

9/8.22

Editeur de texte, ce qui se fait de plus simple

LE PLUS SIMPLE DES ÉDITEURS DE TEXTE

Las de rechercher parmi vos dizaines de disquettes un traitement de texte pour écrire trois lignes de commandes utilisées par CPM, vous vous dites que les programmeurs sur compatibles PC sont beaucoup plus chanceux que vous, car il possèdent la commande :

COPY CON FICHER.TXT

qui permet de sauvegarder automatiquement quelques lignes de textes directement à partir du clavier.

Stoppez donc vos lamentations, et retenez votre bras prêt à lancer l'unité centrale de votre CPC par la fenêtre, car la simulation de cette commande est parfaitement possible sur votre micro-ordinateur préféré.

Il vous suffit pour cela, d'entrer en mode direct, sous Basic les commandes suivantes :

```
OPENOUT "FICHER.TXT"  
PRINT #9, "1ere ligne de texte"  
PRINT #9, "2eme ligne de texte"  
PRINT #9, ... ..  
.....  
PRINT #9, "dernière ligne de texte"  
CLOSEOUT
```

Ces lignes permettent l'ouverture du fichier "FICHER.TXT" (**OPEN-OUT**) sur l'unité de disquette, puis mémorisent une à une les lignes de textes dans le buffer d'écriture. Ces dernières seront effectivement sauvegardées sur la disquette lors de l'exécution de **CLOSEOUT**.

Bien sûr, pour gagner en rapidité, vous pourrez remplacer l'instruction **PRINT** par le ? (point d'interrogation), raccourci le plus connu des programmeurs Basic.

Une variante de ces commandes permet aussi de sauvegarder un programme Basic sous forme ASCII, il suffit d'entrer les commandes suivantes :

```
OPENOUT "PROGRAM.ASC"
LIST #9
CLOSEOUT
```

Ces trois commandes permettent de rediriger le listing du programme sur l'unité de disquette (#9), au lieu de l'écran (LIST équivaut en fait à LIST #0, ou encore sur d'autres fenêtres LIST #1, LIST #2..., ou même sur l'imprimante LIST #8).

UN ÉDITEUR 40 COLONNES ÉVOLUÉ, TRÈS SIMPLE

L'éditeur de texte que nous vous proposons plus haut se suffisait à lui-même pour l'entrée de quelques lignes de commande sans modification possible d'une ligne déjà entrée.

Il est parfois utile de posséder un éditeur de texte rapide à charger pour écrire une dizaine de lignes, qu'il sera éventuellement possible de modifier en cas d'erreur.

Aussi, nous vous proposons un programme très court, qui vous permettra d'entrer jusqu'à 23 lignes de 39 caractères, qui seront automatiquement sauvegardées sur disquette.

```
10 REM *****
20 REM **
30 REM ** UN EDITEUR DE TEXTE **
40 REM ** DES PLUS SIMPLES **
50 REM ** SUR 40 COLONNES **
60 REM **
70 REM *****
80 REM
90 MODE 1
100 CALL &BB4E
110 LOCATE 1,24
120 PRINT CHR$(24); "<CONTROL><Z> POUR TE
RMINER";CHR$(24);
130 POSITIONCOLONNE% = 1
140 POSITIONLIGNE% = 1
150 REM
160 WHILE A$ <> CHR$(26)
170     CURSOR 1
180     A$ = INKEY$
190     IF A$=CHR$(240) THEN POSITIONLIGN
E% = POSITIONLIGNE% - 1:GOTO 280
```

```
200     IF A$=CHR$(241) THEN POSITIONLIGN
E% = POSITIONLIGNE% + 1:GOTO 280
210     IF A$=CHR$(242) THEN POSITIONCOLO
NNE%= POSITIONCOLONNE% - 1:GOTO 280
220     IF A$=CHR$(243) THEN POSITIONCOLO
NNE%= POSITIONCOLONNE% + 1:GOTO 280
230     IF A$ = "" THEN GOTO 280
240     IF A$ = CHR$(13) THEN POSITIONCOL
ONNE% = 1:POSITIONLIGNE% = POSITIONLIGNE
% + 1:GOTO 280
250     IF A$ = CHR$(26) THEN GOTO 330
260     PRINT A$;
270     POSITIONCOLONNE% = POSITIONCOLONN
E% + 1
280     IF POSITIONCOLONNE% > 39 THEN POS
ITIONCOLONNE% = 1
290     IF POSITIONCOLONNE% < 1 THEN POSI
TIONCOLONNE% = 39
300     IF POSITIONLIGNE% < 1 THEN POSITI
ONLIGNE% = 23
310     IF POSITIONLIGNE% > 23 THEN POSIT
IONLIGNE% = 1
320     LOCATE POSITIONCOLONNE%,POSITIONL
IGNE%
330 WEND
340 LOCATE 1,24
350 PRINT CHR$(24);"NOM DU FICHER (8 LE
TTRES MAXI): ";CHR$(24);
360 LINE INPUT NOMDUFICHIERS$
370 NOMDUFICHIERS$ = NOMDUFICHIERS$ + ".TX
T"
380 OPENOUT NOMDUFICHIERS$
390 FOR POSITIONLIGNE% = 1 TO 23
400     FOR POSITIONCOLONNE% = 1 TO 39
410         LOCATE POSITIONCOLONNE%,POSITI
ONLIGNE%
420 CURSOR 1
430         A$ = COPYCHR$(#0)
440         IF A$ = "" THEN A$ =CHR$(32)
450         PRINT #9,A$;
460     NEXT POSITIONCOLONNE%
470     PRINT #9
480 NEXT POSITIONLIGNE%
490 CLOSEOUT
```

