

SMAKY NEWS

10 mai 1979

No 6

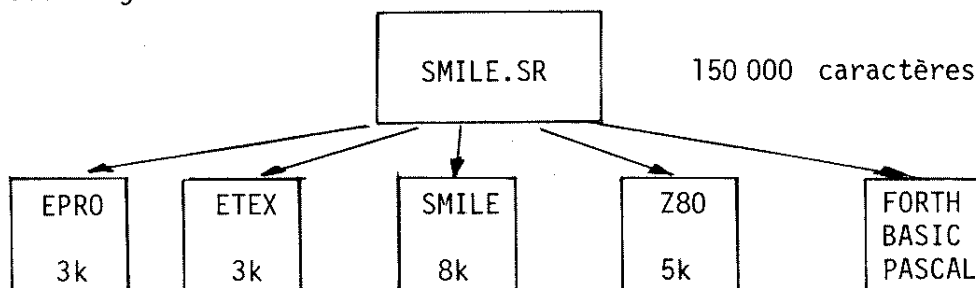
S M I L E 3

Après plusieurs mois d'efforts, une nouvelle révision de SMILE est née; succédant à la révision 2 version 5, un avenir prometteur lui est ouvert.

Malheureusement, cette révision ne fonctionne que dans un environnement software possédant un système d'exploitation. Cet environnement peut être l'un des suivants: floppy (SYSFLO 1-0), cassette rapide 3M, ou système COBUS du LCD-EPFL (COBSYS 1-3). Ceci signifie que les SMAKY "stand alone" ne sont plus supportés par SMILE 3 jusqu'à ce que le software de gestion des périphériques soit au point (SYSPER). Ce software d'environ 2kbytes sera un extrait d'un système d'exploitation complet et ne gèrera que les périphériques. Il est prévu en même temps que la révision 2 du système de base. Le SMAKY doit être pourvu d'un SYSTEME 1-6, ou le futur 2-0 (meilleure gestion des périphériques) pour pouvoir utiliser SMILE 3.

En ce qui concerne l'environnement hardware, un SMAKY 32k RAM suffit.

Avec le SMILE 3 il est possible de générer plusieurs programmes grâce à l'assemblage conditionnel:



* EPRO est un éditeur de programmes (lignes terminées par des ↵) permettant d'éditer des programmes de taille supérieure à la mémoire du SMAKY. Cette taille est finalement limitée par la capacité des disques.

* ETEX possède les mêmes avantages que EPRO, mais il permet l'édition de textes, donc de lignes très longues, le retour de chariot étant réservé à la séparation des paragraphes.

- * SMILE, l'éditeur-assembleur que vous connaissez bien
- * Z80, l'assembleur capable d'assembler de très gros programmes (SMILE par exemple). Il possède les mêmes caractéristiques que l'assembleur de SMILE et génère automatiquement un source avec les erreurs insérées, un listing et un binaire format PDP11.
- * FORTH. Le FORTH en version 3 est en cours d'adaptation (la version standalone est naturellement toujours disponible)
- * BASIC, prévu pour la fin de l'été. Ce sera une sorte de SMILE qui au lieu d'assembler du CALM compilera du BASIC.
- * PASCAL en cours de développement.

QUELQUES NOUVEAUTES DE SMILE 3

- a) L'ordre GET permet, lors de l'édition, d'insérer le contenu d'un buffer autre que le buffer de travail sous le pointeur. Il est réalisé de la façon suivante:

COPY CONTROL 1,...,9

Grâce à cette commande, il est maintenant possible de déplacer une ligne de texte sans quitter le buffer d'édition (COPY ↓ puis COPY CONTROL 0,...,9).

- b) SHOW/COPY DEFINE répondent "file:" et invitent l'utilisateur à taper un nom de fichier qui peut être:
- * le nom d'un fichier disque
 - * le nom d'un périphérique
- (voir respectivement les notices floppy, COBUS et RDOS pour la syntaxe des noms de fichier).

Dans le cas d'un COPY DEFINE sur un fichier qui existe déjà, SMILE demande "DELETE nom ?" et l'utilisateur doit répondre y s'il veut recréer une nouvelle version de ce fichier. Toute action sur une autre touche se solde par un retour à l'éditeur.

- c) DEFINE Le buffer de définition est un buffer spécial (≠ No 9) Il n'est donc plus accessible à l'utilisateur au moyen des touches COPY 0,...,9 ou SHOW 0,...,9. Par contre, l'utilisateur a dix buffers à sa disposition, sans aucune restriction.
- d) L'assembleur gère la mémoire de façon dynamique, ce qui lui permet d'assembler un maximum de binaire et de symboles. Il est possible d'assembler sans problème des programmes de 2k bytes.
- e) De nouvelles pseudo-opérations ont été ajoutées, d'autres complétées et améliorées. Il nous semble utile de répéter encore une fois de façon exhaustive les pseudos reconnues par SMILE.

α) Pseudos de génération

.BYTE	
.WORD	
.B	initialisent les positions mémoire successives avec
.W	les valeurs des expressions qui suivent la pseudo
.BWBW...	

.ASCII génère le code ASCII correspondant au texte qui suit
 .ASCIZ Possibilité d'insérer des valeurs non ASCII au moyen des parenthèses pointues.

