

Le compilateur Basic de Ray MacLaughlin (4)

Le cas des commandes Sedoric (suite de la fin)

par André C.

Dans les articles précédents, j'ai montré comment contourner le rejet des commandes Sedoric par le Compiler de Ray : Puisque ce dernier gère sans problème la commande Basic "CALL", il suffit d'utiliser des bouts de code en langage machine. Le rôle de ces sous-programmes est simple : Mettre en place en Ram Overlay les paramètres de la commande Sedoric et appeler cette dernière.

C'est ainsi que j'ai successivement montré comment :

1. Aller sur la Ram Overlay (exemple : lire la variable "Version de la Rom") et revenir sur la Rom.
2. Utiliser une commande Sedoric sans argument (exemple : la commande "SYS").
3. Utiliser une commande Sedoric avec paramètres (exemple : la commande "LOAD").

En principe, vous avez toutes les clefs en main pour exécuter n'importe quelle commande Sedoric, d'autant que ces commandes sont décrites, ainsi que l'emplacement de leurs paramètres dans le livre "Sedoric 3,0 à nu".

Toutefois, lors de l'adaptation de PacGirl (voir l'article "Conversion de l'Hyper-Basic vers le Basic

1.0/1.1 (2) : De Pac-Girl pour Telestrat à Pac-Girl 2018 pour Atmos"), j'ai été confronté au cas de la commande SAVE et je me suis rendu compte que ce n'est pas aussi simple que pour la commande LOAD.

En effet, le cœur de la commande SAVE est utilisé, selon les paramètres, par un large éventail de sous-commandes : SAVE, SAVEO, SAVEU, SAVEM, ESAVE, KEYSAVE et CREATEW), qui comportent une multitude de syntaxes et de paramètres.

En examinant la syntaxe d'une sous-commande donnée, on peut déduire quels sont les paramètres à mettre à jour avant d'appeler la commande SAVE proprement dite.

Par exemple pour SAVEO, il faut indiquer que le fichier à sauver doit écraser (le "O" signifie "Over") tout fichier du même nom s'il existe déjà (sinon on est ramené à un simple SAVE). La lecture de "Sedoric 3,0 à nu", page 252, révèle que cela se fait tout simplement en plaçant un #00 dans la variable VSALO0. De proche en proche on peut installer tous les paramètres requis dans la Ram Overlay.

Localisation et valeurs des paramètres pour LOAD/SAVE dans le Ram Overlay

Heureusement les paramètres sont groupés en page #C0 (la première page de la Ram Overlay) :

#C028	DRIVE	n° de drive pour LOAD/SAVE (#00 par défaut pour le drive A).
#C029-#C031	BUFNOM	Le nom (complété à 9 caractères avec des espaces).
#C032-#C034	BUFNOM'	L'extension (complétée à 3 caractères avec des espaces).
#C035-#C036	PSDESP	Les coordonnées Piste/Secteur du DEScripteur Principal*.
#C037-#C038	NSTOTP	Le nombre de Secteurs TOTaux + Prot/unprot*.
#C04D	VSALO0	Pour LOAD : #40 si ",V" ou #80 si ",N" (valeur par défaut #00). Pour SAVE : #00 si SAVEO, #40 si SAVEM, #80 si SAVE et #C0 si SAVEU (valeur par défaut #00).
#C04E	VSALO1	Pour LOAD : #40 si ",A" ou #80 si ",J" (valeur par défaut #00).
#C04F-#C050	LGSALO	Longueur du fichier en octets (FISALO - DESALO).
#C051	FTYPE	#40 bloc de données, #41 idem auto, #80 Basic et #81 Basic auto*.
#C052-#C053	DESALO	Adresse de début du fichier (celle du premier octet).
#C054-#C055	FISALO	Adresse de fin du fichier (celle du dernier octet).
#C056-#C057	EXSALO	Adresse d'exécution du fichier s'il est en AUTO.

