

Fichier binaire: ① Entête ② données.

File offset (bytes)	field name	Field Size (bytes)
0	ChunkID	4
4	ChunkSize	4
8	Format	4
12	Subchunk1ID	4
16	Subchunk1 Size	4
20	AudioFormat	2
22	NumChannels	2
24	SampleRate	4
28	ByteRate	4
32	BlockAlign	2
34	BitsPerSample	2
36	Subchunk2ID	4
40	Subchunk2Size	4
44	data	Subchunk2Size

①  
entête

②  
données

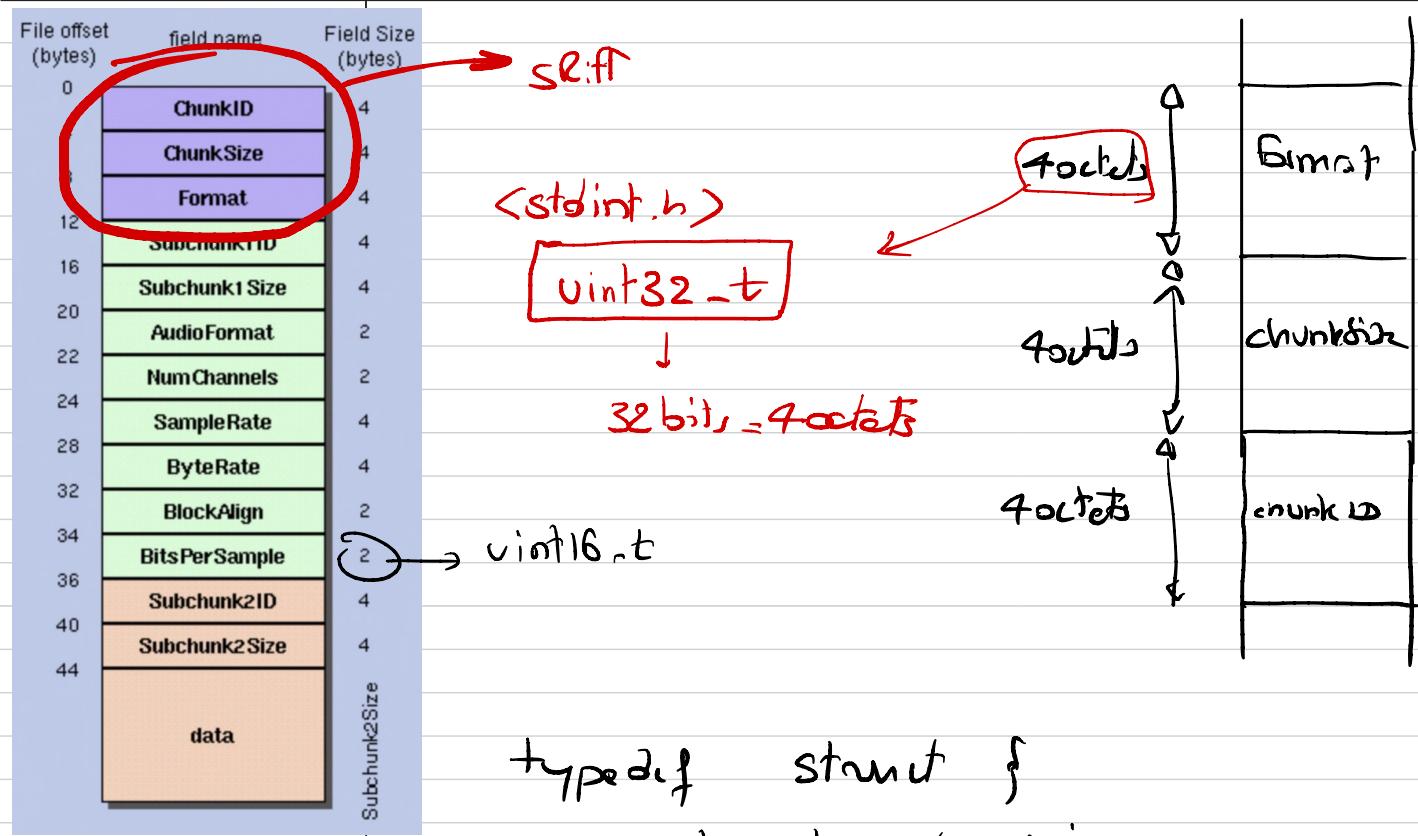


TD 20210318:

- ① Création des 3 structures
- ② Afficher leur taille en octets à l'écran.

fb10

Github → TD → TD 20210318.zip → .png image ↗  
 42. wav  
 99A. wav



typedef struct {

```
    uint32_t chunkID;
    uint32_t chunksize;
    uint32_t format;
```

} sRiff;

printf("size sRiff = %d", sizeof(sRiff));

TD 2021/03/18

③ Création du type SWavHeader comme la réunion des 3 types précédemment créés.

