

INFORME: Laboratorio de Reproductor WAV

Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda - Sede Bogotá

Docente: Camilo Camacho

Autores: Andrea Castillo (andrea.castillo02@usa.edu.co),

Daniel Forero (daniel.forero02@usa.edu.co)

Daniel Jiménez (daniel.jimenez02@usa.edu.co)

Resumen—En este documento encontrará el desarrollo de una práctica de laboratorio, la cual tiene como principal objetivo desarrollar un reproductor de audio, en formato wav, en el que se visualice el espectro de la frecuencia y la potencia obtenida de este mediante una interfaz mostrada por consola permitiendo cargar en la memoria RAM porciones de un segundo de audio, con el fin de que se pueda retroceder y adelantar. No se llegó a la solución satisfactoria del problema debido a que las partes del código quedaban inconexas.

I. INTRODUCCIÓN

Para poder trabajar de forma adecuada en el desarrollo del código de debe tener claridad en los siguientes conceptos.

¿Qué es un archivo de tipo .WAV?

WAV es un programa (subconjunto) de formato de archivo de intercambio de recursos (RIFF) que permite almacenar archivos de audio digital. WAV es muy similar a los formatos 8SVX y AIFF. El formato de archivo WAV ha sido y sigue siendo el mejor formato estándar para CD de audio. Los archivos WAV son sin pérdidas y sin comprimir. Estos archivos se pueden comprimir con el códec Administrador de compresión de audio (ACM), ya que este formato también admite archivos comprimidos. Los archivos WAV son más grandes en comparación con otros formatos de archivos de audio más nuevos. [1]

¿Qué es la transformada rápida de fourier?

"Fast Fourier Transform." FFT para abreviar es un método de medición importante en la tecnología de medición acústica y de audio. Divide la señal en componentes espectrales separados y, por lo tanto, proporciona información sobre su composición. FFT se utiliza para el análisis de errores, control de calidad y monitoreo de condición de equipos o sistemas. Este artículo explica el cálculo de FFT, los parámetros relacionados y su efecto en los resultados de la medición. Estrictamente hablando, FFT es un algoritmo optimizado para implementar la "transformada discreta de Fourier", o DFT para abreviar. En este proceso, la parte limitada en el tiempo de la señal se divide en sus componentes.

Estos componentes son oscilaciones sinusoidales simples a frecuencias discretas con amplitud y fase definidas. Así, la FFT permite la visualización de señales en el dominio de la frecuencia [2].

II. PROCEDIMIENTO

Se identificó que las partes más importantes para poder desarrollar el código eran principalmente tres. Una donde guardamos la información del archivo .WAV, otra donde realizamos la transformada rápida de fourier y por último donde escribimos en consola las gráficas de intensidad.

Cada una de estas partes fue desarrollada de forma exitosa por aparte, pero a la hora de utilizar los datos extraídos para realizar la transformada no fue posible debido a que no se lograron obtener todos los datos necesarios; por consiguiente la impresión en consola no se puede concretar, ya que no nos puede dar un valor coherente.

Se realizó el programa utilizando únicamente librerías nativas del lenguaje c, en este caso se utilizaron stdio.h, stdlib.h, stdint.h y math.h.

III. RESULTADOS

Inicialmente se creó la estructura para el tipo de variable wav:

```
// Estructura WAV
typedef struct {
    char chunkID[4];
    int chunkSize;
    char format[4];
    char subchunk1ID[4];
    int subchunk1Size;
    short int audioFormat;
    short int numChannels;
    int sampleRate;
    int byteRate;
    short int blockAlign;
    short int bitsPerSample;
    char subchunk2ID[4];
    int subchunk2Size;
} WAV;
```

Figura 1: Estructura para el tipo de variable wav.

Se abrió y leyó el archivo de forma satisfactoria debido a que podemos extraer de él cierta información:

```
int main(int argc, char *argv[])
{
    FILE *w;
    WAV archivo;
    uint8_t *data;
    double time_to_advance;
    double *x, *y;
    uint32_t num_samples, bytes_to_advance;
    int selector;

    // Abrir archivo WAV
    w = fopen("ronaldo-siuuuu.wav", "rb");

    // Leer archivo WAV
    fread(&archivo, sizeof(WAV), 1, w);
```

