



Universidade Federal de São Carlos
 027430 A - Tecnologia e Comunicação de
 Dados (2010/01)
 Prof. Dr. Cesar Marcondes
 Mini-Projeto 1

Data de Entrega

9 abril 2010

Mini-Projeto 1: Desenvolver um **programa em C** para :

1) Analisar um arquivo WAV (de preferência, com formato WAV da Microsoft),

fulano@ssh:~\$./meuprograma arquivo.wav

The Canonical WAVE file format

endian	File offset (bytes)	field name	Field Size (bytes)	
big	0	ChunkID	4	The "RIFF" chunk descriptor
little	4	ChunkSize	4	
big	8	Format	4	
big	12	Subchunk1ID	4	The "fmt" sub-chunk describes the format of the sound information in the data sub-chunk
little	16	Subchunk1Size	4	
little	20	AudioFormat	2	
little	22	NumChannels	2	
little	24	SampleRate	4	
little	28	ByteRate	4	
little	32	BlockAlign	2	
little	34	BitsPerSample	2	
big	38	Subchunk2ID	4	
little	40	Subchunk2Size	4	
little	44	data	Subchunk2Size	

The "RIFF" chunk descriptor

The Format of concern here is "WAVE", which requires two sub-chunks: "fmt" and "data"

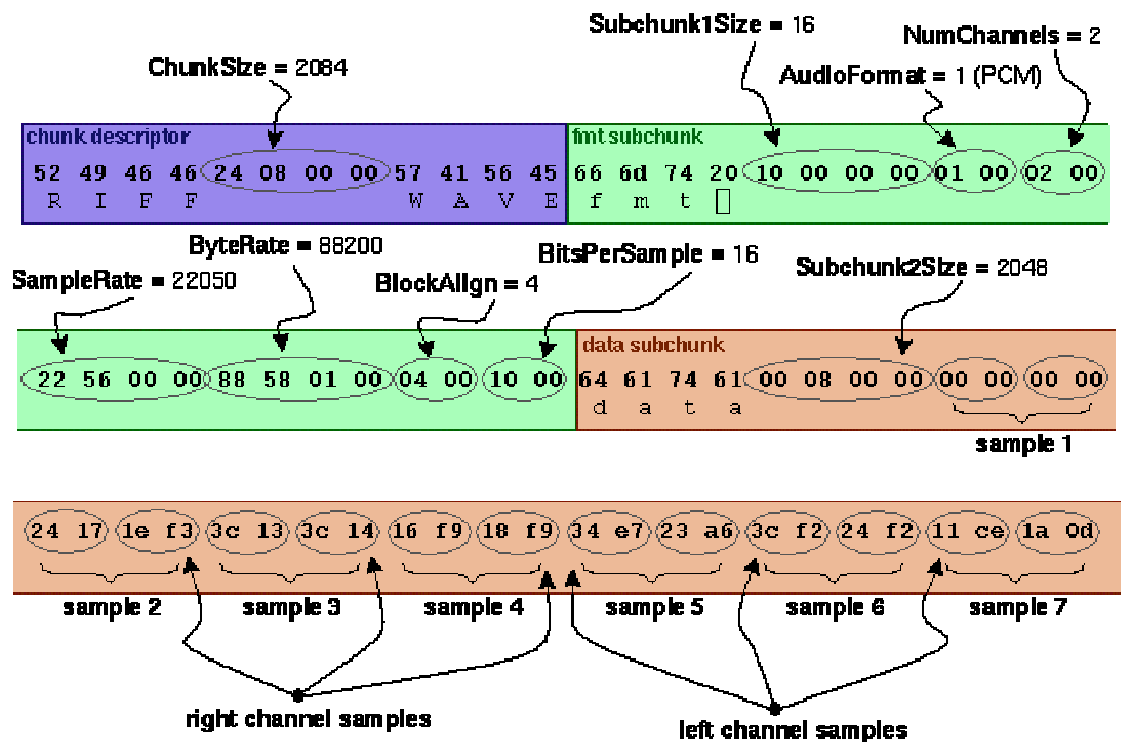
The "fmt" sub-chunk

describes the format of the sound information in the data sub-chunk

The "data" sub-chunk

Indicates the size of the sound information and contains the raw sound data

2) Extrair o header contendo informações sobre o arquivo,



Fonte: <http://ccrma.stanford.edu/courses/422/projects/WaveFormat/>

Em outras palavras, o seu programa deve gerar a seguinte saída (e opcionalmente um arquivo de output contendo as amostras):

