

smaky info

Editeur assembleur SMILE 4.0

Michael WALZ

Le logiciel SMILE permet d'éditer, assembler, exécuter et déverminer des programmes écrits en assembleur. L'éditeur fonctionne exactement comme EDIT et ne sera pas expliqué ici.

Les fonctions les plus importantes spécifiques à SMILE peuvent être exécutées avec les touches fonction (F6) à (F11). D'autres fonctions peuvent être atteintes avec le sous-menu **SMILE** du bouton du milieu ou avec la touche super-shift (**PROGRA**) associée à une autre touche.

fichier	AS assemble	PROGRA Z	assemble le source en mémoire
édition	exécute	PROGRA V	exécute le programme assemblé
smile	continue	PROGRA X	continue l'exécution après un EXIT
zones	AS définitions		définitions d'exécution et d'assemblage
divers	montre	PROGRA R	affiche les registres et la mémoire
	sauve binaire	PROGRA B	sauve le binaire assemblé dans un .CODE
	punch PDP11	PROGRA T	transfère le binaire sur #PUNCH en format PDP11
	Simprom	PROGRA S	transfère le binaire dans le simulateur de PROM
	paramètres	PROGRA P	quantité de mémoire pour l'assembleur
	informations	SHOW I	informations diverses
	moniteur	PROGRA Q	va au moniteur

AS assemble

Le source contenu dans la zone de travail est assemblé.

exécute

Exécute le programme préalablement assemblé.

Il ne faut jamais utiliser un des appels système suivants avec un programme qui tourne sous SMILE (sinon votre SMAKY plante!):

NTREL ?IEPBAIP	NTREL ?CHGSTACK
FOS ?STOP	LIB ?OPELIB
LIB ?CLOLIB	

Pour stopper le programme il ne faut pas utiliser FOS ?STOP qui stopperait également SMILE lui-même, mais l'instruction spéciale EXIT. (Ne jamais utiliser EXIT dans un programme qui ne tourne pas sous SMILE, sinon votre SMAKY plante!)

Un programme s'arrête également lorsqu'une erreur d'exécution se produit (Address error, Bus error, Illegal Instruction, Division par zéro etc.).

Les registres sont initialisés de la façon suivante: A4 pointe une zone de mémoire libre d'une longueur d'un Koctet et A6 pointe au début absolu du code de votre programme. Les autres registres sont initialisés à zéro. Ainsi il n'est pas nécessaire de demander de la mémoire par l'appel ?GETMEM si la taille des variables n'est pas supérieure à 1 Koctet; on peut utiliser directement la zone pointée par le registre A4.

 continue

Continue l'exécution du programme après une instruction EXIT; en effet l'instruction EXIT ne sert pas seulement à stopper un programme, mais aussi à l'interrompre à un moment donné pour pouvoir examiner l'état du processeur et de la mémoire et ensuite de continuer l'exécution.

AS ● définitions

<p>Assemblage:</p> <p><input type="checkbox"/> optimisé</p> <p><input type="checkbox"/> avec listing</p> <p><input type="checkbox"/> listing dans zone 12</p> <p><input checked="" type="checkbox"/> symboles fixes</p> <p><input type="checkbox"/> ?ASCOND = TRUE</p> <p><input type="button" value="ASSEMBLE"/></p>	<p>Exécution:</p> <p><input type="checkbox"/> mode pas à pas</p> <p><input type="checkbox"/> sauve la fenêtre si EXIT</p> <p><input checked="" type="checkbox"/> touche FNCT active</p> <p><input type="button" value="EXECUTE"/> <input type="button" value="CONTINUE"/></p>
<p><input type="button" value="D'ACCORD"/> <input type="button" value="ANNULE"/></p>	

Assemblage

- optimisé
Le prochain assemblage sera effectué de manière optimisée
- avec listage
Au prochain assemblage SMILE demande un nom de fichier dans lequel sera mis le listage du programme. (PROGRA)(L) demande tout de suite le nom du fichier listage et lance l'assemblage.
- avec listage dans la zone 12
Au prochain assemblage un listage est produit et tout de suite chargé dans la zone 12. (PROGRA)(K) assemble tout de suite avec listage dans la zone 12.
- ?ASCOND = TRUE
Met le symbole prédéfini ?ASCOND à TRUE, sinon ce symbole vaut FALSE.

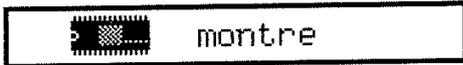
Exécution

- mode pas à pas
Stoppe le programme après chaque instruction exécutée. C'est comme si l'on avait mis EXIT après chaque instruction.
- sauve la fenêtre si EXIT
Au moment de l'EXIT la fenêtre sera sauvée telle qu'elle est visible et elle sera restituée lorsqu'on continue l'exécution du programme. C'est très utile en mode pas à pas car on ne perd pas chaque fois tout ce que le programme a déjà affiché.

