



Tarea Abril – Julio 2018

Módulo de Cotejo de *Soundtrack*

1 Introducción

A través de su tienda en línea iTunes, Apple Inc. actualmente puede vender tanto [películas](#), como sus [soundtracks](#). Sin embargo, el [sistema de recomendaciones](#) actual no es capaz de identificar el *soundtrack* de una película a menos que el usuario ya haya comprado alguna canción relacionada (es decir, no sugiere el *soundtrack* en base a la película, sino en base a otras canciones). Apple ha observado que el [31% de las personas](#) está interesada en los *soundtrack* de las películas, por lo que esta falla en su sistema de recomendaciones les cuesta millones cada año en ventas no realizadas.

Apple desea que usted implemente un programa en C capaz de dar, para cada película, sus canciones de *soundtrack* correspondientes.

2 Características del programa

Se desea que el programa reciba por *stdin* el nombre de una película, e imprima por *stdout* la lista de canciones de su *soundtrack*, cada canción en una línea. El programa no debe imprimir nada adicional: esto permitirá redirigir la entrada. El programa debe esperar un nombre nuevo de película cada vez que termine de imprimir una lista de canciones.

2.1 Archivo de entrada

Su programa debe recibir un archivo que contenga una lista de canciones y películas en el siguiente formato

```
Canción 1  Película de origen
Canción 2  Película de origen
:
Canción n  Película de origen
```

El título de la canción se separa de su película con un tabulador. Se puede colocar un tabulador en un *string* usando el caracter de escape `\t`. Encontrará información de cómo leer archivos en *K&R* §7.5.

2.2 Indización de canciones

Las canciones se deben mantener en memoria en un índice implementado como una Tabla de Hash. Una **Tabla de Hash** es una estructura que mapea identificadores (en este caso, nombres de películas) a elementos (en este caso, canciones) agrupando los identificadores en **buckets** según su Función de Hash y reduciendo así el tiempo de búsqueda (ver Figura 1). Una **Función de Hash** es cualquier función que tome un valor de un conjunto y retorne un valor en un conjunto considerablemente más pequeño (por ejemplo, tomar una letra de la 'A' a la 'Z' y retornar un número de 0 a 9). Encontrará más información en *K&R* §6.6.

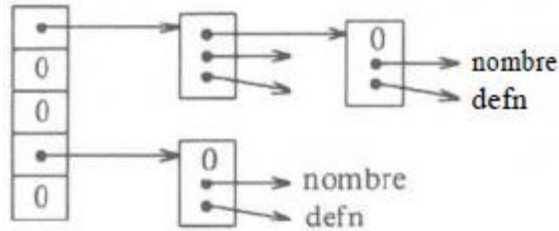


Figura 1. Diagrama conceptual de una tabla de hash de 5 *buckets*, donde 2 de esos *buckets* tienen elementos. Observe que el valor de hash correspondiente al primer *bucket* es mapeado por dos elementos, por lo que los elementos conforman una lista enlazada. Cada elemento de este ejemplo es una estructura conformada por un apuntador a siguiente, un *nombre*, y su definición (*defn*). Los ceros representan apuntadores inicializados en NULL.

(Fuente: K&R, 2ª edición).

Debe diseñar su estructura de datos para contemplar que:

- No siempre existirá una sola coincidencia (una película puede tener más de una canción de *soundtrack* disponible)
- No siempre existirá coincidencia alguna (una película puede no tener ninguna canción de *soundtrack* disponible)
- No todas serán buscadas (una película podría no estar disponible, aunque sus canciones de *soundtrack* sí lo estén)

Su implementación debe ser razonablemente eficiente. Debe solicitar (`malloc/calloc`) y liberar (`free`) memoria dinámica adecuadamente.

2.3 Configuración de la búsqueda

El programa debe ser configurable de la siguiente manera: al escribir como nombre de la película "---" (tres guiones seguidos), en vez de una lista de canciones, debe aparecer un menú con las siguientes opciones:

- Longitud máxima del nombre de la película (por defecto, 1024 caracteres)
- Longitud máxima del nombre de la canción (por defecto, 1024 caracteres)
- Número máximo de canciones a imprimir (por defecto 16 canciones)
- Eficiencia (por defecto 16 *buckets*)
- Cancelar cambio de configuración
- Salir del programa

Los **nombres** se deben truncar en el número de caracteres dado; la tabla de hash debe actualizarse de manera acorde. El **número máximo de canciones** solo afecta la impresión. Cambiar el **número de buckets**, obliga a actualizar la Tabla de Hash adecuadamente (y los parámetros de la Función de Hash para utilizar el nuevo número de *buckets*). "**Cancelar cambio de configuración**" permite cerrar el menú sin efectuar cambios. "**Salir del programa**" cierra la ejecución.


3 Evaluación

Se asignarán:

- 1 punto por correr el archivo de prueba dado
- 1 punto por correr un archivo de prueba grande
- 1 punto por código
- 1 punto por dividir el programa correctamente en archivos `.c` y `.h`
- 1 punto por su *makefile*

El programa debe correr sin errores en las computadoras del LDC.

4 Entrega

Debe entregar su código en un archivo comprimido `.tgz` (ver [guía del Profesor Figueira](#)). Deberá subirlo al [Aula Virtual de la materia](#) en la sección de  **Trabajos** hasta el martes, 15 de mayo. Sólo deberá efectuar una entrega por grupo.