

Laboratoire de High Performance Coding semestre printemps 2025

High Performance Coding (HPC)

Temps à disposition : 4 périodes (deux séances de laboratoire)

Récupération du laboratoire dans l'archive sur Cyberlearn.

1 Objectifs de ce laboratoire

L'objectif de ce laboratoire est de vous familiariser avec le langage C en implémentant un système d'encodage et de décodage DTMF (Dual-Tone Multi-Frequency).

2 Qu'est-ce que le DTMF ?

Le DTMF est un système de signalisation utilisé dans les télécommunications pour coder les chiffres et certaines lettres en utilisant une combinaison de deux fréquences. Ce système est notamment utilisé dans les claviers de téléphones pour composer des numéros.

Chaque touche est associée à une paire unique de fréquences : une fréquence basse et une fréquence haute. Lorsqu'une touche est pressée, ces deux fréquences sont émises simultanément.

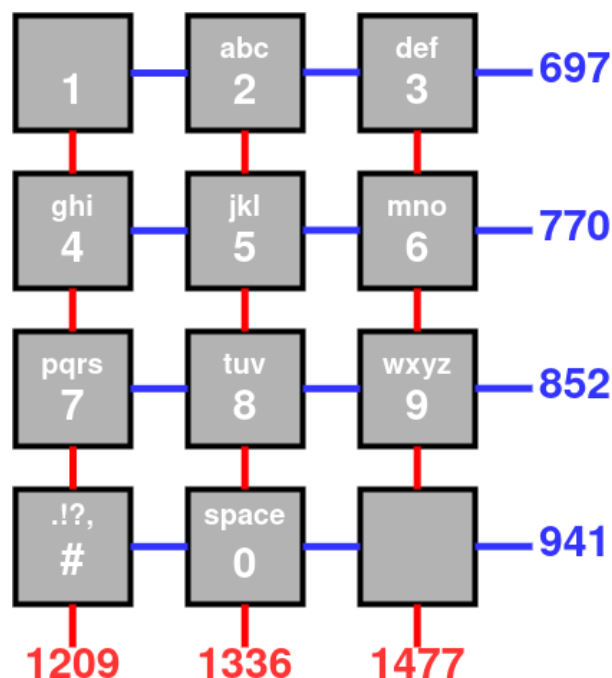


Figure 1: Table des fréquences DTMF

3 Encodage DTMF

L'encodage consiste à convertir un message textuel en un fichier audio où chaque caractère est représenté par un signal DTMF spécifique. Le programme d'encodage va générer ces signaux et les enregistrer dans un fichier au format WAV.

L'algorithme suit les étapes suivantes :

1. Associer chaque caractère à sa paire de fréquences DTMF.
2. Générer un signal audio pour chaque caractère et l'écrire dans un fichier WAV.
3. Insérer un silence entre les caractères pour faciliter la reconnaissance.

Pour générer un son, on utilise la somme de deux sinusoides correspondant aux fréquences DTMF du caractère. La formule mathématique est la suivante :

$$s(t) = A \times (\sin(2\pi f_{low}t) + \sin(2\pi f_{high}t))$$

où :

- A est l'amplitude du signal,
- f_{low} et f_{high} sont les fréquences DTMF associées au caractère,
- t est le temps d'échantillonnage.

Le nombre de samples générés est déterminé par la relation :

$$N = \text{SampleRate} \times \text{Durée du son}$$

Comme seule contrainte d'encodage, vous avez :

1. Une durée d'un son de 0.2 secondes par caractère.
2. Une pause de 0.2 secondes entre deux caractères.
3. Une pause de 0.05 secondes entre plusieurs pressions pour un même caractère.

4 Décodage DTMF

Le décodage consiste à analyser un fichier audio contenant des signaux DTMF et à en extraire le message textuel d'origine.

L'algorithme suit les étapes suivantes :

1. Lire les échantillons du fichier audio et détecter les segments correspondant aux sons DTMF.
2. Analyser le spectre fréquentiel pour identifier les paires de fréquences présentes.
3. Associer ces paires de fréquences aux caractères correspondants.
4. Restituer le message d'origine en respectant la structure temporelle des signaux.

Afin de vous aider à imaginer comment réaliser le traitement de chaque échantillon audio :

1. Vérifier qu'il ne s'agit pas d'un silence.

2. Détecter la touche pressée.
3. Compter les appuis successifs.
4. Vérifier la pause entre les pressions.
5. Décoder le caractère associé à la touche.
6. Vérifier la pause entre les lettres (silence après la pression de la touche).

Pour votre décodeur, il vous est demandé de fournir deux versions différentes d'analyse de fréquence. Cela a pour objectif de permettre une comparaison des temps d'exécution et d'identifier les avantages/inconvénients de chaque implémentation.

5 Informations

Il vous est fourni sur Cyberlearn une archive contenant un `CMakeLists.txt` ainsi qu'un `main.c` vous montrant comment le programme doit être utilisé.

Comme ce travail demande l'utilisation de la librairie `libsndfile`, un exemple de code de lecture et d'écriture de fichier vous est également fourni.

Vous êtes libre d'utiliser d'autres librairies si vous en trouvez l'utilité.

6 Déroulement du travail

Nous allons vous demander de rendre votre décodeur le plus générique possible. C'est la raison pour laquelle vous avez peu de contraintes pour l'encodage. Avant le début de la seconde session de laboratoire, il vous sera demandé de fournir 1 ou 2 fichiers audio chacun sur Cyberlearn, créés par votre encodeur. Ils devront respecter le format `message_encode.wav`. Il vous seront également fourni des fichiers audio réalisés par nous pour vous permettre de vérifier l'adaptabilité de votre décodeur.

À la fin de la seconde session vous devriez avoir testé votre décodeur avec l'ensemble des fichiers audios disponible sur cyberlearn.

7 Mesure du temps d'exécution

Lorsque l'on cherche à évaluer la performance d'un programme, le premier critère à observer est son temps d'exécution. Un outil que vous devez connaître pour mesurer ce critère est `/usr/bin/time`. Nous vous invitons à lire sa documentation et à l'exécuter sur votre programme pour vous familiariser avec son usage. N'hésitez pas à essayer `hyperfine`, un outil bien plus performant.

Une fois que vous savez mesurer le temps d'exécution, nous allons nous intéresser à l'évolution des temps d'exécution des deux implémentations en fonction de la taille des données fournies. Choisissez des fichiers de tailles différentes, exécutez les deux implémentations, récupérez les temps d'exécution, puis générez un graphe comparant les deux algorithmes. Analysez vos résultats et concluez.