④ Afficher la taille de SWavHeader (44 octets)

16'21

```
typedef struct {
```

```
    uint32_t chunkID;
    uint32_t chunkSize;
    uint32_t format;
```

```
} sRIFF;
```



```
typedef struct {
```

uint32_t	chunkID;	4
uint32_t	chunkSize;	4
uint32_t	format;	4
<b>uint8_t</b>	x;	1
<hr/>		
		13

```
} sTest;
```

size sTest:

13

→

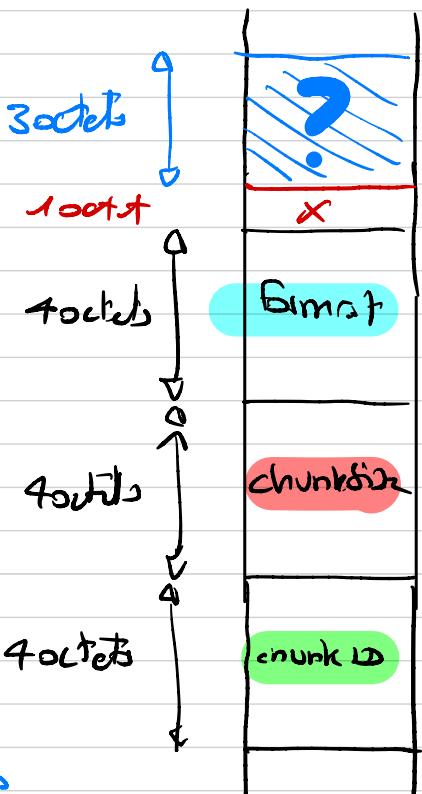
16

!!!

il ajoute dans notre cas

3 octets de padding

Remplissage pour améliorer les performances à l'accès à la mémoire



#pragma pack (1)

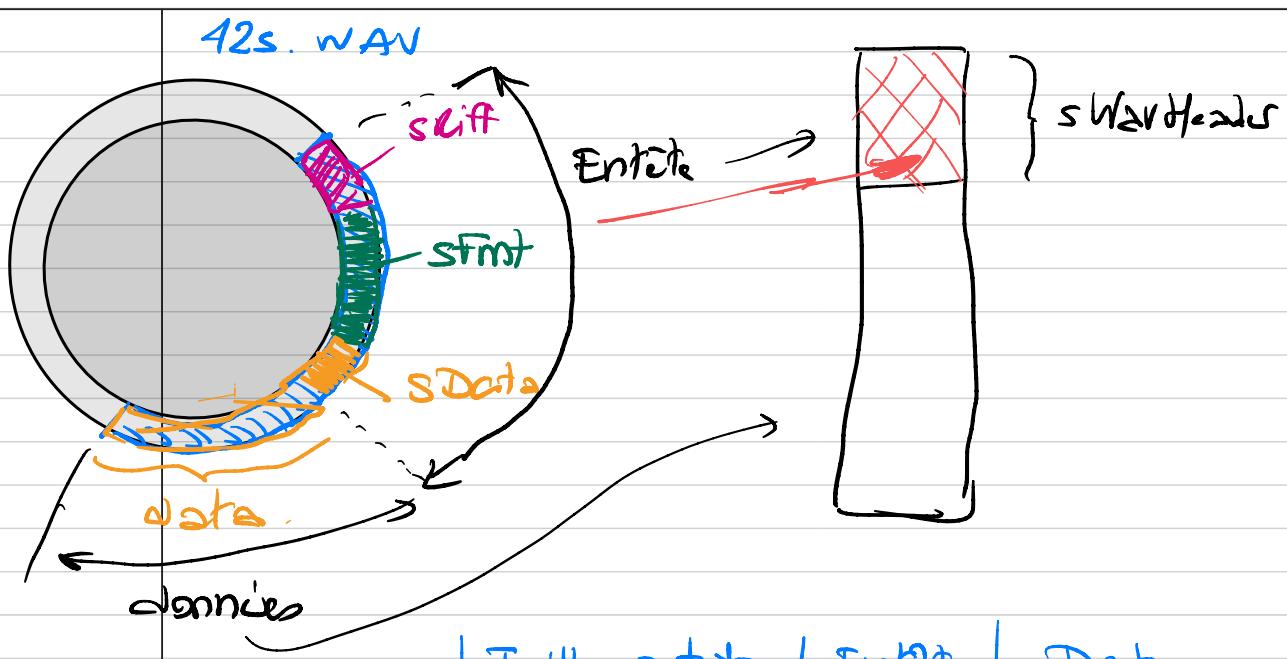
```
13
14     typedef struct {
15
16         uint32_t chunkId;
17         uint32_t chunkSize;
18         uint32_t format;
19     } sRiff;
20
21     typedef struct {
22
23         uint32_t subChunk1Id;
24         uint32_t subChunk1Size;
25         uint16_t audioFormat;
26         uint16_t numChannels;
27         uint32_t sampleRate;
28         uint32_t byteRate;
29         uint16_t blockAlign;
30         uint16_t bitsPerSample;
31
32     } sFmt;
33
34     typedef struct {
35
36         uint32_t subChunk2Id;
37         uint32_t subChunk2Size;
38     } sData;
39
40     typedef struct {
41
42         sRiff riff;
43         sFmt fmt;
44         sData data;
45
46     } sWavHeader;
```