Figura 2: Abrir y leer el archivo.

```
// Mostrar información del archivo WAV
printf("Chunk ID: %s\n", 4, archivo.chunkID);
printf("Chunk size: %d\n", archivo.chunkSize);
printf("Format: %s\n", 4, archivo.format);
printf("Subchunk 1 ID: %s\n", 4, archivo.subchunk1ID);
printf("Subchunk 1 size: %d\n", archivo.subchunk1Size);
printf("Audio format: %d\n", archivo.audioFormat);
printf("Number of channels: %d\n", archivo.numChannels);
printf("Sample rate: %d\n", archivo.sampleRate);
printf("Byte rate: %d\n", archivo.byteRate);
printf("Block align: %d\n", archivo.blockAlign);
printf("Bits per sample: %d\n", archivo.bitsPerSample);
printf("Subchunk 2 ID: %s\n", 4, archivo.subchunk2ID);
printf("Subchunk 2 size: %d\n", archivo.subchunk2Size);
```

Figura 3: Extracción de datos del archivo.

```
> make -s
> ./main
Chunk ID: RIFF
Chunk size: 1167430
Format: WAVE
Subchunk 1 ID: fmt
Subchunk 1 size: 16
Audio format: 1
Number of channels: 2
Sample rate: 44100
Byte rate: 176400
Block align: 4
Bits per sample: 16
Subchunk 2 ID: LIST
Subchunk 2 size: 26
```

Figura 4: datos extraídos mostrados en consola.

El paso a seguir fue realizar la transformada de Fourier:

```
// Función para calcular la DFT de un vector de muestras de audio
for (int k = 0; k < num_samples; k++) {
    y[k] = 0;
    for (int n = 0; n < num_samples; n++) {
        y[k] += w[n] * cos(2 * M_PI * k * n / num_samples);
    }
}
```

Figura 5: Función para realizar la FFT.

Por último se crearon distintas situaciones en las que se debe evaluar el valor de la frecuencia y la intensidad de la misma con el fin de poder expresar de forma gráfica los resultados obtenidos, esto se realizó por medio de un switch que evalúa los posibles escenarios e imprime los caracteres correspondientes. A continuación se muestra como se utilizó esta estructura para la frecuencia de 16K Hz:

```
printf("    Hz");
printf("    |");

for (int m = 0; m < num_samples; m++) {
    switch (w[m]) {
        case (w[m]==16000):
            switch (y[m]) {
                case (y[m] > 0 && y[m] <= 25):
                    printf("16k  |||]")~ //1
                    break;

                case (y[m] > 25 && y[m] <= 50):
                    printf("16k  |||||]")~ //2
                    break;

                case (y[m] > 50 && y[m] <= 75):
                    printf("16k  |||||||]")~ //3
                    break;
```

Figura 6: Estructura general gráficas, parte 1.

```
case (y[m] > 75 && y[m] <= 100):
    printf("16k  |||||||||]")~ //4
    break;

case (y[m] > 100 && y[m] <= 125):
    printf("16k  |||||||||||]")~ //5
    break;

case (y[m] >= 125 && y[m] <= 150):
    printf("16k  |||||||||||||]")~ //6
    break;

case (y[m] > 150 && y[m] <= 175):
    printf("16k  |||||||||||||]")~ //7
    break;

case (y[m] > 175 && y[m] <= 200):
    printf("16k  |||||||||||||]")~
    break;
```

Figura 7: Estructura general gráficas, parte 2

```

    case (y[m] > 200 && y[m] <= 225):
        printf("16k |") //9
        break;

    case (y[m] > 225 && y[m] <= 250):
        printf("16k |")
        break;

    case (y[m] > 250 && y[m] <= 255):
        printf("16k |")
        break;

    default:
        printf("16k |")
        break;
}
break;

```

Figura 8: Estructura general gráficas, parte 3

IV. CONCLUSIONES

A lo largo del desarrollo de la actividad fue notorio el hecho de que se tiene claridad en los conocimientos generales como lo son cargar el archivo, qué es y cómo traducir a código la transformada rápida de fourier y el uso de caracteres para elaborar representaciones gráficas, pero se vió que la complejidad del ejercicio radica en el hecho de que involucra la combinación de distintos conocimientos y si esta unión no se realiza de forma completa se imposibilita acabar la actividad.

En nuestro caso el problema se presentó a la hora de obtener las frecuencias, debido a que no supimos cómo leer dicha información y sin este dato no se puede realizar la transformada y sin realizar dicha transformada no es posible imprimir el nivel de potencia de cada una de las frecuencias.

REFERENCIAS

- [1] A. Cervera. ¿que es un archivo wav? ¿sabes realmente lo que es? (recuperado el 30 de enero de 2023). [Online]. Available: <https://recoverit.wondershare.es/audio-recovery/what-is-wav-file.html>
- [2] Nti-audio. Transformada rápida de fourier fft conceptos básicos. (recuperado el 30 de enero de 2023). [Online]. Available: <https://www.nti-audio.com/es/servicio/conocimientos/transformacion-rapida-de-fourier-fft>