

Sinfo2 - Reallocation

29. Fév. 2021

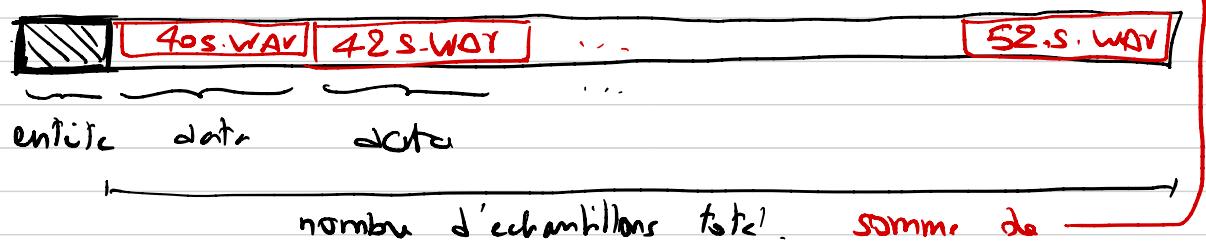
Fichier txt

`melody.txt` : Liste des .WAV

→ Lire le fichier, extraire `fs`, `numSamples`.

			fs	N
1	TD20210426-data/40s.wav	wav filename:	TD20210426-data/40s.wav	22050
2	TD20210426-data/42s.wav	wav filename:	TD20210426-data/42s.wav	22050
3	TD20210426-data/44s.wav	wav filename:	TD20210426-data/44s.wav	22050
4	TD20210426-data/45s.wav	wav filename:	TD20210426-data/45s.wav	22050
5	TD20210426-data/47s.wav	wav filename:	TD20210426-data/47s.wav	22050
6	TD20210426-data/49s.wav	wav filename:	TD20210426-data/49s.wav	22050
7	TD20210426-data/51s.wav	wav filename:	TD20210426-data/51s.wav	22050
8	TD20210426-data/52s.wav	wav filename:	TD20210426-data/52s.wav	22050

but final.



Méthode pour créer la zone mémoire complète.

- 1) faire la somme de N puis faire un malloc en une fois.
- 2) allouer et d'agrandir au fur et à mesure la zone mémoire nécessaire.

fonction: `realloc` (pointeur, dimension en octets)

Création du 1^{er} bloc: `realloc` avec un pointeur NULL

```
char *p = NULL;
char *newp = NULL;
newp = (char *) realloc ( p, 10 * sizeof ( char ) );
```

NULL => malloc.

```
if ( newp != NULL ) // → OK
    p = newp;
else
    // → erreur.
```

Augmentation du bloc:

```
newp = (char *) realloc ( p, 20 * sizeof ( char ) );
```

\neq NULL

10x: 83660 samples.

TD20210928: - reprend le code de TD20210726

- faire un parcours des .WAV
- Allocation / Réallocation mémoire pour les échantillons.
- free.

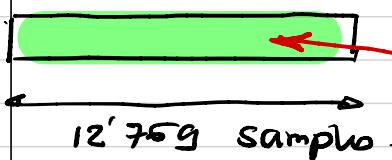
→ ✓ Alloc. / realloc fonctionnel.

Transfert des "data" des .WAV → zone allouée dynamiquement.

```
newT = (sStereoSample *)realloc(t, totalNumSamples * sizeof(sStereoSample));  
if(newT!=NULL) {  
    t = newT;  
    // TODO: read data from wav file  
}  
else {  
    printf("Error while allocating %lu bytes\n", totalNumSamples * sizeof(sStereoSample));  
}
```

1^{er}
WAV

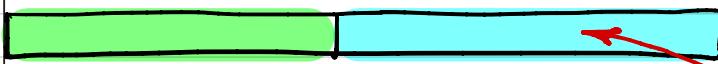
t:



40s.WAV

2^{er}
WAV

t:



48s.WAV

Transférer les data des .WAV → zone allouée dynamique

1 → 48

```
→ TD20210429 git:(main) ✘ ./app  
wav filename: TD20210429-data/40s.wav      22050   12769  
Allocation 12769 samples...  
wav filename: TD20210429-data/42s.wav      22050   11746  
Allocation 24515 samples...  
wav filename: TD20210429-data/44s.wav      22050   10824  
Allocation 35339 samples...  
wav filename: TD20210429-data/45s.wav      22050   8894  
Allocation 44233 samples...  
wav filename: TD20210429-data/47s.wav      22050   10559  
Allocation 54792 samples...  
wav filename: TD20210429-data/49s.wav      22050   9655  
Allocation 64447 samples...  
wav filename: TD20210429-data/51s.wav      22050   9816  
Allocation 74263 samples...  
wav filename: TD20210429-data/52s.wav      22050   9397  
Allocation 83660 samples...
```

Analyse pour le transfert des données wav → tableau.

1^e wav



numSamples = 12769

total/NumSamples = 12769

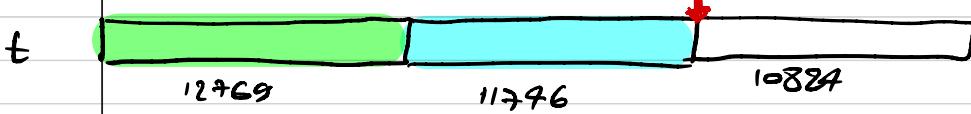
transfert des données du .WAV ↓



numSamples = 11796

total/NumSamples = 24515

transfert des données du .WAV ↓



numSamples = 10824

transfert des données du .WAV ↓ total/NumSamples = 35339

→ : adresse où stocker les data depuis le .wav.

1^e cas :

2 :

3 :

t

$t + 12769$

$t + 24515$

totalNumSamples - numSamples.

Après la mallocation:

new function `getWavData (filename, adresse)`

- ouvrir filename
- "passer" l'entête
- copier les data → adresse

Fin: Génération du .WAV complet

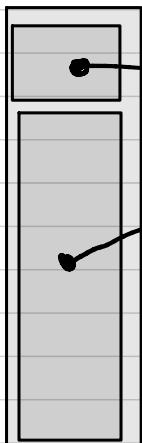


Entête wavPreparHeader. (codeur Entête,
Nombre de canaux,
Sampling Rate,
numSamples
bitPerSample)

sWavHeader h;

wavPreparation (&h,
E_STEREO,
fs,
totalNumSamples
E_16B_SAMPLE);

Fichier WAV binaire "output.wav" "wb"
(mode)



fwrite (&h, sizeof (sWavHeader), 1, fOut)
fwrite (t, sizeof (s8TusSamples), totalNumSamples, fOut)
fclose (fOut)