ELO 330 - Tarea 2: Procesando audio usando código libre

Gabriela Altamirano Rol: 2803052-5 Sebastián Marquez Rol: 2921038-1

I. Descripción de Módulos

El programa que se creó para solucionar el problema se encuentra en el archivo `csa.c` y cuenta con las siguientes funciones

- `main`
- `lectura_audio`
- `escritura_audio`
- `saturar`
- `get_fix_data`

y utiliza los scripts octave_notebook.m y sound_plots.m

main

Secuencialmente, el main lo primero que hace es obtener los argumentos de entrada (**char* fn_audio, float gain, int offset**), de manera de poder leer el archivo de audio que se le entrega (el cual se almacena en **shor int * raw_data**), y luego, poder saturarlo con la ganancia indicada (cuyo resultado se almacena en **shor int * sat_data**). A continuación se crean dos pipas y se realiza un fork, de manera de tener dos procesos.
En el proceso hijo (pid=0), se configura el uso de las pipas desde el punto de vista del proceso hijo, en donde la lectura del proceso es realizada por la pipa **int pipe1** y la escritura del proceso por **int pipe2**. Esto significa que la escritura de pipe1, y la lectura de pipe2, deben ser cerradas. Luego se utiliza el comando execlp para que el proceso mute a octave.
En el proceso padre, se cierran los descriptores de archivos correspondientes y se crean dos archivos asociados a los descriptores de archivos pipe1 y pipe2, **FILE * sd** para escritura por parte del proceso padre, y **FILE * sd2** para la lectura. Luego se procede a enviar el valor de las variables que octave necesita para poder graficar las señales y obtener una señal suavizada a partir de la saturada. Una vez enviado los datos se envia un comando que ejecuta los scripts `octave_notebook.m` y `sound_plots.m`.
Para recibir los datos de la señal suavizada **short int * fix_data**, se utiliza la función `get_fix_data`.
Finalmente, se calcula el error entre la señal original y la recupera (**float error**) se invoca a una función para la escritura de los archivos generados, y se reproducen los tres archivos generados si corresponde.

lectura_audio

La función recibe el nombre del archivo a abrir por medio de **char * fn_audio** y recibe un puntero para almacenar el número de muestras del archivo (**int * size_data**. La función abre el archivo y lee cada dos bytes en un ciclo while para obtener el tamaño de muestras. La función retorna **short int * raw_data**.

saturar

La función recibe los datos leídos en **raw_data**, el número de muestras **size_data** y la ganancia ingresada como argumento **gain**. La función comprueba si la multiplicación de la ganancia con un dato de raw_data supera el umbral que permite representar 16 bits (32767 y -32767). De ser así, en sat_data guarda el valor máximo dividido por la ganancia, en caso contrario copia el valor original. De esta manera se realiza la saturación y el reescalamiento en un solo paso. La función retorna **short int * sat_data**.

get_fix_data

Esta función se utiliza para recuperar datos desde octave. Recibe los argumentos **short int ** fix_file,int* size_data**, **FILE** pd**, **FILE** pd2**. Uno de los problemas de leer desde octave, fue que cada vez que se ingresaba un comando, pero no se leía de vuelta, quedaba el número de la línea de comando de octave en el buffer, pero no se podía leer si lo enviado no retornaba un valor. La forma que encontramos de solucionar esto fue utilizar la función sscanf, con previo conocimiento de lo que se iba a recibir en el buffer (el string correspondiente a la línea de comandos, y luego los datos que se habían pedido con la función disp). Para no tener que asignar un tamaño demasiado grande al buffer, partes de la señal se van leyendo en un ciclo for. Esta función no retorna nada, ya que los datos fueron pasados por referencia.

escritura_audio

Esta función recibe los datos a escribir **short int * audio_data**, el nombre del archivo **char* name** y el tamaño del arreglo **int size_data** y guarda los archivos en el disco duro con la función fwrite.

octave_notebook.m

En este script se encuentran todas las instrucciones para realizar el suavizamiento de la señal por medio de octave. Recibe la señal saturada **sat** y se crea la variable a retornar **x**, inicializada con los valores de sat. El script busca en el arrego el valor máximo y el valor mínimo, y luego se crean dos arreglos (**index_max** y **index_min**) que contienen el índice de la señal con los valores saturados. En un ciclo for se revisan estos arreglos para buscar índices contiguos, cuando termina de encontrar una región saturada (delimitada por **first_index** y **last_index**), realiza un fit polinomial, luego lo evalua y se cambian los valores de x correspondientes. Luego x es transformado a un entero de 16 bits.

sound_plots.m

Recibe el **offset** de tiempo en milisegundos y hace los gráficos de las 3 señales en base a este valor. En caso de que el offset pedido sea muy elevado, los gráficos muestran los últimos 40 ms de la señal, independiente del offset.

II. Diagrama de Alto Nivel