

Organización del Computador II

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Trabajo Práctico Final: PedalerIA64

Preinforme

Integrante	LU	Correo electrónico
Laporte Matías	686/09	matiaslaporte@gmail.com

1. Motivación

La motivación del presente Trabajo Práctico Final (de ahora en más, **TP**) surge de poder aplicar los contenidos teóricos de la materia en un entorno que no se haya incursionado en la misma, como es el procesamiento de audio. Se eligió tal contexto por motivos personales del programador, el poder explorar y vincular dos intereses propios (música y programación).

2. Objetivos

La idea del TP es implementar una serie de algoritmos que produzcan efectos sobre audio, en los lenguajes *C* y *ASM* (mediante el uso de instrucciones *SSE*), para proceder a la comparación del rendimiento de ambos (mediante la metodología utilizada -contar ticks del procesador- en el TP2 de la materia, 1er. cuatrimestre de 2011). El TP consistirá en lo siguiente:

- Base teórica: conceptos de procesamiento de señales y de audio en particular, herramientas utilizadas (librerías, lenguajes).
- Utilización del programa: requerimientos a instalar para el programa, ejemplos de cómo correrlo
- Desarrollo: explicación de cada uno de los efectos implementados con pseudocódigo y prosa, cuál es el efecto que generan.
- Conclusiones: resultados obtenidos, comparación con lo esperado, dificultades encontradas a la hora de desarrollar el TP, etc.

Todavía no se encuentran definidas todos los efectos que se implementarán. Por lo pronto, ya se realizaron (visibles en el código adjunto) una función de copiado del archivo (sin efecto), y otra de delay simple sin feedback, ambas muy básicas. Se esperan desarrollar efectos que involucren FTT ¹ (Fast Fourier Transform) y Convolución ² (que utiliza la FTT); ambos, procesos fundacionales en procesamiento de audio.

Una prueba que me gustaría hacer para la entrega final es ver si lo desarrollado se puede aplicar en audio en tiempo real, algo que todavía no fue experimentado (principalmente porque el manejo de audio en tiempo real varía mucho según la instalación de Linux, la placa de audio, el soft que use el OS para Audio (ALSA, OSS, etc.), etc.).

3. Uso del código adjunto

Como se dijo previamente, como ejemplo se adjuntó el código base para empezar a desarrollar el TP entero, con dos funciones, una de copiado, y otra de delay simple sin feedback, ambas implementadas tanto en *C* como *ASM*. Para probarlo, se necesita instalar el paquete **libsndfile1**, que se encuentra en los repositorios de Ubuntu. Dicha librería se utiliza para el manejo (lectura/escritura) de los archivos de audio.

¹<http://www.dspguide.com/ch12/2.htm>

²[http://en.wikipedia.org/wiki/Convolution\\$_\\$_reverb](http://en.wikipedia.org/wiki/Convolution$_$_reverb)

Una vez instalado el paquete, compilar el programa con el makefile (comando: *make*). Ejemplos de uso del programa:

Ver opciones: *./main*

Copiado de archivo en C: *./main archivo_entrada.wav archivo_salida.wav -c*

Copiado de archivo en ASM: *./main archivo_entrada.wav archivo_salida.wav -C*

Delay de 1.5 segundos con 0.6 de decay en C: *./main archivo_entrada.wav archivo_salida.wav -d 1.5 0.6*

Delay de 1.5 segundos con 0.6 de decay en ASM: *./main archivo_entrada.wav archivo_salida.wav -D 1.5 0.6*

Se incluyen archivos de Audio de ejemplo en la carpeta *inputExamples*.

4. Bibliografía tentativa

4.1. Libros

Referencias

- [1] F. Richard Moore, *Elements of Computer Music*, 1990, Prentice Hall, New Jersey
- [2] Richard Boulanger, Victor Lazzarini *The Audio Programming Book*, 2011, The MIT Press, Massachusetts
- [3] Steven W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, Second Edition, 1999, California Technical Publishing, California

4.2. Internet

- <http://www.mega-nerd.com/libsndfile/api.html>
- <http://stackoverflow.com>
- <https://ccrma.stanford.edu/~jos/>
- <http://www.musicdsp.org/>
- <http://www.kvraudio.org/>