==========

Insérer un mot à partir d'une position n dans chaque ligne d'un fichier texte

C.Turrier – 18 juin 2023

==========

On peut avoir besoin d'insérer, un mot donné, ne contenant pas de caractère '\n' (OA en hexa) de passage à la ligne, à partir d'une position n donnée, dans chaque ligne d'un fichier texte (pouvant contenir ou non un caractère '\n' de passage à la ligne). Cela peut être utile pour formater un texte pour une utilisation dans des pages HTML ou pour créer un texte de documentation.

Par exemple :

```
0001 \documentclass[14pt,twoside, openright]{extbook}
0002 %=================================================
0003 % PAQUETS PRINCIPAUX
0004 %=================================================
0005 \usepackage[T1]{fontenc}
0006 \usepackage[utf8]{inputenc}
0007 \usepackage[french]{babel}
...
```

devient

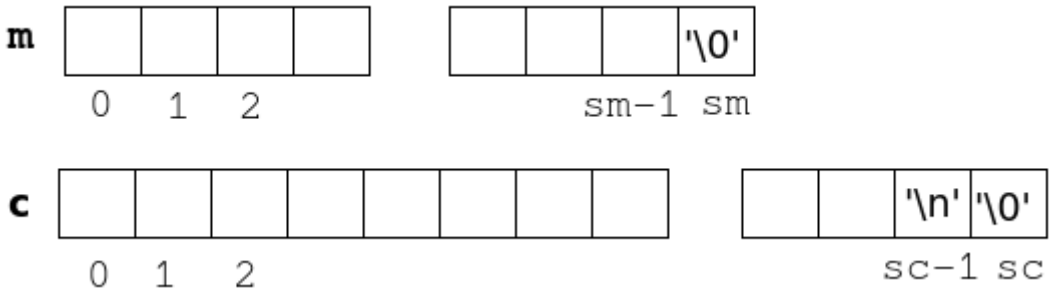
```
0001-> \documentclass[14pt,twoside, openright]{extbook}
0002-> %=================================================
0003-> % PAQUETS PRINCIPAUX
0004-> %=================================================
0005-> \usepackage[T1]{fontenc}
0006-> \usepackage[utf8]{inputenc}
0007-> \usepackage[french]{babel}
...
```

On insère le mot m="-> " de taille sm=2 à partir de la position n=4 (en partant de 0) de chaque ligne c.

Soient
- char **m**[80]; un mot
- int **sm**; la taille du mot
- char **c**[128]; une ligne
- int **sc**; la taille de la ligne
- int **n**; le n° d'emplacement, dans la ligne line, à partir duquel on souhaite insérer le mot m

Remarque :
Il faut naturlellement que n soit inférieur à sc-1

`size(m)=sm et size(c)=sc`
(positions 0 à sm-1 et 0 à sc-1 car l'intruction size() ne prend pas en compte le caractère '\0' de fin de ligne)



Le programme **ins.c** suivant, écrit en langage C, réalise ce travail automatiquement, ce qui est bien pratique lorsque le nombre de lignes à traiter est important.

La définition du mot m à insérer, ainsi que sa position de départ n dans chaque ligne du fichier texte, sont précisées au début du fichier **ins.c**.

Pour compiler ce programme avec GCC, afin d'obtenir le fichier exécutable **ins**, il suffit de saisir la commande suivante depuis le terminal ouvert dans le répertoire où se trouve **ins.c** :

```
gcc -Wall ins.c -o ins
```

Il suffit ensuite de placer l'exécutable **ins** dans le répertoire où se trouve le fichier **in.txt** dont les lignes sont à traiter".

Pour exécuter le programme, il suffit alors de saisir la commande suivante :

```
./ins
```

Le fichier **out.txt** est alors créé automatiquement et contient les lignes modifiées.

