

NDLR : cet article paraîtra en plusieurs fois.

Ce mois ci nous allons aborder la routine RAMOV qui correspond à l'interpréteur du Sédoric. Cette routine est située en #D3AE- #D44E, dans la RAM Overlay de l'Oric. Elle est appelée à chaque entrée d'instruction dans le buffer, qui est logé entre #35 et #80 (page zéro), et fait suite à la routine d'amorce de la page 4.

La routine RAMOV fait appel à diverses sous-routines qui sont détaillées dans le compte-rendu que nous vous présentons. Elle nécessite aussi 3 tableaux pour l'analyse des mots-clefs:

- un premier tableau de syntaxe des mots-clefs du SED (Système d'Exploitation du Disque) situé entre #C9DE et #CBB8. Les mots clefs qui comportent un token BASIC sont codés de deux manières différentes. Effectivement, si l'utilisateur a choisi de taper tous les mots du Sédoric en minuscules, pour les distinguer des mots du BASIC, l'encodage ne sera pas réalisé par la ROM en #ECB9 (Atmos) et chaque lettre du mot sera alors significative. On peut d'ailleurs ici faire la remarque suivante: il est conseillé à l'utilisateur de taper son texte en majuscules, car de la sorte les mots sont raccourcis et l'analyse syntaxique ultérieure par RAMOV en est accélérée.

- un second tableau qui fournit dans l'ordre, l'adresse de la syntaxe détaillée du premier mot clef débutant par la première lettre pointée par TXTPTR, le numéro d'ordre du mot clef et le nombre de mots débutants par cette initiale.

- un dernier tableau, très utile, qui donne les adresses d'exécution des mots clefs (adr- 1) du Sédoric.

L'interprétation de l'interpréteur, il fallait le faire, n'a pas été chose aisée. Aussi, si vous souhaitez participer avec nous à l'analyse du Sédoric, nous apprécierions beaucoup votre aide (de quel niveau qu'elle soit!).

Dans un prochain numéro du C.E.O mag, nous vous entretiendrons de la commande LINFUT ou du SAVE/ LOAD, selon l'état d'avancement de nos travaux. Cependant, nous insistons sur un point: si vous avez déjà pu travailler sur le Sédoric ou si vous souhaitez décortiquer une autre commande, nous serions heureux de pouvoir faire figurer votre participation, aussi modeste soit-elle, à cette rubrique!

Bonne lecture à tous et à bientôt dans les colonnes du C.E.O mag!

XCRGET Saisit dans A le caractère suivant (+ min -> MAJ)

D398- 20 3A D3 JSR D33A exécution de la routine 00E2 (CHARGET)
D39B- 4C A1 D3 JMP D3A1 conversion minuscule -> MAJUSCULE

XCRGOT Saisit dans A le caractère courant (+ min -> MAJ)

D39E- 20 42 D3 JSR D342 exécution de la routine 00EB (CHARGOT)
et continue en D3A1 (conversion min -> MAJ)

minMAJ Convertit (si nécessaire) en MAJUSCULE le caractère dans A

D3A1- 08 PHF sauve P, c à d les indic init par 00E2 ou 00EB
D3A2- C9 61 CMP #61 compare à "a"
D3A4- 90 06 BCC D3AC inutile de continuer si lettre < "a"
D3A6- C9 7B CMP #7B compare à "e" (lettre qui suit "z")
D3A8- B0 02 BCS D3AC inutile de continuer si lettre >= "e"
(les lettres accentuées ne sont pas converties)
D3AA- 29 DF AND #DF si "a" <= lettre <= "z" effectue un ET logique
avec masque 1101 1111, c'est à dire conserve tous les bits sauf le
b5 qui est remis à zéro. Exemple: "b"=#62 = 0110 0010
ET 1101 1111
= 0100 0010 = #42 = "B"

D3AC- 28 PFL récupère P
D3AD- 60 RTS et retourne

ENTREE SEDORIC: RECHERCHE ADRESSE EXECUTION MOT-CLE SEDORIC

Résumé: Lecture du caractère présent au TXTPTR (sans incrém de TXTPTR)
Conversion éventuelle en maj si lettre comprise entre "a" et "z"
Conversion en valeur de #00 (A) à #19 (Z) ou #1A (autre caractère)
Calcul index coordonnées dans sous-table selon 1ère lettre mot-clé
Lect de: adr 1er mot-clé même initiale ds table principale (18/19)
nombre de mots-clés ayant la même initiale (mis en F2),
numéro d'ordre du mot-clé Sédoric (mis en X)
Examen correspondant avec l'un des mots-clés ayant la même initiale
si trouve: exécution en D417
si trouve pas: continue en D428

D3AE- A2 00 LDX #00 ;
D3B0- 8E FD 04 STX 04FD ; RAZ de ERROR (numéro de l'erreur Sédoric)
D3B3- BA TSX ; sauve pointeur de pile en SAUVES
D3B4- 8E 23 C0 STX C023 ; pour sortie éventuelle par erreur
D3B7- A5 E9 LDA E9 ;
D3B9- A4 EA LDY EA ;
D3BB- 8D 1F C0 STA C01F ; sauvegarde de TXTPTR en C01F/C020 (SVTPTR)
D3BE- 8C 20 C0 STY C020 ;
D3C1- 20 9E D3 JSR D39E ; XCRGOT Saisit dans A le caractère à TXTPTR
; et conversion min->MAJ
D3C4- E9 41 SBC #41 ; teste si #40<A<#5C (A contient-il une lettre)
D3C6- A0 1A LDY #1A ; #1A = val finale si car dans A pas une lettre
D3C8- 90 0B BCC D3D2 ; A<#41 pas 1 lettre, donc A=#1A par défaut
NB: BCC D3D2 teste C résultant de SBC#41 (C n'est pas affecté par LDY#1A)
D3CA- C9 1A CMP #1A ; on compare en réalité à #41+#1A = #5B
D3CC- B0 04 BCS D3D2 ; A>=#5B pas 1 lettre, donc A=#1A par défaut
D3CE- A8 TAY ; #00 <= Y <= #19 pour un caractère de A à Z
D3CF- 20 3A D3 JSR D33A ; Incrémenter TXTPTR
D3D2- 98 TYA ; A de #00 (A) à #19 (Z) ou #1A (autre caract)
D3D3- 0A ASL ; calcule index Y en multipliant A par 4 car
D3D4- 0A ASL ; on va indexer avec Y une sous-table en CBBB
D3D5- A8 TAY ; constituée de groupes de 4 octets (qu'il faut
; donc passer pour une valeur donnée de A)
D3D6- B9 BB CB LDA CBBB,Y ; lit 2 premiers oct (adr dans table principale
D3D9- 85 18 STA 18 ; (C9DE-CBBA) où commencent mots-clés de même
D3DB- B9 BC CB LDA CBBC,Y ; initiale) et place ces 2 octets en 18/19
D3DE- 85 19 STA 19 ; (travail encodage/décodage mots clés)

à suivre...