

# Lectura de Archivos de Audio con JACK y libsndfile

# libsndfile

- Instalar por medio de:
  - `sudo apt-get install libsndfile1-dev`
- Se incluye en el compilador por medio de:
  - `gcc -o prog code.c -ljack -lsndfile`
- Y el header es:
  - `#include <sndfile.h>`
- Referencia Oficial:
  - <http://www.mega-nerd.com/libsndfile/api.html>

# Ejemplo de Agente de JACK

## Lectura Simple

- Descarguen:
  - `jack_read_file.c`
  - `makefile`
- Mi regalo para ustedes...

# Apertura de Archivo

- `sf_open`
  - Dónde está el archivo de audio
  - Si vamos a leerlo (`SFM_READ`) o escribirlo (`SFM_WRITE`).
  - Apuntador a una estructura tipo `SF_INFO` que:
    - Nos dice información del archivo a leer.
    - O, describimos el archivo a escribir.
- Regresa apuntador a archivo tipo `SNDFILE`.

# SF\_INFO

typedef struct

```
{  sf_count_t frames ; //datos
   int      samplerate ; //frecuencia de muestreo
   int      channels ; //cuantos canales
   int      format ; //formato de archivo
   int      sections ; //secciones
   int      seekable ; //si es "buscable"
} SF_INFO ;
```

# sf\_read\_float

- Hay varias funciones de lectura, cada una regresa un tipo de valor diferente. El que es compatible con JACK es el de “float”.
- Requiere:
  - Un apuntador SNDFILE (creado por sf\_open)
  - Un arreglo de tipo float.
  - Cantidad de muestras (*frames*) a guardar en el arreglo.
- Regresa cuantos muestras guardó:
  - Usualmente es igual a la cantidad que se le pide, a menos de que haya llegado al final del archivo.

# sf\_close

- Cierra el archivo.

# Para probar el agente...

- Tenemos tres archivos de audio de prueba descargables de la página del curso:
  - test16000.wav
  - test44100.wav
  - test48000.wav
- El número representa la frecuencia de muestreo del archivo.
  - Escojan el más cercano al de su computadora.



¿Qué sucede si utilizan un archivo con una frecuencia de muestreo diferente al de su máquina?

¿Por qué?

# Frecuencia de Muestreo en Archivos

- La frecuencia de muestreo de un archivo indica la frecuencia con la que se grabó.
- Por lo tanto, es la frecuencia con la cual se tiene que leer para que se escuche adecuadamente.

# Caso #1:

- JACK: 48000 kHz.
- Archivo: 16000 kHz.

$$\text{JACK/Archivo} = 48000/16000 = 3.$$

- JACK está leyendo 3 veces más rápido que como el Archivo fue grabado.
  - Voz de ardilla.

## Caso #2:

- JACK: 44100 kHz.
- Archivo: 48000 kHz.

$$\text{JACK/Archivo} = 44100/48000 = 0.91.$$

- JACK está leyendo 0.91 veces más lento que como el Archivo fue grabado, casi 1.
  - Casi la misma voz.

Digamos que tenemos un Archivo grabado a 16000 kHz y JACK está corriendo a 44100 kHz.

¿Cómo le hacemos para que se oiga bien?

# Solucion #1

- Cambiar la frecuencia de muestreo de JACK.
- PRO:
  - No necesitamos cambiar nada de nuestro código.
- CON:
  - TODOS los agentes de JACK cambiarían su frecuencia de muestreo nada más por los problemas de un agente.
  - No es posible hacerlo mientras se corre el agente, se tiene que hacer manualmente, matando el servidor JACK, hacer el cambio a mano, y reiniciarlo.
  - Es posible que la frecuencia de muestreo no sea admitido por JACK, como es el caso de 16000 kHz.

# Solucion #2

- Cambiar la frecuencia de muestreo del archivo.
- PRO:
  - No necesitamos cambiar nada de nuestro código.
- CON:
  - Tenemos que hacerlo, manualmente, para cada archivo que leamos.

# Un buen “compa”: SOund eXchange

- Conocido como “sox”.

```
sudo apt-get install sox
```

- La “Navaja Suiza” de conversión de archivos de audio.
- Para que de información de un archivo de audio:  

```
sox --info entrada.wav
```



# Un buen “compa”: SOund eXchange

- Para cambiar la frecuencia de muestreo de una archivo a 16 kHz:

```
sox entrada.wav -r 16000 salida.wav
```

- También tiene otros trucos:
  - [http://stefaanlippens.net/audio\\_conversion\\_cheat\\_sheet](http://stefaanlippens.net/audio_conversion_cheat_sheet)

# Solucion #3

- Cambiar la frecuencia de muestreo del archivo, *en línea* (mientras que corre el agente).
- PRO:
  - Todo es automático.
- CON:
  - Se requiere un cambio substancial a nuestro código.

# libsamplerate

- Se instala por medio de:  
`sudo apt-get install libsamplerate0-dev`
- Se incluye en el compilador por medio de:  
`gcc -o prog code.c -ljack -lsamplerate`
- Y el header es:  
`#include <samplerate.h>`
- Referencia Oficial:
  - [http://www.mega-nerd.com/SRC/api\\_full.html](http://www.mega-nerd.com/SRC/api_full.html)

Nuevo ejercicio...

... no se crean ...

# El cambio que tendríamos que hacer...

- Es tan substancial, que nos aleja bastante del tema central del curso.
- Pero, no quiero dejar el tema así nada más, por lo que les voy a dar otro regalo:

```
jack_read_file_samplerate.c
```

```
makefile
```

# Aún otro regalo...

- También les regalo la contraparte a `jack_read_file`: un agente de jack que escribe un archivo de audio a partir de una de sus entradas:
  - `jack_write_file.c`
  - `makefile`
- Toma como argumento el nombre del archivo a guardar y lo escribe con la frecuencia de muestreo que tiene JACK configurado.

Siguiente clase...

Transformada de Fourier y  
la librería FFTw3