

ELO 330: Programación de Sistemas

TAREA 2

24 de Noviembre de 2015

Richard Atuan Toledo 201121028-K

Felipe Fernández Pino 201121011-5

Descripción del programa

El programa *rpas* (**R**eparador de **P**érdidas en **A**udio **S**treaming.) simula la pérdida de paquetes de audio en una transferencia de audio streaming y luego realiza una recuperación de audio en los tramos del audio en los que se haya perdido información. Rpas también almacena los dos archivos de audio generados en formato *.raw*, muestra en un gráfico un tramo de 400 [ms] de cada uno de los tres audios (original, con pérdidas y recuperado) y (en caso de ser requerido) reproduce los archivos de audio.

La recuperación consiste en insertar en los tramos de pérdida la información contenida en el tramo de audio anterior. Para los casos en donde existen dos tramos de pérdidas el segundo tramo se deja sin información (se rellena con ceros).

Para reducir la distorsión producida al insertar tramos de audio no continuos se realiza una interpolación de datos en los extremos iniciales y finales de cada tramo insertado, suavizando así la unión de los datos.

Para llevar a cabo los cálculos matemáticos, la muestra gráfica de las señales y la reproducción de audio se hace uso de los softwares *octave*, *gnuplot* y *aplay* respectivamente.

Estrategia de solución

I. Ejecución de otros procesos:

Para poder ejecutar otro proceso a partir del programa principal se utilizan las funciones `fork()` y `exec()`. Primero se crea un proceso hijo a partir del proceso principal (padre) con la función `fork()`, luego, en el proceso hijo, se realizan los ajustes necesarios para la comunicación entre el proceso padre y el proceso hijo. Finalmente se ejecuta el programa *octave* con el uso de la función `exec()`.

II. Comunicación entre procesos:

Para hacer uso del software *octave* es necesario enviarle comandos matemáticos por su entrada estándar. Para lograr esto se utiliza un *pipe* encargado de traspasar información desde el proceso padre (rpas) al proceso hijo (octave). También es necesario poder leer los resultados que *octave* entrega de los cálculos matemáticos que realiza; para esto es necesario utilizar otro *pipe* que se encarga de traspasar información desde el proceso hijo (octave) al proceso padre (rpas). En la Figura 1 se puede observar esta conexión.

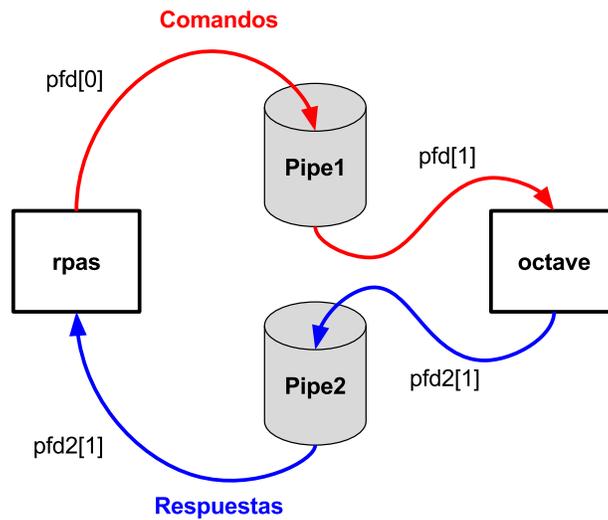


Figura 1: Comunicación entre procesos.

Resultados

Para probar los resultados obtenidos se utiliza el software *Audacity*, capaz de mostrar gráficamente las señales y de reproducirlas.

En la Figura 2 se puede observar un tramo de señal en los tres archivos de audio utilizados (original, con pérdidas y recuperado). Se puede apreciar que entre los tiempos $t = 0,16$ y $t = 0,18$ se pierde información en el archivo “conCeros”. También se puede apreciar que en el archivo “recuperado” el mismo tramo de tiempo contiene información. Esta información corresponde a la copia del tramo anterior que luego es adaptado mediante interpolación para reducir distorsión en la señal.

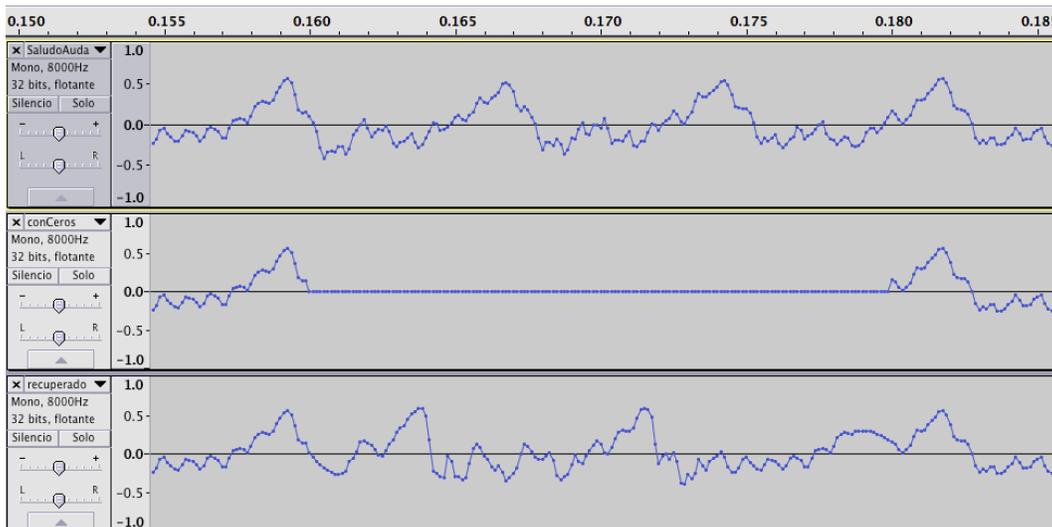


Figura 2: Recuperación de uno de los tramos.