

rbuffer.h, un buffer tournant

Bernard Tatin

2013/2017

Contents

1	rbuffer	1
1.1	premières définitions	1
1.2	le code final	2

1 rbuffer

C'est un `buffer tournant` le plus simple possible, capable de gérer des lignes délimitées par *LF* (`'\n'`) mais *CR* (`'\r'`) n'est pas pris en compte.

1.1 premières définitions

Pour limiter les calculs, le code..., la taille du `buffer` est une puissance de 2 d'où la définition du nombre de bits qui ouvre le bal :

```
1  <intro-bits 1>≡ (2b)
    #define _RBUFFER_BITS    8
    #define RBUFFER_SIZE    (1 << _RBUFFER_BITS)
    #define RBUFFER_MASK    (RBUFFER_SIZE - 1)
Defines:
    _RBUFFER_BITS, never used.
    RBUFFER_MASK, never used.
    RBUFFER_SIZE, used in chunk 2a.
```

Il reste encore du travail...

2a $\langle tsrbuffer\ 2a \rangle \equiv$ (2b)
 /**
 * @struct TSrbuffer
 * La structure gérant le buffer tournant.
 */
 typedef struct {
 volatile int in; /**< index du caractère à ajouter */
 volatile int out; /**< index du caractère à sortir */
 volatile int line_count; /**< nombre de lignes contenues dans le tampon */
 volatile char buffer[RBUFFER_SIZE]; /**< le tampon */
 } TSrbuffer;
 Defines:
 TSrbuffer, never used.
 Uses RBUFFER_SIZE 1.

1.2 le code final

2b $\langle * 2b \rangle \equiv$
 $\langle intro-bits\ 1 \rangle$
 $\langle tsrbuffer\ 2a \rangle$

Defined Chunks

$\langle * 2b \rangle$ 2b
 $\langle intro-bits\ 1 \rangle$ 1, 2b
 $\langle tsrbuffer\ 2a \rangle$ 2a, 2b

Index

_RBUFFER_BITS: 1
 RBUFFER_MASK: 1
 RBUFFER_SIZE: 1, 2a
 TSrbuffer: 2a