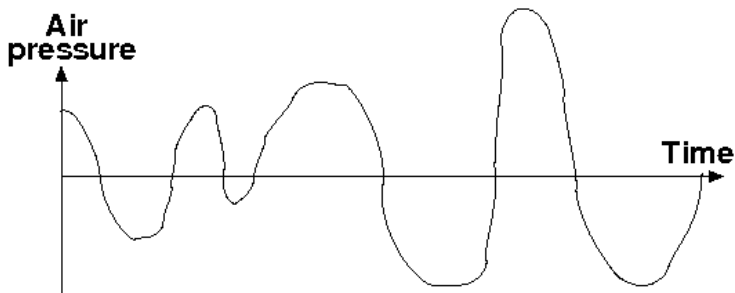


# Systèmes d'exploitation

## Gestion du son

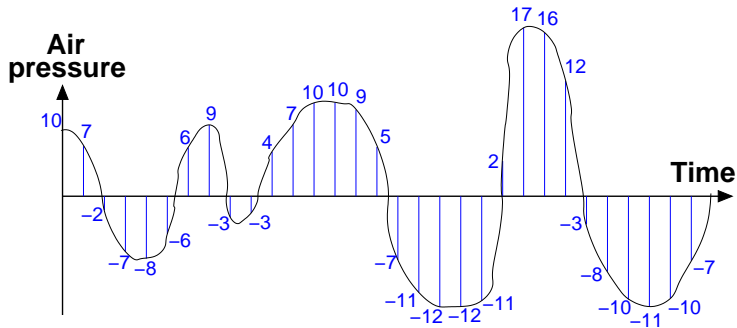
Guillaume Pierre

- Le son est une onde mécanique



- Comment représente-t-on une onde analogique en une représentation numérique?

- Solution: échantillonner l'onde à intervalles réguliers



- Un fichier audio contient la suite des mesures effectuées
  - ▶ Ici: 10, 7, -2, -7, -8, -6, 6, 9, -3, -3, 4, etc.
- La plupart des formats audio compressent les valeurs pour prendre moins de place
  - ▶ Mais les fichiers WAV contiennent simplement les échantillons sans aucune compression (plus facile!)

- Tous les fichiers audio commencent par un **en-tête** pour indiquer comment le fichier a été encodé:
  - ▶ Combien de **canaux?** (1=mono; 2=stéréo)
  - ▶ **Fréquence d'échantillonnage?** (les CDs audio contiennent des fichiers WAV échantillonnés à 44100Hz (c'est-à-dire 44100 échantillons par seconde); d'autres fichiers WAV peuvent utiliser d'autres fréquences)
  - ▶ Quelle est la **représentation d'un échantillon?** (entiers 8 bits non-signés ou entiers 16 bits signés)

- La librairie contient une fonction pour lire l'en-tête du fichier:

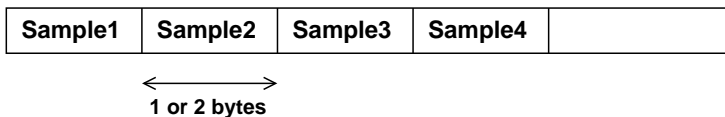
```
#include "audio.h"
int aud_readinit(char *filename, int *sample_rate,
                 int *sample_size, int *channels);
```

- ▶ Indiquer le nom du fichier à ouvrir (filename)
- ▶ Passer des pointeurs vers trois entiers (déjà alloués)
- ▶ La fonction va:
  - 1 Ouvrir le fichier
  - 2 Lire l'en-tête WAV et écrire la fréquence d'échantillonnage, la taille des échantillons et le nombre de canaux dans vos variables
  - 3 Retourner le descripteur du fichier (qui pointe juste après l'en-tête WAV)
- ▶ Vous pourrez ensuite lire les échantillons du fichier en appelant simplement la fonction read()

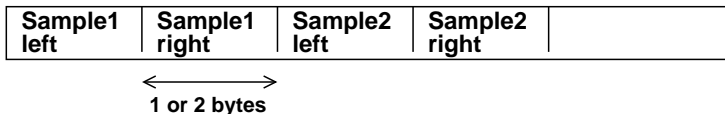
# Organisation des échantillons dans un fichier WAV

- Un fichier WAV contient simplement les différents échantillons l'un après l'autre

- ▶ Mono:



- ▶ Stéréo:



- N'oubliez pas de fermer le fichier après usage!

- Vous devez d'abord ouvrir le “device audio” et déclarer quel genre de fichier WAV vous allez fournir:

```
#include "audio.h"
int aud_writeinit(int sample_rate, int sample_size, int channels);
```

- ▶ Il faut passer la fréquence d'échantillonnage, la taille des échantillons et le nombre de canaux qui correspondent au fichier que vous voulez jouer
  - ▶ Attention: vous ne pouvez pas passer n'importe quelles valeurs et espérer que cela va marcher. Si vous passez des valeurs “bizarres” le système choisira une valeur proche à votre place...
- La fonction `aud_writeinit()` retourne un descripteur de fichier (ou -1 en cas d'erreur)
  - ▶ Il suffit d'écrire vos échantillons dedans!
  - ▶ Et ne pas oublier de fermer le fichier ensuite...

- Je vous demanderai de programmer des petits *filtres* qui transforment le signal sonore
  - ① **Facile:** mentir au device audio
    - ★ Exemple: le fichier est codé à 44100Hz mais vous prétendez qu'il est à 22050Hz
    - ⇒ La carte son va jouer le son deux fois moins vite!
  - ② **Plus intéressant:** manipuler les échantillons!
    - ★ Echanger les deux canaux
    - ★ Pour ajuster le volume: multiplier tous les échantillons par une constante
    - ★ Ajouter de l'écho
    - ★ Etc.