

TRABAJO PRÁCTICO N.º 2 – Indexación automática de archivos MP3

1) Objetivo

El objetivo de este T.P. consiste en desarrollar un aplicativo de consola con comandos en línea de órdenes y escrito en lenguaje ANSI C, que permita construir un índice de los archivos MP3 indicados como parámetro, ordenado de acuerdo con un criterio particular y presentado en formato texto (CSV, HTML, XML).

2) Alcance

Mediante el presente T.P. se busca que el alumno adquiera y aplique conocimientos sobre los siguientes temas:

- programas en modo consola;
- argumentos en línea de órdenes (*CLA*);
- modularización;
- makefile;
- archivos de texto y binarios;
- memoria dinámica;
- punteros a función;
- tipo de dato abstracto, T.D.A. Vector. T.D.A. *ad hoc*;
- estructura del encabezado de un archivo MP3 (*ID3v1*);
- métodos de ordenamiento;
- estructura básica de un archivo CSV;
- estructura básica de un documento XML;
- estructura básica de un documento HTML (optativo).

3) Desarrollo

En este T.P. se pide escribir un programa ejecutable que genere documentación sobre pistas de audio en formato MP3.

El programa ejecutable, denominado “mp3explorer.exe” (WinXX) o “mp3explorer” (Unix), debe ser invocado de la siguiente forma:

WinXX:

```
mp3explorer -fmt <formato> -sort <criterio> -out <salida> <arch 1>...<arch N>
```

UNIX:

```
./mp3explorer -fmt <formato> -sort <criterio> -out <salida> <arch 1>...<arch N>
```

Los comandos en línea de órdenes utilizados por la aplicación son los indicados a continuación.

Formato del índice a generar

Comando	Descripción	Valor	Tipo de dato	Observaciones
-fmt	Formato del índice a generar	“csv”	Cadena de caracteres	Obligatorio
		“xml”	Cadena de caracteres	Obligatorio
		“html”	Cadena de caracteres	Optativo

Criterio de ordenamiento para los temas

Comando	Descripción	Valor	Tipo de dato	Observaciones
-sort	Criterio de ordenamiento	"name"	Cadena de caracteres	Nombre del tema (obra).
		"artist"	Cadena de caracteres	Autor de la obra.
		"genre"	Cadena de caracteres	Género de la obra.

Si se ordena, por ejemplo, por género, no se pide aplicar un nuevo criterio para todos los elementos con el mismo género.

Archivo de salida <salida>

Es la ruta del archivo índice que se desea construir.

Archivos de entrada <arch j>

Se trata del archivo MP3 j-ésimo de entrada.

Nota: Se puede asumir que los comandos en línea de órdenes estarán en cualquier orden (en pares), si esta estrategia simplifica el desarrollo de la presente aplicación, pero se debe consignar la decisión en el informe.

Operación

El programa debe analizar los archivos MP3 suministrados al programa y generar una tabla en memoria con los atributos de cada tema, para su posterior ordenamiento e impresión del índice en el formato correspondiente.

Formato de entrada

Se debe extraer la información de cada tema a partir del *frame header* del archivo MP3 mediante la lectura de los últimos 128 bytes del archivo. Se sugiere utilizar la función de biblioteca `fseek()`. Se debe trabajar con archivos ID3v1.

Formato de salida

a) Formato CSV:

El formato del documento CSV de salida a generar es:

```
<nombre de tema>|<artista>|<género>
...
<nombre de tema>|<artista>|<género>
```

Se trata de un archivo CSV sin encabezado, campos sin calificador y carácter separador "|".

b) Formato XML:

El formato del documento XML de salida a generar es:

```
<?xml version="1.0" ?>
<tracks>
  <track>
    <name>...</name>
    <artist>...</artist>
    <genre>...</genre>
  </track>
  ...
```

`</tracks> ...`

Nota: A los efectos de este T.P. se permite obviar la transformación de juegos de caracteres y entidades para un documento XML.

c) Formato HTML (optativo):

Formato HTML tabular libre, a elección del desarrollador.

4) **Restricciones**

La realización de este programa está sujeta a las siguientes restricciones:

- Se debe recurrir al uso del T.D.A. Vector para almacenar los elementos de información correspondientes a una pista de audio.
- Se debe recurrir al uso de punteros a función a fin de parametrizar la impresión del índice.
- Se deben utilizar funciones y una adecuada modularización.
- Se debe construir un proyecto mediante la utilización de *makefile*.
- Hay otras cuestiones que no han sido especificadas intencionalmente en este requerimiento, para darle al desarrollador la libertad de elegir implementaciones que, según su criterio, resulten más favorables en determinadas situaciones. Por lo tanto, se debe explicitar cada una de las decisiones adoptadas y el o los fundamentos considerados para ellas.

5) **Entrega del Trabajo Práctico**

Se debe presentar la correspondiente documentación de desarrollo. Ella se debe entregar en forma impresa y encarpetaada, de acuerdo con la numeración siguiente:

- 1) Carátula. Incluir una dirección de correo electrónico.
- 2) Enunciado (el presente documento).
- 3) Diagrama de flujo básico y simplificado del programa (1 carilla A4).
- 4) Estructura funcional del programa desarrollado (diagrama de funciones, 1 carilla A4).
- 5) Explicación de las alternativas consideradas y las estrategias adoptadas.
- 6) Resultados de la ejecución (corridas) del programa, en condiciones normales e inesperadas de entrada.
- 7) Reseña de los problemas encontrados en el desarrollo del programa y las soluciones implementadas para subsanarlos.
- 8) Conclusiones.
- 9) *Script* de compilación.
- 10) Código fuente.

6) **Bibliografía**

Se debe incluir la referencia a toda bibliografía consultada para la realización del trabajo: libros, artículos, etc.

7) **Última fecha de entrega: sábado 30/06/2018**