



PROJETO

Compressão de Áudio Digital

Divulgação: 14/09/2018.

Data de entrega: 30/10/2018.

Grupos de no máximo 2 alunos.

Objetivos

O objetivo principal do projeto é colocar em prática alguns conceitos teóricos abordados em sala de aula relacionados a técnicas de compressão. Para isso, os grupos deverão implementar algumas técnicas de compressão/descompressão – a saber – aplicadas a áudio digital. Serão empregadas as técnicas de **codificação por diferenças**, **codificação por carreira (*run-length*)** e **codificação por Huffman**. As três técnicas deverão ser aplicadas, de modo combinado ou isolado, a um arquivo de áudio em formato WAV sem compressão. Os dados resultantes da compressão devem ser armazenados em um arquivo binário.

Entrada esperada

Para o processo de compressão: um arquivo de áudio digital em formato WAV sem compressão. Para o processo de descompressão: um arquivo binário contendo a representação compactada do arquivo de áudio original.

Tanto codificador quanto decodificador devem ser executados por linha de comando (sem interface gráfica), obedecendo o seguinte formato:

Codificação

```
encode -d -c -h <entrada.wav> <saida.bin>
```

onde:

encode	nome do codificador
-d	utilização da codificação por diferenças
-c	utilização da codificação por carreira
-h	utilização da codificação Huffman
<entrada.wav>	arquivo original sem compressão
<saida.bin>	arquivo binário comprimido

Exemplos:

```
encode -d -c original.wav codificado.bin
```

Significado: codifica o arquivo “original.wav” usando a codificação por diferenças e a codificação por carreira, criando o arquivo codificado chamado de “codificado.bin”.

```
encode -h -c original.wav codificado.bin
```

Significado: codifica o arquivo “original.wav” usando a codificação por carreira e a codificação por Huffman, criando o arquivo codificado chamado de “codificado.bin”.

Decodificação

```
decode <entrada.bin> <saida.wav>
```

onde:

decode nome do decodificador
<entrada.bin> arquivo binário comprimido
<saida.wav> arquivo WAV sem compressão

Exemplo:

```
decode codificado.bin descomprimido.wav
```

Significado: decodifica o arquivo “codificado.bin” criando o arquivo “descomprimido.wav”.

Saída esperada

Para o processo de compressão espera-se como saída um arquivo binário contendo um fluxo de bits que represente o arquivo de áudio compactado. Para o processo de descompressão espera-se como saída uma aproximação do áudio original em formato WAV não comprimido.

Critérios de avaliação

O projeto será avaliado segundo os critérios abaixo, por ordem de importância:

1. Corretitude (Faz tudo o que deveria?).
2. Qualidade dos resultados (Taxa de compressão, qualidade do áudio decodificado).
3. Usabilidade (Confiabilidade, facilidade de uso, obediência aos requisitos do projeto e parâmetros de entrada e saída de linha de comando).
4. Qualidade do código: organização e documentação interna.

Considerações sobre o trabalho 1:

- A implementação deverá ser feita em C, Java ou Python. O programa deverá operar por linha de comando (sem interface gráfica).
- O uso de bibliotecas externas deverá ser pré-aprovado pelo professor. Mande um e-mail para mmanzato@icmc.usp.br com o link, descrição e motivo de uso de tal biblioteca no trabalho.
- Espera-se poder aplicar as técnicas de compressão de modo isolado (uma técnica por vez) ou de modo conjunto, gerando saídas diferentes para cada caso. No modo conjunto, é livre a escolha da ordem de aplicação das técnicas.
- A implementação das técnicas deverá primar pela busca da maior taxa de compressão com melhor qualidade de áudio possível.

- O decodificador deverá ser capaz de detectar qual codificador foi usado (diferença, carreira ou Huffman) a partir de uma análise do arquivo binário codificado (pode-se usar um cabeçalho para isso).
- Deverão ser entregues: os arquivos contendo o código-fonte; um arquivo TXT (readme) descrevendo o grupo, com nome completo e no. USP, e instruções de compilação e execução do programa. Tais instruções serão seguidas à risca na correção, sendo que qualquer falha na execução dos passos acarretará nota zero.
- A entrega deve ser realizada via o ambiente Tidia: um dos componentes do grupo deve fazer o upload do trabalho em seu “escaninho”. Para isso, crie uma pasta chamada “projeto_1” e insira os arquivos lá.
- O compilador GCC em ambiente Unix será utilizado para correção dos trabalhos. Programas desenvolvidos em outros ambientes e compiladores, caso não possam ser compilados com GCC/Unix, terão a nota zerada.
- Qualquer tipo de cola (de colegas ou da Internet) acarretará nota zero.
- Trabalhos com entrega a partir de 30/10/2018 terão uma penalidade de menos 1 ponto por dia de atraso.

Referências:

- [1] Formato de um arquivo WAVE: <http://soundfile.sapp.org/doc/WaveFormat/>
- [2] Site contendo vários arquivos WAVE para testar: <http://freewavesamples.com/yamaha-tg100-whistle-c5>
- [3] Repositório da disciplina no Tidia: programa em C que lê um arquivo WAVE.