Ce programme, que nous vous conseillons de placer sur chacune de vos disquettes lors du formatage, une fois lancé, vous proposera un écran de 39 colonnes sur 23 lignes sur lequel vous pourrez entrer vos textes, en mode plein écran.

Le déplacement dans la page s'effectue à l'aide des touches fléchées.

Une seule touche de contrôle est testée : le retour chariot (<RETURN>, pour passer à la ligne suivante).

Par soucis de simplicité du programme, nous n'avons pas permis d'autres fonctions, aussi les corrections seront effectuées en revenant sous les caractères concernés et en les frappant directement.

Pour effacer du texte, il vous suffira de frapper sur la touche **ESPACE**.

La sauvegarde du texte s'effectuera par l'appui simultané sur les touches <CONTROL> et <Z>, et après avoir répondu au message vous demandant le nom du fichier ; l'extension **.TXT** est automatiquement ajoutée à celui-ci.

L'ÉDITEUR 80 COLONNES

Toujours par soucis de simplicité, mais pour les programmeurs plus exigeants sur la longueur des lignes de texte, voici le listing du programme permettant d'entrer 23 lignes de texte sur 79 colonnes.

```

10 REM *****
20 REM **
30 REM ** UN EDITEUR DE TEXTE **
40 REM ** DES PLUS SIMPLES **
50 REM ** SUR 80 COLONNES **
60 REM **
70 REM *****
80 REM
90 MODE 2
100 CALL &BB4E
110 LOCATE 1,24
120 PRINT CHR$(24);" <CONTROL><Z> POUR TE
RMINER";CHR$(24);
130 POSITIONCOLONNE% = 1
140 POSITIONLIGNE% = 1
150 REM
160 WHILE A$ <> CHR$(26)
170     CURSOR 1
180     A$ = INKEY$
190     IF A$=CHR$(240) THEN POSITIONLIGN
E% = POSITIONLIGNE% - 1:GOTO 280

```

```
200     IF A$=CHR$(241) THEN POSITIONLIGN
E% = POSITIONLIGNE% + 1:GOTO 280
210     IF A$=CHR$(242) THEN POSITIONCOLO
NNE%= POSITIONCOLONNE% - 1:GOTO 280
220     IF A$=CHR$(243) THEN POSITIONCOLO
NNE%= POSITIONCOLONNE% + 1:GOTO 280
230     IF A$ = "" THEN GOTO 280
240     IF A$ = CHR$(13) THEN POSITIONCOL
ONNE% = 1:POSITIONLIGNE% = POSITIONLIGNE
% + 1:GOTO 280
250     IF A$ = CHR$(26) THEN GOTO 330
260     PRINT A$;
270     POSITIONCOLONNE% = POSITIONCOLONN
E% + 1
280     IF POSITIONCOLONNE% > 79 THEN POS
ITIONCOLONNE% = 1
290     IF POSITIONCOLONNE% < 1 THEN POSI
TIONCOLONNE% = 79
300     IF POSITIONLIGNE% < 1 THEN POSITI
ONLIGNE% = 23
310     IF POSITIONLIGNE% > 23 THEN POSIT
IONLIGNE% = 1
320     LOCATE POSITIONCOLONNE%,POSITIONL
IGNE%
330 WEND
340 LOCATE 1,24
350 PRINT CHR$(24);"NOM DU FICHIER (8 LE
TTRES MAXI): ";CHR$(24);
360 LINE INPUT NOMDUFICHIER$
370 NOMDUFICHIER$ = NOMDUFICHIER$ + ".TX
T"
380 OPENOUT NOMDUFICHIER$
390 FOR POSITIONLIGNE% = 1 TO 23
400     FOR POSITIONCOLONNE% = 1 TO 79
410         LOCATE POSITIONCOLONNE%,POSITI
ONLIGNE%
420 CURSOR 1
430         A$ = COPYCHR$(#0)
440         IF A$ = "" THEN A$ =CHR$(32)
450         PRINT #9,A$;
460     NEXT POSITIONCOLONNE%
470     PRINT #9
480 NEXT POSITIONLIGNE%
490 CLOSEOUT
```

