

Fichiers TXT

- nom + chemin (absolu ou relatif)
- mode "w" pour l'écriture
- création fopen() FILE *
- ↳ gestion des erreurs
- écriture fprintf()
- fermeture fclose()) aussi

Lecture d'un fichier TXT

- nom + chemin
- mode "r" comme read
- ouverture fopen() FILE *
- à la fin : fermeture fclose()
- 3 modes d'accès

① Par caractère

'a' 'n' 'g' ...

② Par ligne

"1.12, +0.955, +0.999\n"

③ champs par champs

⚠ on doit connaître le format du fichier

	Info2-TIN-C > TD20210304 > data.csv
1	angle,sinus,cosinus
2	0,+0.000,+1.000 \n par char.
3	1,+0.017,+1.000
4	2,+0.035,+0.999 \n par ligne
5	3,+0.052,+0.999
6	4,+0.070,+0.998 par champs
7	5,+0.087,+0.996 délimiteur des champs
8	6,+0.105,+0.995
9	7,+0.122,+0.993
10	8,+0.139,+0.990
11	9,+0.156,+0.988
12	10,+0.174,+0.985

lecture en mode caractère par caractère

char c = ' ' ; // signe !

c = fgetc (fp) ; // lit 1 char depuis fp

↳ renvoie -1 si c'est la fin du fichier.

EOF: End Of File

TD 2021/08/28:

- 1) ouverture du fichier "data.csv" en lecture
- 2) comptage du nombre de caractères dans le fichier
- 3) affichage du nombre de caractères.

15:35

Analyse

compteur "count" val. init à un entier non signé.
caractère lu "c"
char signé

boucle pour lire et compter les caractères
jusqu'à la détection de EOF.
si le caractère lu ≠ EOF ⇒ incrémenter count.

boucle: ~~For~~ | ~~While~~ | do... while

do

[c ← lire un caractère depuis le fichier
if c ≠ EOF
[COUNT ← COUNT + 1
END if
while c ≠ EOF

lire ligne par ligne.



on récupère une chaîne de caractères

hypothèse: la chaîne qui suit la ligne lire depuis le fichier est assez grande.

ex: data.csv : 1 ligne ≈ 18 car. $\rightarrow 1024$
`#define MAX_LENGTH 1024`

char s[MAX_LENGTH] = {0};

char * $\leftarrow \text{fgets}(s, \text{MAX_LENGTH}, \text{fp});$

\hookrightarrow NULL si fin de fichier
 \hookrightarrow s si lecture correcte

TD20210308b: compter et afficher le nombre de lignes du fichier "data.csv"

s[1023]

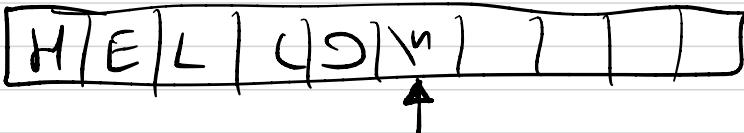
variable "r" comme résultat du fgets (char*)
compteur "count" val. init à un signalé int
boucle pour les lignes
r指向 une ligne
si r \neq NULL alors incrementer count
tant que r \neq NULL

#include <string.h>

NULL \rightarrow strchr(s, '\n')

si pas trouvé

adresse du '\n' dans s si trouvé.



s[0] & s[0]

s[0]

2034

lecture par champs

Info2-TIN-C > TD20210308b > data.csv	
1	
2	0,+0.000,+1.000
3	1,+0.017,+1.000
4	2,+0.035,+0.999
5	3,+0.052,+0.999
6	4,+0.070,+0.998
7	5,+0.087,+0.996

```
char *s = "1,0.017,1.000";
```

The diagram shows the format string "1,0.017,1.000" with annotations below it:
- The first '1' is labeled 'int'.
- The comma after '1' is labeled 'double'.
- The '0.' part of '0.017' is labeled 'double'.
- The '0.017' part of '0.017' is labeled 'double'.
- The comma after '0.017' is labeled 'double'.
- The '1.' part of '1.000' is labeled 'double'.
- The '000' part of '1.000' is labeled 'double'.

Clavier

```
ret = scanf("%d,%lf,%lf", &a, &si, &co);
```

si tout va bien : r ← 3

```
ret = sscanf(s, "%d,%lf,%lf", &a, &si, &co);
```

si tout va bien : r ← 3

```
ret = fscanf(fp, "%d,%lf,%lf", &a, &si, &co);
```

si tout va bien : r ← 3