```
fulano@ssh:~$ ./meuprograma -o amostras.txt arquivo.wav
ChunkSize = 2084
SubChunkSize = 16
AudioFormat = PCM
NumChannels = 2
SampleRate = 22050
ByteRate = 88200
BlockAlign = 4
BitsPerSample = 16
SubChunk2Size = 2048
```

Enquanto que as amostras são jogadas para um arquivo texto. Como esse som é estéreo (canais right e left), podemos ter como saída 2 arquivos amostras1.txt e amostras2.txt.

Conteúdo de Amostras1.txt	Conteúdo de Amostras2.txt
0000h	0000h
2417h = (9239 em decimal)	1ef3h (7923 em decimal)
3c13h	3c14h
16f9h	18f9h
34e7h	23a6h
3cf2h	24f2h
11ceh	1a0dh

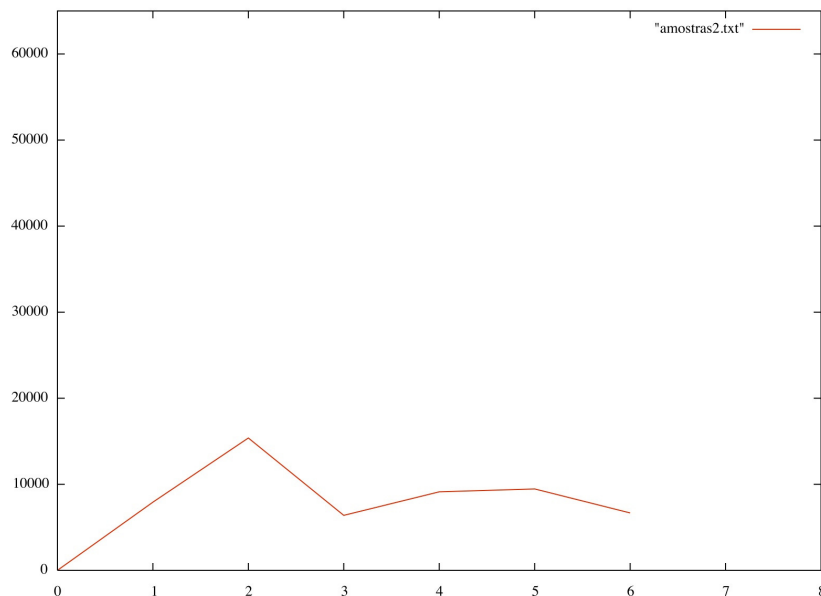
Obs.: Seria importante converter os números de hexadecimal para decimal quando escrever nos arquivos contendo as amostras.

3) Extrair os níveis de quantização do mesmo

É parecido com o passo 2, onde são extraídas as amostras. Cabe ao grupo pudesse plotar as amostras de 1 ou 2 segundo no eixo do tempo, como exemplo, usando os valores acima. Convertendo os valores de amostra2.txt para decimal (e como o formato wav disse que está codificando os valores em 16 bits, o máximo nível de quantização é por volta de ~65000 . Então usando o gnuplot (<http://www.gnuplot.info/>)

gnuplot> plot [0:8][0:65000] "amostras2.txt" with lines

Temos o seguinte resultado para essas 7 amostras (7 em 22050 por segundo, de acordo com o formato). Cada intervalo de tempo (eixo x equivale a 45.35 microsegundos, logo esses 7 pontos são em 317usec.

