

TP N°01

COMPRESSION EN MP3

Dans ce TP nous allons voir comment compresser un fichier audio wav en format MP3 en utilisant la célèbre bibliothèque de LAME.

Le TP sera réalisé en C++ sous l'environnement de développement Visual Studio.

Pour cela, nous allons suivre les étapes suivantes :

1. Créer un nouveau projet vide de type « Empty project » appelé « TP01 » ;
2. Créer la classe « WaveAudio » contenant les informations du fichier wav. Elle contient comme attributs : le nom du fichier wav, un buffer de type (short int) appelé `wav_buffer` qui doit contenir les données audio, les paramètres du fichier audio : `NumChannels` (MONO ou STEREO), `SampleRate` (16000hz, 44100hz, ...), `BitsPerSample` (8bits, 16bits, ...), `ByteRate` (16kb/s, 32kb/s, 64kb/s, ...).
3. Ecrire le constructeur de la classe `WaveAudio` qui reçoit en paramètre le nom du fichier wav, fait l'allocation du buffer de taille 8192 (2^{13}), ouvre le fichier en lecture binaire « rb », remplit les paramètres audio de l'entête wav.

Le format de l'entête wave se trouve sur ce lien :
<http://soundfile.sapp.org/doc/WaveFormat/>

4. Ecrire le destructeur de la classe `WaveAudio` qui libère l'espace mémoire du buffer, et ferme le fichier.
5. Ecrire une méthode de la classe `WaveAudio` appelée `getNextBuffer()` qui permet de lire le prochain buffer de taille 8192 à partir du fichier et le placer dans le buffer `wav_buffer`. Retourner à la fin le résultat de la méthode `fread()`.
6. Créer un nouveau fichier appelé `CompressionMP3.cpp` contenant la méthode `main()`.
7. Dans la méthode `main` :
 - Créer un objet `myAudio` de la classe `WaveAudio`
 - Initialiser lame en utilisant la méthode `lame_t lame = lame_init()`.
 - Initialiser la fréquence d'échantillonnage de lame à partir de `SampleRate` de l'objet `myAudio` en utilisant la méthode `lame_set_in_samplerate(lame, value)` de lame.
 - Initialiser le mode de la séquence audio de lame par (MONO ou STEREO) à partir de `NumChannels` de l'objet `myAudio` en utilisant la méthode `lame_set_mode(lame, mode)` de lame.
 - Initialiser le byte rate de lame à partir de $(ByteRate/1000)$ (1000 pour travailler en Kb/s) de l'objet `myAudio` en utilisant la méthode `lame_set_brate(lame, value)` de lame.

- Initialiser les paramètres de lame en utilisant la méthode `lame_init_params()` de lame.
- Créer un buffer de type `unsigned char` pour contenir les données compressées appelé `mp3_buffer` de taille 8192.
- Dans une boucle, lire le prochain buffer en utilisant `getNextBuffer()`.
- Si le résultat de `getNextBuffer()` est 0 appeler la méthode `lame_encode_flush(lame, out_buffer, size)`.
- Sinon appeler `lame_encode_buffer_interleaved(lame, wav_buffer, in_read, out_buffer, size)`.
- Ecrire le buffer dans un fichier.
- Après la boucle, fermer lame `lame_close(lame)` et le fichier.