

SMARKY NEWS

No 8

1er novembre 1979

E T E X (EDITEUR DE TEXTES)

- * La base de ce nouvel éditeur est identique à celle de l'éditeur de programmes. Il permet l'édition de texte "au kilomètre", le retour de chariot étant réservé aux délimitations de paragraphes.
- * La mise en page (écran) peut se faire de trois façons différentes.
 - a) Normale (la plus rapide); les mots sont coupés à droite
 - b) Justifié à gauche en drapeau à droite; les mots ne sont pas coupés
 - c) Justifié à gauche et à droite; des espaces sont insérés entre les mots afin d'aligner à gauche et à droite

EXEMPLE (académique, le nombre de colonnes étant trop petit)

```
ceci est un
n texte qu
i n'est pas
s justifié
```

```
ceci est
un texte
qui est
justifié à
gauche
```

```
ceci est
un texte
qui est
justifié à
gauche et
à droite.
```

* Choix de la justification

- N Normal
 M Justifié à gauche
 , Justifié à droite

Choix de la largeur

- D Simple
 S Double

Choix de la hauteur

- C Simple
 X Double

Choix du générateur de caractères

- U Normal
 I Italique
 O Gras
 P Spécial utilisateur

Choix du soulignage

- H Souligné
 J Cadre bas
 K Cadre haut
 L Cadre complet

- * Un mode d'emploi se trouve sur disque et s'appelle ETEX.HP. Il est visualisé au moyen de H

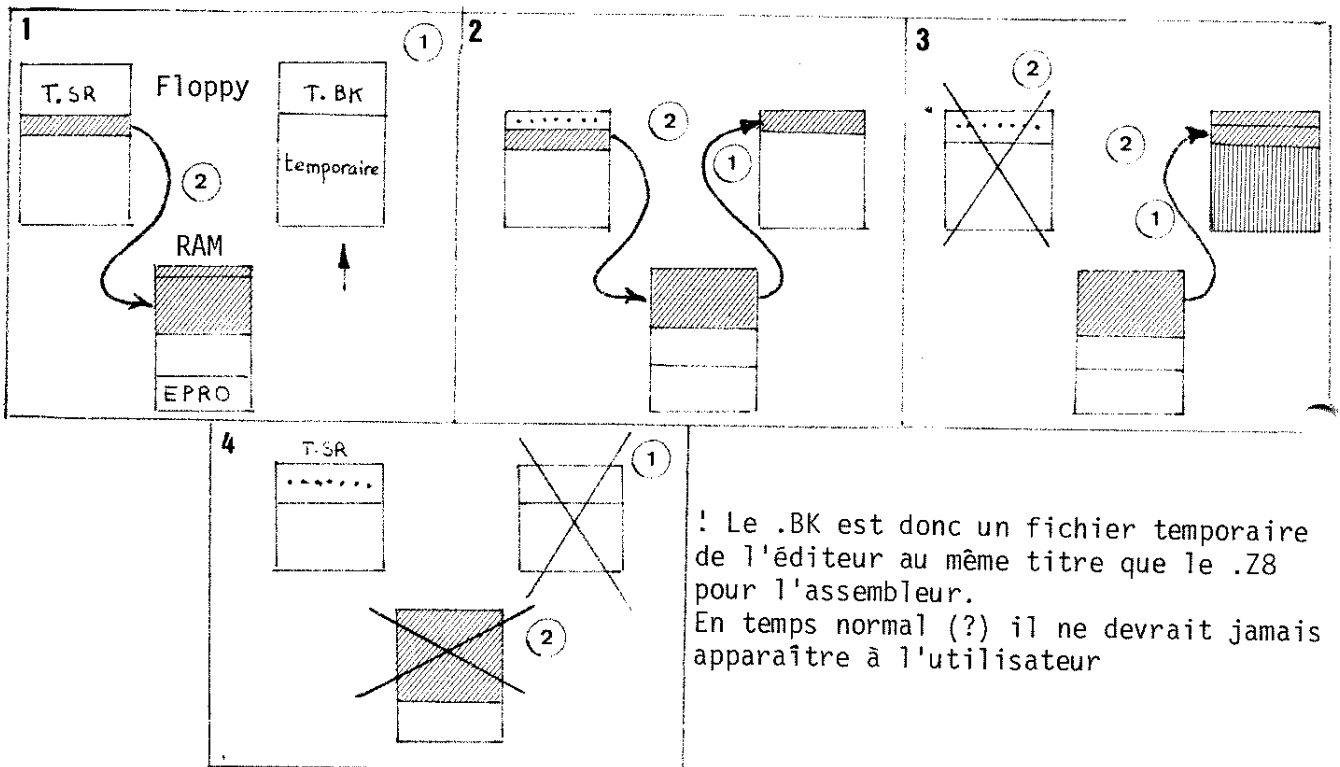
E P R O (EDITEUR DE PROGRAMMES)

- * La vitesse des entrées-sorties a été améliorée, la lecture du floppy se faisant de façon plus optimisée
- * Les pertes de fichiers (.BK) sont maintenant évitées