Zone  
Size  
Padding

# pragma pack (0)

# Fichier Binaire WAV

R. FG. 2021



	Taille octets	Entête	Data
42.wav	23 536	44	23 492
42s.wav	47028	99	46 984 <del>2</del>

entête + data

42.wav : mono      1 canal  
 42s.wav : stéréo      2 canaux      ) en durée

Transférer le fichier depuis le disque dur vers la mémoire du programme.

- ① ouvrir le fichier
- ② variable mémoire pour recevoir les données
- ③ Transfert fichier → mémoire.

> ls -al \*.wav

```
-rwxr-xr-x 1 root root 23536 Feb 7 12:16 42.wav
-rwxr-xr-x 1 root root 47028 Feb 7 12:16 42s.wav
```

Transférer le fichier depuis le disque dur vers la mémoire du programme.

- ① ouvrir le fichier
  - ② variable mémoire pour recevoir les données
  - ③ Transfer fichier → mémoire
- 

① FILE \*f = NULL;

**fichier.txt**  
→

filename : "42s.wav"

mode : "rb"

( fopen + fstat f ≠ NULL )

② SWavHeader h;

↳ h.riff.chunkID (1)  
h.fmt.sampleRate (2)  
h.data.subchunk2Size (3)

③ fread

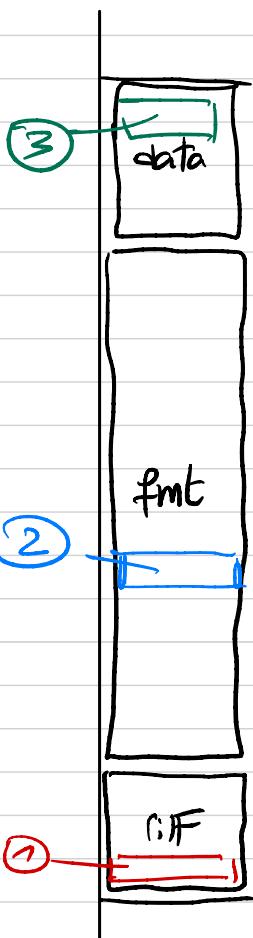
n = fread (&h, sizeof(SWavHeader), 1, f);

zone  
mémorie  
à remplir

dimension de la  
zone

nombre d'éléments  
à lire

le nombre d'éléments lus 1



	lire l'en-tête de	42.wav	42s.wav
afficher:			
les valeurs de	numChannels sampleRate bits Per Sample subchunk2Size	1 22050 16 23492	2 22050 16 46984