

Sur les traces de Frankenstein

par André C.

Etat de la question.

Je ne suis pas un spécialiste du développement croisé, mais je sais qu'il existe différents assembleurs en concurrence. Mon propos n'est pas de prendre parti pour tel environnement de développement plutôt que pour un autre. Le problème est tout autre. J'ai entre les mains deux fichiers sources pour l'assembleur Frankenstein. Il me faut apporter des modifications à ces sources pour produire un nouveau code adapté à mes besoins. Et comme ces fichiers sont énormes, je n'ai aucune envie de les modifier pour pouvoir les utiliser avec un autre assembleur. Donc, il ne me reste plus qu'à utiliser l'Assembleur Frankenstein. Voyons voir ça.

Le Kit Frankenstein «minimum»

L'assembleur Frankenstein est une usine à gaz, conçu par Mark Zenier pour une vingtaine de microprocesseurs. Nous nous en tiendrons à la version 6502. Ce kit «minimum» comporte:

- Fasm.exe, l'assembleur proprement dit, qui gère des sorties au format HEX.
- Hex2oric.exe, un utilitaire qui convertit du format HEX au format .tap Oric.
- Deux fichiers .doc pour Fasm, que j'ai converti à partir des documents originaux écrits en Runoff.
- Le fichier source basic11.asm de Fabrice F. (pour servir de modèle de syntaxe, il s'agit du source de la Rom v1.1 de l'Atmos).

```

9801 A0 00    LDY #00    ; Mettre la valeur zéro dans le registre Y
9803 B9 0F 98 LDA 980F,Y ; Lire l'octet présent dans la mémoire d'adresse 980F+Y
9806 F0 06    BEQ 980E   ; Si la valeur lue = zéro, c'est fini, termine en 980E
9808 99 82 BB STA BB82,Y ; Sinon on copie l'octet en BB82+Y (ligne service)
980B C8        INY Y=Y+1 ; Pour indexer l'octet suivant
980C D0 F5    BNE 9803  ; On reboucle en 9814 tant que Y ne repasse pas à zéro
980E 60        RT          ; Fin du programme on retourne au point d'appel
980F 53 61 6C 75 74 20 6C 65 73 20 67 61 72 73 20 21 00 ; «Salut les gars !»

```

Il faut revoir ça pour le mettre dans une syntaxe compréhensible pour un assembleur, ici pour

-Les divers fichiers produits par la démo d'aujourd'hui.

Ce kit Frankenstein.zip sera sur la prochaine disquette trimestrielle. Il vous suffira de le décompresser dans un répertoire «Frankenstein», situé où vous voulez.

Ce qu'il vous faut

Pas grand chose de plus que le kit Frankenstein. En fait même rien, si vous utilisez le bloc note de Windows pour élaborer les fichiers asm et bat. Mais je vous recommande quand même d'utiliser plutôt un Notepad un peu plus convivial. Regardez par exemple sur <www.notetab.com>, vous y trouverez la version light de NoteTab, excellent programme gratuit <http://www.fookes.com/ftp/free/NoteTab_Setup.exe>. Mais il y en a beaucoup d'autres, par exemple l'excellent Crimson Editor.

Facultativement, si vous voulez examiner les fichiers produits et/ou retirer l'entête du fichier tap pour récupérer uniquement le code, vous aurez besoin d'un éditeur hexadécimal. Là aussi, il en existe des tas: WinHex, Hex Editor, Hex Workshop, etc.

Le cahier des charges pour faire un petit essai

Pour nous faire la main, nous n'allons pas compiler un gros truc, mais juste une petite démo-test. Il s'agit simplement d'afficher un message sur la ligne service.

Voici la routine source qui servira de test, exprimée dans notre style habituel:

```

SERVICE equ $BB82    ; ligne service
          org $9801    ; le programme sera implanté en $9801
          ldy #$00     ; mettre la valeur zéro dans le registre Y
BOUCLE   lda MESSAGE,Y ; lire l'octet présent dans la mémoire d'adresse 980F+Y
          beq FIN      ; si la valeur lue est zéro, c'est fini on termine en 980E
          sta SERVICE,Y ; sinon on copie l'octet en BB82+Y (dans ligne service)
          iny          ; Y=Y+1 pour indexer l'octet suivant
          bne BOUCLE   ; on reboucle en 9814 tant que Y ne repasse pas à zéro
FIN       RTS          ; fin du programme on retourne au point d'appel
MESSAGE   string «Salut les gars !»
          db $00

```

Comme il faut économiser son énergie (c'est à la mode), nous allons construire un petit fichier batch: lanceur.bat, qui va assurer les lancements

```
bin\frasm -l salut.lst salut.asm >nul
if errorlevel 1 goto end
bin\frasm -o salut.hex salut.asm
bin\hex2oric -o salut.tap salut.hex
```

Commentaire: La première invocation de Frasm teste s'il y a des erreurs dans le fichier source et produira salut.lst, témoin du travail de l'assembleur. La seconde invocation de Frasm.exe produira le fichier salut.hex, qui est une suite d'octets complètement abscons. Enfin, l'invocation de Hex2oric.exe produira notre fichier cassette. Placer ce fichier Lanceur.bat dans le répertoire

```
16161624000080C7981F98010000
A000
B90F98
F006
9982BB
C8
D0F5
60
53616C7574206C65732067617273202100
```

en séquence de Frasm.exe et de Hex2oric.exe, avec les bons paramètres, soit:

«Frankenstein» et double cliquez dessus. Avant que vous ayez le temps de ciller, plof! il y a trois nouveaux fichiers dans le répertoire «Frankenstein»:

salut.lst (ouf! pas d'erreur, ni d'avertissement)
salut.hex (la suite de d'octets abscons)
salut.tap (se lance sans problème sous Euphoric).
Voici la structure du fichier salut.bat produit:

```
entête sans nom de fichier, Auto, 9801-981F
Lda #$00
LDA 980F,Y
BEQ 980E
STA BB82,Y
INY
BNE 9803
RTS
«Salut les gars !»
```

Admirez la recopie d'écran ci-contre, avec son superbe message sur la ligne service! Séduit? Allez, faites-vous un petit plaisir, assemblez donc le fichier basic11.asm de Fabrice! Il suffit de remplacer «salut» par «basic11» dans le fichier Lanceur.bat, de double cliquer dessus pour obtenir une belle Rom Atmos! Enfin presque, car il faudra retirer l'entête du fichier tap (ici les 14 premiers octets) pour obtenir la Rom proprement dite, qui est identique à celle de votre Atmos (fichier Basic11B.rom de Euphoric). Et dire que ça n'a pris que le temps d'un clin d'oeil! Bon trêve d'enthousiasme, on cherche un volontaire pour se mettre au travail. A partir de basic11.asm, il y a pas mal de possibilité de produire une BASIC11C. ROM corrigeant les bugs de l'Atmos...