Il faut déclarer, au début du programme le mot m à insérer dans chaque ligne du fichier in.txt ainsi que la position n de début d'insertion du mot dans chaque ligne. Dans notre exemple :

```
//-----------------------
strcpy(m, "->");
n=4;
//------------------------
```

Code source du programme

```
/*************************************************************
Insère mot - Auteur C.Turrier - 18 juin 2023
source: ins.c
compilation: gcc -Wall ins.c -o ins
exécution: ./ins
Ce programme  un mot m donné, ne contenant pas de caractère
'\n' (OA en hexa) de passage à la ligne,  à partir d'une
position n donnée, dans chaque ligne c (pouvant contenir
ou non un caractère '\n' de passage à la ligne) d'un
fichier texte in.txt
Il crée un fichier résultat out.txt à partir du fichier in.txt
*************************************************************/
#include <stdio.h>
#include <string.h>
int main(int argc, char* argv[])
{
FILE* pout = fopen("out.txt", "w+");
FILE* pin = fopen("in.txt", "r");
char m[80];
char line[128];
char newline[128];
int i,n;
int size, taille;
//-----------------------
strcpy(m, "->");
n=4;
//-----------------------
taille=strlen(m);
while (fgets(line, sizeof(line), pin))
{
size=strlen(line);
for(i=0;i<n;i++)
{
newline[i]=line[i];
}
for(i=n;i<n+taille;i++)
{
newline[i]=m[i-n];
}
for(i=n+taille;i< taille+size;i++)
{
newline[i]=line[i-taille];
}
newline[taille+size]='\0';
fputs(newline,pout);
}
fclose(pin);
fclose(pout);
return 0;
}
```

Rappel

En langage C, si on déclare char s[128]; puis qu'on écrit l'instruction strcpy(s, "abc");
on obtient les résultats suivants :
s[0] = 'a'
s[1] = 'b'
s[2] = 'c'
s[3] = '\0' (le caractère nul de fin de chaîne)

Dans ce cas, strlen(s) renverra la valeur 3, car il compte tous les caractères précédant le caractère nul de fin de chaîne.

Si on a un caractère de fin de ligne '0A' à la fin de la chaîne s, on a
s[0] = 'a'
s[1] = 'b'
s[2] = 'c'
s[3] = '\n' (le de fin de ligne)
s[4] = '\0' (le caractère nul de fin de chaîne)

Dans ce cas, strlen(s) renverra la valeur 4, car il compte tous les caractères précédant le caractère nul de fin de chaîne.

Fonctionnement du programme

Le programme effectue les actions suivantes :

1. Il inclut les bibliothèques standard stdio.h et string.h nécessaires pour les opérations d'entrée/sortie et de manipulation de chaînes de caractères, respectivement.
2. Dans la fonction main(), il ouvre deux fichiers : "out.txt" en mode écriture et lecture ("w+") dans le fichier pointé par pout, et "in.txt" en mode lecture ("r") dans le fichier pointé par pin.
3. Il déclare plusieurs variables, notamment m de type char pour stocker une chaîne de caractères, line et newline de type char pour stocker des lignes lues à partir du fichier d'entrée et des lignes modifiées, respectivement, i et n de type int pour les compteurs et size et taille de type int pour stocker la taille des lignes.
4. Le programme copie la chaîne "->" dans m à l'aide de la fonction strcpy().

5. Le programme calcule la taille de la chaîne m à l'aide de la fonction strlen() et la stocke dans taille.
6. Ensuite, le programme lit chaque ligne du fichier d'entrée pin à l'aide de la fonction fgets(). La lecture s'arrête lorsque la fin du fichier est atteinte.
7. Pour chaque ligne lue, le programme effectue les opérations suivantes :
    - Il copie les n premiers caractères de la ligne dans newline à l'aide d'une boucle for.
    - Il copie les caractères de la chaîne m dans newline à partir de la position n jusqu'à n+taille.
    - Il copie les caractères restants de la ligne dans newline à partir de la position n+taille jusqu'à la fin de la ligne.
    - Il ajoute un caractère nul ('\0') à la fin de newline pour former une chaîne de caractères valide.
8. Le programme écrit la chaîne newline modifiée dans le fichier de sortie pout à l'aide de la fonction fputs().
9. Après avoir parcouru toutes les lignes du fichier d'entrée, le programme ferme les fichiers pin et pout avec les fonctions fclose().
10. Enfin, le programme se termine et retourne 0, indiquant ainsi une exécution réussie.

En résumé, ce programme lit les lignes d'un fichier d'entrée, modifie chaque ligne en ajoutant la chaîne "->" à une position spécifiée par n, puis écrit les lignes modifiées dans un fichier de sortie.