Exemple: .ASCIZ /CECI EST UN TEXTE<CR>/

.BLKB Réserve une certaine quantité de mots/bytes sans les initialiser
 .BLKW

β) Pseudos du compteur de locations

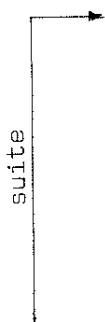
.LOC initialise le PC de l'assembleur (par défaut 0 dans Z80 et 53000 dans SMILE)

.PC spécifie le No du PC à employer (0,...,7)

Exemple.

PCRAM = 1
 PCROM = 2
 RAM = 40000
 ROM = 0

.PC PCRAM
 .LOC RAM
 .PC PCROM
 .LOC ROM



.PC PCROM
 NOP
 LOAD A,VAR
 ...
 RET
 .PC RCRAM
 VAR: .BLKB 1 variables en RAM
 .END

code en ROM

.END signale la fin d'un assemblage (optionnel dans SMILE) et, s'il est suivi d'une expression, initialise l'adresse de start du programme

γ) Bases d'entrée et de sortie

.HEX initialisent la base de sortie de l'assembleur
 .OCT (listing)

.RDX initialisent la base d'entrée, base dans laquelle sont évaluées les expressions
 .RADIX

δ) Contrôles du listing

.TITLE donnent un nom, qui sera imprimé au début de chaque page, au programme et à ses modules
 .SBTTL

.PAGE éjecte une page

.LIST inhibe le listing si la valeur de l'expression qui suit la pseudo est nulle

ε) Contrôles de l'assemblage

.IF assemble les lignes qui suivent la pseudo si l'expression qui suit la pseudo est non nulle

.ELSE inverse le .IF précédent

.ENDIF signale la fin d'une section à assemblage conditionnel

Les .IF peuvent être imbriqués jusqu'à 8. niveaux

ζ) Références externes

.SM6 considère le fichier SM6.ST comme fichier de définition et charge ce fichier dans la table des symboles (floppy, COBUS). Cette instruction doit être située directement après le .TITLE.

.REF \mathcal{F} considère \mathcal{F} comme fichier de définition (ces fichiers se trouvent sur disque). Il n'est pas possible de cumuler des .REF Une seule définition de fichier peut apparaître dans un programme

.SYM \mathcal{F} crée \mathcal{F} comme fichier de définition à la fin de l'assemblage. Ce fichier contient tous les symboles créés par le programme

f) Assemblage sous SMILE: il est possible d'assembler un programme de trois façons différentes:

- PROGRA Z** assemble tout court
- PROGRA L** assemble et génère un listing sur fichier à choix
- PROGRA B** assemble et génère un binaire image mémoire sur fichier à choix

g) TRAP. Le TRAP (367) sera remplacé par 166 qui correspond à l'instruction WAIT à partir de la révision 2 du SYSTEME. La pin "WAIT" sera reliée à la pin "NMI" du Z80 et de ce fait un TRAP aura enfin le même effet qu'un NMI.

Lors de l'exécution de ce TRAP, le contrôle est redonné à SMILE qui se contente de faire un BUZZ et d'attendre que l'utilisateur presse une touche quelconque. Ceci a pour effet d'éviter de "démolir" l'écran.

h) SMILE n'assemble que jusqu'à concurrence de 20. erreurs suite de quoi le message "too many errors" apparaît (ce n'est pas le cas de Z80).

CONCLUSION

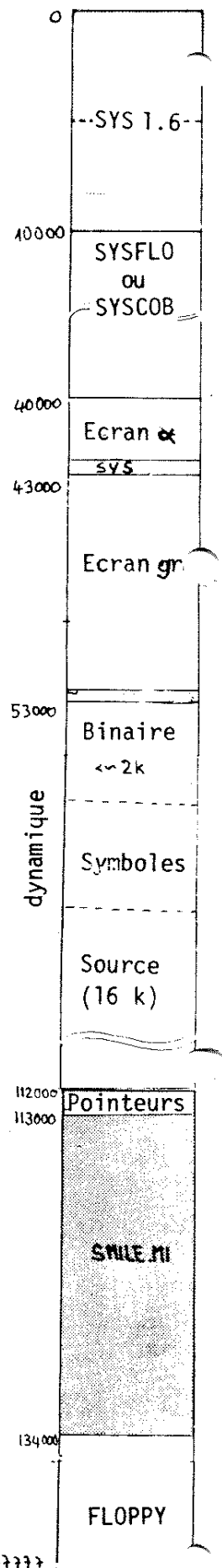
SMILE devient "utilisable", s'est-on laissé dire...

Ronald Forster

SYSTEME REVISION 1-6

Des ennuis avec l'interruption ont permis de découvrir une instruction du Zilog qui fonctionne mal. La philosophie d'interruption a été changée: tous les appels reviennent ION. Ceci a permis de gagner de la place et 8 appels nouveaux ont été ajoutés, qui facilitent sensiblement l'écriture des programmes, surtout si SMILE 3 est à disposition. Une notice et une feuille de codage simplifiée ont été préparées pour les étudiants de l'EPFL.

Smaky News 6 p.4



EPSITEC-system sa

Chemin de la Mouette, CH - 1092 Belmont