

Archivo de audio en formato WAV

A. Beleño, M. Angulo, M. Barragán

Universidad Sergio Arboleda

andrea.beleno01@usa.edu.co, martin.angulo01@usa.edu.co, maria.barragan02@usa.edu.co

I. INTRODUCCIÓN

WAV es una aplicación (subconjunto) de Resource Interchange File Format (RIFF) cuya función principal es almacenar archivos de audio digital, a diferencia de otros formatos de audio como MP3 y AAC, los archivos WAV no utilizan compresión, lo que significa que el tamaño del archivo será mayor pero la calidad del sonido será mejor. Los archivos de audio WAV son compatibles con una amplia variedad de software y dispositivos, incluyendo reproductores, editores y programas de grabación de audio. Otra de las características importantes de mencionar es que los archivos WAV también se pueden utilizar para almacenar información adicional como metadatos y etiquetas, lo que facilita su organización y búsqueda.

II. MARCO TEÓRICO

El formato WAV (WaveForm Audio File) es un archivo que desarrolló originalmente Microsoft para guardar audio. Los archivos tienen extensión *.wav, es ideal para guardar audios originales a partir de los cuales se puede comprimir y guardar en distintos tamaños de muestreo para publicar en la web. Es un formato de excelente calidad de audio. Sin embargo, produce archivos de un peso enorme. Una canción extraída de un CD (16 bytes, 44100 Hz y estéreo) puede ocupar entre 20 y 30 Mb.

Compresión: Los archivos WAV se pueden guardar con distintos tipos de compresión. Las más utilizadas son la compresión PCM y la compresión ADPCM. No obstante, incluso definiendo un sistema de compresión, con un audio de cierta duración se genera un archivo excesivamente pesado. El formato WAV se suele utilizar para fragmentos muy cortos (no superiores a 3-4 segundos), normalmente en calidad mono y con una compresión Microsoft ADPCM 4 bits [2].

La "Transformación rápida de Fourier", FFT para abreviar, es un importante método de medición en la tecnología de medición de audio y acústica. Descompone una señal en sus componentes espectrales individuales y así proporciona información sobre su composición. Los FFT se utilizan para el

análisis de errores, el control de calidad y la monitorización de las condiciones de las máquinas o sistemas [3].

III. PROCEDIMIENTO

Para la primera parte de cargar un archivo de audio .wav en la memoria RAM se realizaron los siguientes pasos por secciones:

1. Abrir archivo WAV en modo binario, para ello se usó la función "fopen" con la variación de que el archivo se abre en modo binario gracias a la especificación 'rb'
 - a. Input: Nombre del archivo
 - b. Output: Puntero al archivo
2. Leer cabecera del archivo WAV, para ello se usó la función "fseek" con la condición de buscar al final del archivo y así poder determinar su tamaño
 - a. Input: Puntero al archivo
 - b. Output: Información de la cabecera
3. Alocar memoria para el buffer, se hace dinámica la memoria con la función 'malloc' para poder usar el programa con cualquier archivo WAV que se desee usar sin especificar su tamaño
 - a. Input: Tamaño del archivo
 - b. Output: Puntero al buffer
4. Leer datos del archivo WAV y almacenarlos en el buffer, lee los datos con la función "fread" dados por el archivo de audio ya que estos se guardaron como matriz
 - a. Input: Puntero al archivo, puntero al buffer
 - b. Output: Datos del archivo almacenados en el buffer
5. Cerrar archivo, como abrimos el archivo con la función 'fopen' debemos cerrar el archivo a su vez con una función "fclose"
 - a. Input: Puntero al archivo
 - b. Output: Archivo cerrado

Dentro de esta primera parte presentamos las siguientes consideraciones acerca del funcionamiento del código:

¿Por qué usar un buffer para almacenar el archivo de audio .wav? El buffer es un arreglo temporal utilizado para almacenar los datos del archivo de audio antes de copiarlos a un arreglo de destino en la memoria RAM. La función de este buffer es mejorar la eficiencia y evitar problemas de memoria al leer grandes cantidades de datos de un archivo.

Al utilizar un buffer, se pueden leer pequeñas cantidades de datos del archivo a la vez, en lugar de intentar cargar todo el archivo de una sola vez en la memoria RAM. Esto reduce la cantidad de memoria que se utiliza a cualquier momento y evita problemas de memoria al cargar archivos grandes [4].