*Notes :

- 1) PSDESP et NSTOTP sont automatiquement mis à jour et il n'y a pas besoin de s'en occuper.
- 2) Pour FTYPE, seuls les cas les plus courants sont indiqués ici. Pour plus de précision, voir le manuel Sedoric page 100). Notez que lors de l'analyse de syntaxe normale, FTYPE est automatiquement mis à jour en fonction de la présence ou pas d'adresses de début et de fin et de la présence ou pas des paramètres ",T" ou ",AUTO". Mais dans notre cas, il faudra initialiser correctement FTYPE.

Un petit exemple

Pour illustrer le processus, voici ce qu'il faut faire pour exécuter la commande suivante :
SAVEO"nnnnnnnnn.eee",A#BB80,E#BFFF.

Outre le #00 à placer dans VSALO0 (en #C04D) pour "Over", il faut initialiser le nom du fichier dans BUFNOM (de #C029 à #C034), l'adresse de début #BB80 dans DESALO (#80 en #C052 et #BB en #C053) et l'adresse de fin #BFFF dans FISALO (#FF en #C054 et #BF en #C055) et mettre #40 dans FTYPE (en #C051).

C'est tout !

Une application réelle

Vous trouverez à la page suivante le détail du programme en langage machine que j'ai utilisé pour adapter Pac-Girl de l'Hyper-Basic au Basic de l'Atmos.

Même si vous n'êtes pas un crac en langage assembleur, vous pourrez repérer les 3 sous-programmes suivant :

- SP1 qui réalise le LOAD"PACGIRL.CHS" et les LOAD"PACGIRLx.SCR" (la chaîne de caractères est éditée par le programme Basic au cours de l'exécution de celui-ci, par

exemple x est édité de 0 à 6 par un POKE approprié).

- SP2 qui exécute la commande LOAD"PACGIRLSV.SCR" (restauration de la partie précédemment sauvee).
- SP3 qui exécute la commande SAVEO"PACGIRLSV.SCR",A#BB80,E#BFFF (sauvegarde de la partie en cours).

Je pense que le détail de ces 3 sous-programmes est facilement compréhensible avec les explications qui précèdent et avec le commentaire de chaque ligne du fichier source.

Puisque le fichier de sauvegarde PACGIRLSV.SCR a vocation à être écrasé à maintes reprises, vous remarquerez que c'est la commande SAVEO qui est utilisée. Il aurait été possible d'utiliser SAVEU (qui conserve le fichier déjà existant avec l'extension .BAK). Dans ce cas, il aurait fallu mettre #C0 dans VSALO0 au lieu de #00.

Le fichier source ci-dessus est présent dans le fichier Compiler-4.zip qui accompagne ce numéro du CEO-mag. Voici les adresses des labels les plus intéressants :

#9801	SP1	CALL#9801 exécutera un LOAD"chaîne1".
#981D	chaîne1	Dans le code assemblé, cette chaîne vaut initialement "PACGIRL CHS", mais elle peut être éditée afin de pouvoir charger d'autres fichiers. Il suffit de POKer d'autres caractères entre #981D et #9818 (ne pas toucher au #00 se trouvant en #9819).
982A	SP2	CALL#982A exécutera un LOAD"chaîne2".
#9846	chaîne2	Dans le code assemblé, cette chaîne vaut "PACGIRLSVCHS". On aurait pu utiliser SP1 en éditant la chaîne 1, mais au cours du jeu Pac-Girl, on est amené soit à charger les labyrinthes qui suivent (l'édition de la chaîne 1 se limite alors à celle du numéro de labyrinthe en #9824), soit le fichier de sauvegarde de la partie. Il était plus simple d'avoir 2 chaînes distinctes.
#9853	SP3	CALL#9853 exécutera un SAVEO"chaîne2". Dans le code assemblé, cette chaîne vaut "PACGIRLSVCHS", comme indiqué ci-dessus. Ce sous-programme restaure la partie précédemment sauvegardée.
#9896	fin	L'octet en #9896 ne fait plus partie du code qui va donc de #9801 à #9895.