RESUME DES COMMANDES

- SHOW** **DEFINE** filename insère le fichier après le pointeur jusqu'à concurrence de la mémoire texte disponible
- COPY** **DEFINE** filename recopie le contenu du buffer courant sur le disque
- 1** **COPY** **SHOW** **DEFINE** filename édite un fichier de grande taille en recopiant le buffer courant sur disque et amenant le début du fichier jusqu'à concurrence de 3/4 de la mémoire texte
- 2** **COPY** **SHOW** **DEF** amène la suite du fichier
- 3** **KILL** **CURSOR** **DEFINE** sortie "normale"; termine la copie du vieux fichier sur le nouveau, détruit le vieux et sauve le nouveau
- 4** **KILL** **COPY** **CURSOR** **DEF** sortie "rapide". Ne recopie pas le vieux fichier et détruit l'éventuelle nouvelle copie

EXEMPLES



- * Les noms de tous les fichiers disque peuvent être insérés dans un buffer d'édition au moyen de **PROGRA DEFINE**

- * Un mode d'emploi résumé se trouve sur disque et peut être affiché au moyen de **PROGRA H** Le fichier s'appelle EPRO.HP

ASM (ASSEMBLEUR PARAMETRISABLE)

- * Grâce au travail d'Alain Richard qui a fourni les tables pour les processeurs 8048, 6800 et bientôt Z8000 et 2650, ainsi que les modifications à faire pour les processeurs 8080 et 8085, il est maintenant possible avec le même assembleur d'obtenir du code pour différents processeurs.

Chaque section doit simplement être précédée d'une des pseudos suivantes:
/.Z8000/.Z80/.8080/.8085/.8048/.6800/.2650/.SCMP/

- * Sur le disque se trouvent les fichiers contenant les tables des processeurs:
Z80.ST Z8000.ST 8080.ST 8085.ST 8048.ST
6800.ST 2650.ST SCMP.ST

- * La ligne de commande de ASM est la suivante:

```
ASM      F1      < F2 >
          source & binaire  listing
```

Toute ancienne version d'un fichier nommé est automatiquement détruite.

- * ASM ne génère plus de fichier binaire au format PDP11, mais directement une image mémoire (F.SM).

- * Précisions:

.SM6 cherche sur le disque (DXØ:) le fichier SM6.ST et le lit
.FLO cherche sur le disque (DXØ:) le fichier FLO.ST et le lit
.COB cherche sur le disque (DXØ:) le fichier COB.ST et le lit
.REF F<ST> cherche sur le disque (DXØ:) le fichier F.ST et le lit
(.REF SM6 = .SM6)

.SYM F<ST> génère une nouvelle table des symboles dans le fichier .ST
(l'ancienne est automatiquement détruite).

Il n'est pas indispensable de spécifier l'extension .ST.

Dans un même programme .SYM et { .SM6
.REF } sont compatibles.

Par contre, deux .REF ou .SM6 et .REF sont refusés.

- * Arithmétique:

L'opérateur "NON" logique est le caractère "~" apparaissant sous la forme de "÷" sur l'écran avec le générateur de caractères actuel.

Exemple: .IF FLOPPY } = .IF ~FLOPPY
.ELSE

Les dépassements de capacité 16 bits sont détectés; par exemple:

```
LOAD HL,# 100000 + 2*60000
```

SMILE (EDITEUR-ASSEMBLEUR)

- * SMILE a gagné toutes les améliorations d'EPRO pour les entrées-sorties
- * SMILE a gagné toutes les améliorations d'AZ80 (multi-processeur)
- * Les références à une table des symboles sur disque (.SM6) génèrent les bruits suivants: CLAC vrout - vrout CLOC, contrairement à l'ancienne version:
CLAV vrout CLOC CLAC CLOC CLAC CLOC CLAC CLOC CLAC vrout CLOC.
On y voit une économie de CLOC CLAC appréciable !
- * Un debug symbolique (avec exécution en pas à pas) sera bientôt disponible.

R. Forster



Le SMAKY peut être livré dans un boîtier unique comprenant alimentation, clavier, écran Ball Brother, 2 floppies MICROPOLIS.

Avec 32k RAM et 8 k ROM
ou 64k RAM et 2 k ROM

AFFICHAGE DE TEXTES FIXES SUR L'ECRAN

Pour afficher des textes sur l'écran, les appels ?DITEX et ?TEXT permettent de grouper tous les textes au début ou à la fin du programme et d'appeler plusieurs fois le même texte.

Il peut être pratique (lisibilité plus grande), de mettre le texte dans le programme, à l'endroit où le texte est appelé. Il suffit d'écrire pour cela la routine TEXTIM (texte immédiat) et de l'appeler comme dans le petit programme:

<pre> TEST: LOAD C,# NLI .W ?IDIS CALL TEXTIM .ASCIZ "premier texte" LOAD A,B ... CALL TEXTIM .ASCIZ "second texte" ... </pre>	<pre> ;routine TEXTIM: affiche le texte ;qui suit, continue l'exécution à la ;première instruction suivant le texte ;in: (SP) adr. retour, déb. texte ;out: affiche texte, continue ; à l'adresse suivant e ;mod: TEXTIM: EX (SP),HL .W ?DITEX EX (SP),HL RET </pre>
---	--

Dans la révision 1.8 du système, la routine TEXTIM a été insérée, et répond au numéro d'appel 136. Il suffit donc d'écrire:

```

...
.W    ?TEXTIM
.ASCIZ "TEXTE"
LOAD  A,B
...

```

SMAKY news 8, p.4

ADRESSEZ VOS COMMUNICATIONS A:

EPSITEC-system sa

Chemin de la Mouette, CH - 1092 Belmont