¿Por qué se debe alocar la memoria? Se aloca memoria para el buffer con la función 'malloc' para asignar un bloque de memoria dinámico del tamaño adecuado para almacenar los datos del archivo de audio. La memoria se aloca dinámicamente porque no se conoce el tamaño exacto del archivo de audio antes de abrirlo, por lo que no se puede declarar un arreglo de tamaño fijo en tiempo de compilación.

Además, alocar memoria dinámicamente permite que el programa sea más flexible y pueda manejar archivos de audio de diferentes tamaños sin tener que recompilar el programa cada vez que se desea cargar un archivo diferente [5].

Después de leer los datos del archivo en el buffer, se pueden copiar los datos al arreglo de destino en la memoria RAM y liberar la memoria alocada para el buffer con la función 'free'. De esta manera, se asegura que la memoria se utilice de manera eficiente y se evitan problemas de fuga de memoria.

No obstante, con este primer código no se podía conocer en que espacios de memoria se alojan los datos para proceder a hacer la transformada rápida de Fourier, es por ello que volvimos a revisar la estructura que tienen los archivos WAV para así crear una estructura que nos permita obtener los datos sin la necesidad de usar un buffer y reservando el código para futuros usos.

Dentro de las consideraciones nuevamente revisadas encontramos que el formato de archivos WAV funciona de forma similar a un archivo RIFF, tiene un encabezado de archivo seguido de una secuencia de paquetes de datos. La diferencia es que un archivo WAV sólo utiliza un único paquete de datos, divididos en trozos de dos datos. El primero, es un trozo de transmisor de frecuencia modulada (FMT), que especifica el formato de datos de sonido. El segundo es un fragmento de datos, que contiene la información sonora tamaño y datos de sonido crudos [6].

IV. RESULTADOS

Como resultado se pudo cargar el archivo de audio WAV con ayuda de una estructura, que nos permitió observar los datos del archivo para una posterior utilización en la realización de la transformada rápida de Fourier, la cual al representar un desafío a nivel matemático y teoría de señales termino por complicar más el primer proceso que se había llevado a cabo con éxito.

Otro de los contratiempos que se presentaron dentro de la programación fue la no utilización de librerías de terceros para la realización de nuestro código, de esta manera dentro de la planeación para llevar a cabo el laboratorio surgieron varios códigos que para nosotros resultaban bastante útiles, pero en realidad solo nos hicieron perder tiempo y esfuerzos.

V. CONCLUSIONES

Inicialmente había muchas dudas sobre el procedimiento a seguir, pero con la orientación que el profesor nos brindó en la clase pudimos aclarar un poco más el panorama. Dicha explicación nos permitió de serializar de manera correcta el archivo .wav. Por otro lado, la FFT fue un caso diferente y aunque esta fue la parte a la que más tiempo se le dedicó fue imposible hacer un código que realmente funcionara, especialmente porque se tenía realizar sin librerías externas, eso complicó bastante la situación.

VI. REFERENCIAS

- [1] E. de C. de Movavi, “¿Qué es un archivo WAV?”, Movavi.com. [En línea]. Disponible en: <https://www.movavi.com/es/learning-portal/wav-file.html>. [Consultado: 27-ene-2023].
- [2] “Cinco (5) formatos de audio y sus características - 100201_572 Grupo”, Google.com. [En línea]. Disponible en: <https://sites.google.com/site/100201572grupo/cinco-5-formatos-de-audio-y-sus-caracteristicas>. [Consultado: 27-ene-2023].
- [3] “FFT”, Nti-audio.com. [En línea]. Disponible en: <https://www.nti-audio.com/es/servicio/conocimientos/transformacion-rapida-de-fourier-fft>. [Consultado: 30-ene-2023].
- [4] C. Gestor, “CISSET Centro de Innovación”.
- [5] “Programación en C con Memoria Dinámica”, Google.com. [En línea]. Disponible en: <https://sites.google.com/site/programacioniuno/temario/unidad-1---manejo-de-memoria-dinmica/programacin-en-c-con-memoria-dinmica>. [Consultado: 31-ene-2023].
- [6] “WAV - Marketing Directo”, Marketing Directo. [En línea]. Disponible en: <https://www.marketingdirecto.com/diccionario-marketing-publicidad-comunicacion-nuevas-tecnologias/wav>. [Consultado: 31-ene-2023].