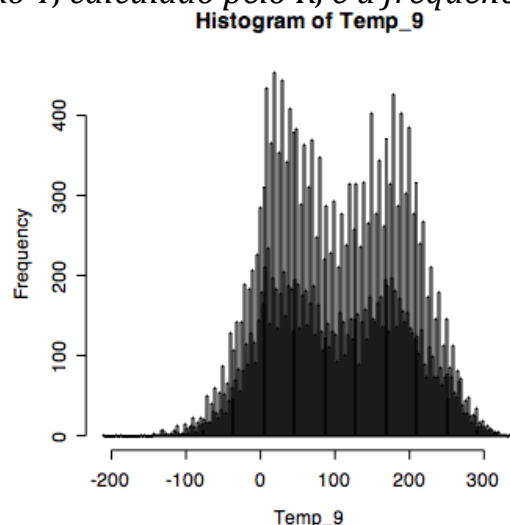


4) Estudar 5 arquivos de Voz e 10 Musica em formato WAV e fazer a estatística dos níveis de quantização utilizados através de histogramas.

Separar os arquivos, e separar 30 segundos por arquivo. Aqui (<http://www.nch.com.au/acm/>) tem uns exemplos de voz, ou talvez seja necessário converter para WAV. Ou usar um programinha de gravação de som para gravar suas vozes. Musica é só pegar da Internet.

5) Os histogramas devem ser feitos usando o dado da amplitude de cada intervalo das 44100 amostras, depois variando para 2 intervalos das 44100 amostras (usando a media entre pares de amplitudes) e 5 intervalos das 44100 amostras (usando a media entre as cinco ultimas amplitudes). A idéia de usar intervalos maiores é capturar sinais regulares que se repetem em intervalos maiores que 1.

*Para criar uma base "estatística" boa, seria essencial que o grupo pudesse usar vários segundos (ex. uma "fala" de 30 segundos) por ARQUIVO ESTUDADO. Assim, cada segundo tem 44100 amostras * 30 = 1.323.000 amostras. **Enviar isso para um arquivo de texto usando "printfs" a partir do seu programa em C** (de 3 ou 4MB, considerando 8 bits por caractere). Depois aciona a ferramenta estatística "R" (<http://www.r-project.org/>), ver o comando "scan" para ler os dados (<http://www.math.csi.cuny.edu/Statistics/R/simpleR/stat022.html>). e plotar o histograma (<http://rankexploits.com/musings/2007/make-a-file-easy-to-read-data-into-r-and-create-a-histogram/>) sendo que o histograma, o grupo tem que ajustar na mão o melhor tamanho dos boxes. Exemplo de histograma seria (onde o eixo X são os valores das amostras de amplitude, e o eixo Y, calculado pelo R, é a freqüência daquele valor):*



Para o restante do exercício, 2 intervalos e 4 intervalos, no seu programa em C ... quando o seu programa estiver lendo as 44100 amostras, o seu programa pode receber como parâmetro que ele deve agrupar as amostras em pares (2 intervalos) e fazer uma média a cada 2 amostras. Logo, sua saída será de 22050 amostras * 30 seg = 661.500 amostras. Envia os dados para o R, e extrai a histograma.

Ex.: Usando a tabela de "amostras2.txt"

Resultado Original	Agrupando os Dados em Pares
0	3961,5
7923	10886,5
15380	9292
6393	
9126	
9458	

Se o agrupamento em par tiver alguém sobrando, complete com 0 e faça a média.

E para 4 intervalos, agrupar 4 amostras, faz a média das 4 amostras, resultando em $11,025 * 30 = 330,750$ e repete a leitura dos dados no R para gerar o histograma.

6) No final, uma pagina para discutir os resultados estatísticos. Eles são semelhantes? Existe uma tendência em alguma amplitude específica?

Essa é uma análise crítica de cada grupo sobre os resultados do trabalho.