Vous pouvez adapter ce fichier source à vos propres besoins, éventuellement à l'aide du livre "Sedoric 3,0 à nu". Ce livre est téléchargeable sur le site du CEO : https://www.oric.org/ftp/ceo/ceo_mag/downloads/ebooks/sedoric3anu3ed.pdf
Il s'agit d'une version PDF au format "texte" qui

peut être utilisée pour effectuer des recherches sur des mots, ce qui est très pratique, car avec 630 pages, il est difficile de trouver ce dont on a besoin ! Par exemple, on peut chercher toutes les occurrences de "FISALO".

Bon amusement avec votre Oric !

```

; Sous-programme SAVEO/LOAD en langage machine
; pour execution avec un CALL à partir d'un programme Basic
org $9801          ; Adresse d'implantation du sous-programme
SP1                ; Charge PACGIRL.CHS et PACGIRLx.SCR
JSR $04F2          ; Passage sur la Ram overlay
LDA $C009          ; Lit le numero de drive par default
STA $C028          ; et le copie devant BUFNOM
LDX #$00           ; Initialise l'index de lecture/ecriture
boucle1
LDA chaine1,X     ; Lit les 12 octets de la chaine
BEQ suite1        ; Si l'octet lu est #$00, c'est fini
STA $C029,X       ; Sinon le copie dans BUFNOM
INX                ; Vise l'octet suivant
BNE boucle1       ; et reboucle
suite1
JSR $E052         ; Execution de la commande Sedoric LOAD
JMP $04F2         ; Retour sur la Rom
chaine1
string "PACGIRL  CHS"
db $00
SP2                ; Charge PACGIRLSV.SCR (restaure la partie)
JSR $04F2          ; Passage sur la Ram overlay
LDA $C009          ; Lit le n° de drive par default
STA $C028          ; et le copie devant BUFNOM
LDX #$00           ; Initialise l'index de lecture/ecriture
boucle2
LDA chaine2,X     ; Lit les 12 octets de la chaine
BEQ suite2        ; Si l'octet lu est #$00, c'est fini
STA $C029,X       ; Sinon le copie dans BUFNOM
INX                ; Vise l'octet suivant
BNE boucle2       ; et reboucle
suite2
JSR $E052         ; Execution de la commande Sedoric LOAD
JMP $04F2         ; Retour sur la Rom
chaine2
string "PACGIRLSVSCR"
db $00
SP3                ; Sauve PACGIRLSV.SCR (sauve la partie)
JSR $04F2          ; Passage sur la Ram overlay
LDA $C009          ; Lit le numero de drive par default
STA $C028          ; et le copie devant BUFNOM
LDX #$00           ; Initialise l'index de lecture/ecriture
boucle3
LDA chaine2,X     ; Lit les 12 octets de la chaine
BEQ suite3        ; Si l'octet lu est #$00, c'est fini
STA $C029,X       ; Sinon le copie dans BUFNOM
INX                ; Vise l'octet suivant
BNE boucle3       ; et reboucle
suite3
LDA #$00
STA $C04D          ; VSALO0 type de sauvegarde
STA $C04E          ; VSALO1 type de sauvegarde
STA $C056          ; EXSALO LL adresse execution
STA $C057          ; EXSALO HH adresse execution
LDA #$40
STA $C051          ; FTYPE, ici bloc de donnees
LDA #$80
STA $C052          ; LL adresse de debut DESALO
LDA #$BB
STA $C053          ; HH adresse de debut soit $BB80
LDA #$FF
STA $C054          ; LL adresse de fin FISALO
LDA #$BF
STA $C055          ; HH adresse de fin soit $BFFF
JSR $DE0B         ; Execution de la commande Sedoric SAVEO
JMP $04F2         ; Retour sur la Rom
fin

```