Ce programme fonctionne comme l'éditeur 40 colonnes, aussi, vous voudrez bien vous reporter aux explications du paragraphe précédent pour son utilisation.

Ces deux programmes n'ont bien sûr pas la prétention de remplacer vos traitements de texte, mais permettent de répondre à un besoin ponctuel de saisie de quelques lignes de texte, rappelons-le sur CPC-6128.

ADAPTATION AU CPC-464

Vous aurez pu remarquer dans ce programme l'utilisation des instructions **CURSOR** et **COPYCHR\$(#0)** aux lignes 170, 420 et 430.

Or, il s'avère que ces fonctions ne sont pas présentes sur les Basic des CPC-464, aussi nous vous proposons de les simuler.

CURSOR 1

CURSOR 1 permet, lors de l'exécution d'un programme de faire apparaître à l'écran le caractère représentant le curseur.

Pour simuler cette instruction, il suffit d'appeler les vecteurs **&BB7B** et **&BB8A**.

&BB7B autorise l'affichage du curseur à l'écran.

&BB8A quant à lui, l'affiche.

Aussi vous remplacerez les instructions des lignes 170 et 420 par :

```
170 CALL &BB7B : CALL &BB8A
```

```
420 CALL &BB7B : CALL &BB8A
```

Signalons que pour éteindre le curseur il suffit d'un second appel du vecteur **&BB8A** (simulation de l'instruction **CURSOR 0**).

COPYCHR\$(#0)

La fonction **A\$ = COPYCHR\$(#0)** permet de lire dans **A\$** le caractère situé sous le curseur (amené avec l'instruction **LOCATE**). Cette fonction est réalisée par le vecteur situé à l'adresse **&BB60**.

Pour simuler cette fonction, il sera nécessaire d'entrer un programme assembleur, heureusement très court dans la mémoire du CPC, que voici sous la forme de mnémoniques :

```

                                COPYCH: EQU $
A000 CD60BB                    CALL 0BB60H
A003 3210A0                    LD  A, (OCTET)
A006 C9                        RET
A010 00                        OCTET:  DEFB 00H
                                END

```

Ce programme que nous implanterons à partir de l'adresse **&A000** nous permettra de connaître le code ASCII du caractère lu à l'écran, placé dans la case mémoire **&A010**.

Ainsi, pour modifier les programmes d'édition de textes précédents, nous vous proposons d'insérer les lignes suivantes :

```
101 MEMORY &9FFF
102 RESTORE 500
103 FOR ADRESSE = &A000 TO &A006
104     READ VALEUR$
105     VALEUR = VAL("&" + VALEUR$)
106     POKE ADRESSE, VALEUR
107 NEXT ADRESSE
```

Les données « pokées » en mémoire seront placées sous forme de DATA, en ligne 500 :

```
500 DATA CD, 60, BB, 32, 10, A0, C9
```

Il est alors possible de modifier l'instruction **A\$ = COPYCHR\$(#0)** en ligne 430 par :

```
430 CALL &A000 : A$ = CHR$(PEEK (&A010))
```

Il vous sera possible d'utiliser cette simulation d'instruction dans tous vos programmes, en vous signalant que celle-ci ne fonctionne que sur la fenêtre **#0**, contrairement à **COPYCHR\$(#fenetre)** qui peut s'utiliser sur toutes les fenêtres de l'écran.

