar al mismo tiempo para separar las cadenas separadas por “|” y las cadenas separadas por “,”.

Grabación de datos en un fichero

El fichero debe actualizarse con los últimos datos disponibles por lo que, si tenéis un fichero precargado que habéis introducido por línea de comandos, debéis actualizarlo abriéndolo para escritura y escribiendo los datos del árbol de personajes. Tened cuidado con el recorrido que hacéis para escribir, pues si elegís un recorrido inorden los datos se guardarán en el archivo por orden alfabético y por tanto la siguiente vez que los leamos el árbol estará totalmente desequilibrado. El fichero de salida debe tener el mismo formato que el fichero de entrada, para poder leerse posteriormente. En caso de que no hayáis precargado un archivo por línea de comandos, debéis preguntar al usuario el nombre del fichero de salida con el que quiere guardar la base de datos, y así servirá para siguientes ejecuciones.

Ejemplo de ejecución versión 2 (lectura/escritura de archivos)

```

1 > cat personajes.txt
2 Aegon Targaryen|Targaryen|1|Elia Martell,Rhaegar Targaryen|Rhaenys
   Targaryen,Jon Snow||
3 Aeron Greyjoy|Greyjoy|0|Balon Greyjoy,Euron Greyjoy|||
4 Aerys II Targaryen|Targaryen|1|Rhaella Targaryen|Brandon Stark,Rickard
   Stark||
5 Arya Stark|Stark|0|Eddard Stark,Catelyn Stark|Robb Stark,Sansa Stark,Bran
   Stark,Rickon Stark|Polliver,Rorge,Ghita,Meryn Trant,Lothar Frey,Walder
   Frey,Petyr Baelish,The Night King,White Walker,Viserion|
6
7 > ./juegodetronos -f personajes.txt
8
9 Bienvenid@ al programa de gestion de personajes GoT
10 A. Anhadir personaje
11 L. Listar personajes
12 E. Eliminar personaje
13 S. Salir
14 Seleccione una opcion: L
15
16 Name: Aegon Targaryen

```



```

17     House: Targaryen
18     Royal: Yes
19     Parents: Elia Martell, Rhaegar Targaryen
20     Siblings: Rhaenys Targaryen, Jon Snow
21
22 Name: Aeron Greyjoy
23     House: Greyjoy
24     Parents: Balon Greyjoy, Euron Greyjoy
25
26 Name: Aerys II Targaryen
27     House: Targaryen
28     Royal: Yes
29     Parents: Rhaella Targaryen
30     Siblings: Brandon Stark, Rickard Stark
31
32 Name: Arya Stark
33     House: Stark
34     Parents: Eddard Stark, Catelyn Stark
35     Siblings: Robb Stark, Sansa Stark, Bran Stark, Rickon Stark
36     killed: Polliver, Rorge, Ghita, Meryn Trant, Lothar Frey, Walder Frey,
        Petyr Baelish, The Night King, White Walker, Viserion
37
38 A. Anhadir personaje
39 L. Listar personajes
40 E. Eliminar personaje
41 S. Salir
42 Seleccione una opcion: A
43
44 Nombre del personaje: Alton Lannister
45     House (? si desconocido): Lannister
46     Royal (0/1): 0
47     Parents (fin para terminar): fin
48     Siblings (fin para terminar): fin
49     Killed (fin para terminar): fin
50
51 A. Anhadir personaje
52 L. Listar personajes
53 E. Eliminar personaje
54 S. Salir
55 Seleccione una opcion: S
56
57 Gracias por usar nuestro sistema GoT
58
59 >
60 > cat personajes.txt
61 Aegon Targaryen|Targaryen|1|Elia Martell,Rhaegar Targaryen|Rhaenys
    Targaryen,Jon Snow||
62 Aeron Greyjoy|Greyjoy|0|Balon Greyjoy,Euron Greyjoy|||
63 Aerys II Targaryen|Targaryen|1|Rhaella Targaryen|Brandon Stark,Rickard
    Stark||
64 Arya Stark|Stark|0|Eddard Stark,Catelyn Stark|Robb Stark,Sansa Stark,Bran
    Stark,Rickon Stark|Polliver,Rorge,Ghita,Meryn Trant,Lothar Frey,Walder
    Frey,Petyr Baelish,The Night King,White Walker,Viserion|
65 Alton Lannister|Lannister|0|||
66
67 >

```