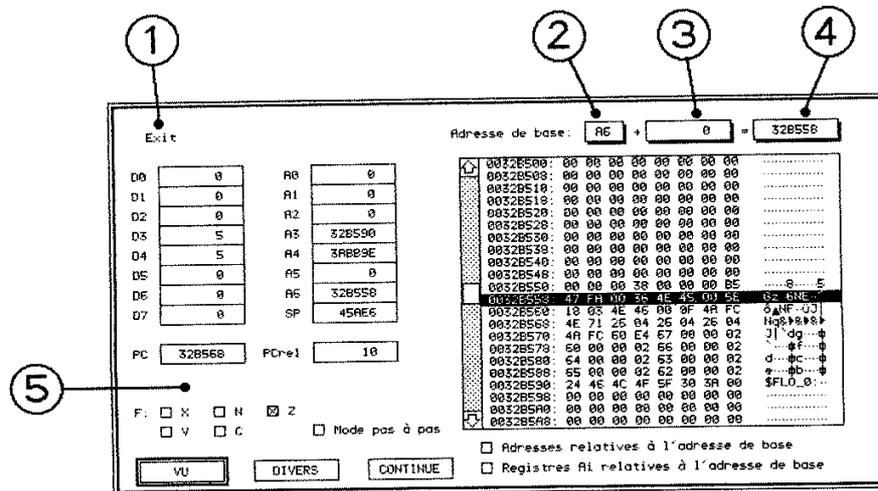
☒ FNCT active

La touche **(FNCT)** permet de stopper un programme qui "boucle". Toutefois il faut enlever ce mode lorsqu'on détermine des programmes avec des processus fils. Sinon votre programme va se bloquer sur l'appel NTREL ?WFABDESC. Si cette situation se présente tout de même, on peut essayer de presser sur **(SHIFT)(FNCT)**, mais pas sur **(FNCT)** tout court. Si vous essayez de stopper un programme avec des processus fils à l'aide de **(FNCT)** votre SMAKY plante généralement.

Les touches **(ASSEMBLE)**, **(EXECUTE)** et **(CONTINUE)** permettent d'assembler, d'exécuter ou de continuer tout de suite sans quitter cette boîte de dialogue.



Affiche l'état du processeur au dernier EXIT et le contenu de la mémoire. Les contenus des registres peuvent être édités en y cliquant avec la souris. (Attention avec SP et A6). Les bits du registre F peuvent être modifiés en cliquant dans les cases correspondantes. (croix allumée: le fanion est à 1, autrement il est à zéro).



On peut visualiser le contenu de la mémoire dans un rayon de 32Koctets autour de l'adresse de base. Cette adresse de base est constituée d'une adresse 32 bits tapée au clavier (case 3) plus éventuellement le contenu d'un des registres d'adresse (case 2). La case 4 indique l'adresse de base. On peut avancer et reculer dans la mémoire à l'aide de l'ascenseur. Un clic dans la case 4 revient à l'adresse de base.

☒ Mode pas-à-pas

enclenche ou déclenche le mode d'exécution pas à pas.

☒ Adresses relatives à l'adresse de base

Si on allume cette croix, les adresses affichées seront relatives par rapport à l'adresse de base, et non absolues.

☒ Registres Ai relatifs à l'adresse de base

affiche les registres A0 à A7 relatifs par rapport à l'adresse de base.

L'emplacement désigné par le chiffre 5 est réservé pour l'affichage du fichier dans lequel le programme s'est arrêté. Si le programme s'est arrêté normalement, rien n'est affiché, mais si le programme s'est arrêté par exemple dans la librairie SCROLL, il sera affiché:

Arrêt dans SMA_SCROLL.LIB. Le PC relatif affiché sera alors relatif par rapport à l'adresse de chargement du fichier correspondant. En effet un programme peut produire une erreur d'exécution en dehors de son propre code (par exemple dans une librairie ou dans un pilote).

DIVERS Affiche une boîte de dialogue avec des informations supplémentaires:

NTREL	Timeout:	infini
	Priorité:	23
Temps CPU	Total:	0 ms
	Dernier:	0 ms
Temps IDLE	Total:	0 ms
	Dernier:	0 ms
Pile utilisée: 0 octets		
<input type="button" value="VU"/>		

Timeout: du processus au moment de l'EXIT

Priorité: du processus au moment de l'EXIT

Temps CPU et temps IDLE total: mesuré depuis le début de l'exécution.

Temps CPU et temps IDLE dernier: mesuré depuis le dernier CONTINUE.

Pile utilisée: Nombre d'octets de pile utilisés par votre programme.

 **sauve binaire**

Permet de sauver le programme assemblé dans un fichier. L'extension par défaut est .CODE mais on peut mettre d'autres extensions.

 **punch PDP11**

Transfère le programme assemblé sur le lien logique #PUNCH en format PDP11. #PUNCH est généralement assigné à une des lignes série du SMAKY.

 **paramètres**

Transfère le programme assemblé dans le simulateur de PROM s'il est connecté.

 **Simprom**

Permet de choisir la taille mémoire qu'on veut donner à l'assembleur. La taille nécessaire dépend fortement de votre programme (longueur, nombre de symboles etc.). Par défaut cette taille est de 80 Koctets ou 60% de la mémoire restante, mais on peut la changer et la mémoriser. 250 Koctets suffisent généralement, même pour les programmes très longs.

 **moniteur**

Pour le spécialiste: permet d'aller dans le moniteur du Smaky. La touche K revient à SMILE.

La prochaine révision de SMILE comprendra un mode dans lequel on voit défiler le listage du programme pendant l'exécution. On pourra alors suivre exactement le déroulement du programme.