Conversion de programmmes Sédoric V1.0 -> V2.x

Pour doubler la capacité des disquettes Sédoric, Ray McLaughlin a dû sacrifier la "Table des vecteurs système", située à l'origine de #FF43 à #FFF9. Ceci entraîne une perte de compatibilité pour les programmes qui utilisaient cette table. Nous allons faire le tour de cette question et tout d'abord examinons le cas des utilitaires livrés avec Sédoric:

GAMEINIT, DEMO.COM, ROMORIC1.COM et ROMATMOS.COM sont inchangés. En ce qui concerne GAMEINIT, il n'est donc pas possible de formater des disquettes "Games" de plus de 1919 secteurs. Ceci n'a guère d'importance, puisqu'avec les nouvelles disquettes de Ray, on n'est pas à court de place.

Tous les autres utilitaires ont été modifiés. C'est le cas de ADRESSE.* qui n'a pas seulement changé de nom (ADDRESS.*), mais aussi de taille. Il en est de même pour ALPHA.COM, STAT.COM et VERSION.COM. CONVERT a été traduit en anglais et modifié; il a toutefois gardé la même taille. Le jeu MARC.COM a été remplacée par KRILLYS (4 fichiers), les 2 fichiers SECLIBRE.* par 3 fichiers SECMAP.* et HELLO.COM par MENU.COM. Ont été ajoutés : NIBFIX.COM permettant de modifier l' éditeur de disquettes NIBBLE.COM et 3 fichiers V20.* qui décrivent les modifications apportées dans la version 2.0.

Incompatibilités:

Si un programme tourne avec la version 1.0, mais pas avec la version 2.x, c'est probablement dû à la suppression de la table des vecteurs (suite de JMP servant de points d'entrée fixes, pour faciliter l'utilisation des versions ultérieures de Sédoric). Heureusement, la plupart des programmeurs n'ont pas fait transiter les appels par cette table, mais se sont adressés directement aux routines.

Voici une bouée de sauvetage pour ceux d'entre vous qui aimeraient modifier certains programmes comme Ray l'a fait pour NIBBLE. Dans vos programmes, il suffit de remplacer les appels dans la table par les appels directs correspondants. Exemple: si un de vos programme appelle la très importante routine XRWTS par un JSR FFC4 en 98D8, il suffit de remplacer par JSR CFCD (avec un DOKE#98D9,#CFCD).

Repérages des appels à la "Table des vecteurs système"

Voici un petit programme BASIC qui vous aidera à savoir si le programme défectueux fait appel à la "Table des vecteurs système" et à quel endroit. Soyez patient, il est très lent. Le programme affiche l'adresse où il trouve des JSR ou JMP#FFxx ainsi que les 3 octets concernés. Remplacez chaque FFxx par la valeur correspondante indiquée ci-dessous dans la "Table des vecteurs système".

```
110 CLS:INPUT"Nom du fichier";F$
120 LOAD F$,N:LOAD F$,V
130 FOR I=ST TO ED
140 X=PEEK(I)
150 IF X=#20 OR X=#4C OR X=#6C THEN 160 ELSE 180
160 Y=PEEK(I+1):Z=PEEK(I+2)
170 IF Z=#FF THEN PRINT HEX$(I),HEX$(X)HEX$(Z)HEX(Y)
180 NEXT I
```

"Table des vecteurs système"

FF43 JMP **ED36** **XLINPU** appel à la routine LINPUT. TRAV0 contient la longueur de la chaîne à saisir, au retour TRAV2 contient le mode de sortie et #D0-#D1 l'adresse de début de la chaîne

FF46 JMP **D398** **XCRGET** lit le caractère suivant à TXTPTR et le convertit en MAJUSCULE (identique à CHRGET du BASIC, plus convertion des minuscules en MAJUSCULES)

FF49 JMP **D39E** **XCRGOT** relit le caractère à TXTPTR et le convertit en MAJUSCULE (identique au CHRGOT du BASIC, convertion des minuscules en MAJUSCULES)

FF4C JMP **D44F** **XNF** lit nom de fichier non-ambigu à TXTPTR et l'écrit dans BUFNOM

FF4F JMP **D451** **XNFA** lit un nom de fichier ambigu à TXTPTR et l'écrit dans BUFNOM

FF52 JMP **D364** **XAFSC** affiche le X+1ème message externe terminé par "caractère + 128" EXTMS doit contenir l'adresse - 1 du premier message

FF55 JMP **F3F3** vérifie l'existence du "pseudo-tableau" **FI** au début des tableaux et le crée si besoin

FF58 JMP **F4A8** place adr début "Channel Buffer" en 00/01, adr début du "Channel's own Data Buffer"

en 02/03, adr "Descriptor Buffer" en 04/05, adr début du "General Buffer" en 06/07 et met à jour C083 (long. fiche ou #00) et 0B (flag "**S**/**R**/**D**") retourne avec Y = #00

FF5B JMP **FDD9** si pas fin du fichier, copie un enregistrement complet du "Channel's own Data Buffer" (charge un second secteur dans le buffer si nécessaire), vers le "General Buffer", 06/07 pointe sur le type, la longueur et la valeur du data

FF5E JMP **FE38** écrit l'enregistrement du "General Buffer" dans le "Channel's own Data Buffer" en utilisant le secteur suivant si nécessaire

FF61 JMP **FD46** sauve sur la disquette le secteur du fichier qui est présent dans le "General Buffer"

FF64 JMP **D62A** **XAFCAR** affiche le caractère ASCII contenu dans A

FF67 JMP **D613** **XAFHEX** affiche en hexadécimal le contenu de A

FF6A MP **D637** **XAFSTR** affiche la chaîne terminée par un 0 et dont l'adresse est donnée par AY

FF6D JMP **D5D8** **XROM** exécute à partir de la RAM une routine ROM. Le JSR XROM doit être suivi dans l'ordre de l'adresse de la routine pour la V1.0, puis de l'adresse pour la V1.1

FF70 JMP **E0EA** **charge fichier** X = POSNMX, POSNMP, POSNMS, VSALO0, VSALO1, DESALO

FF73 JMP **E0E5** **XLOADA** charge le programme selon BUFNOM, VSALO0, VSALO1, DESALO

FF76 JMP **DE28** **XDEFSA** positionne les valeurs par défaut pour XSAVEB (pour programme BASIC)

FF79 JMP **DFE6** **XDEFLO** positionne les valeurs par défaut pour XLOADA

FF7C JMP **DE9C** **XSAVEB** sauve fichier BUFNOM, VSALO0 VSALO1, DESALO, FISALO, EXSALO

FF7F JMP **E266** **XNOMDE** détruit le fichier indexé par POSNMX, dont le secteur de catalogue est dans BUF3 (en fait, tout est positionné comme après un XTVCAT)

FF82 JMP **DD2D** **XCREAY** crée une table piste secteur de AY secteurs, en fait marque dans la bitmap en BUF2 que le secteur AY est occupé

FF85- JMP **DD15** **XDETSE** libère le secteur Y, piste A sur le bitmap courant

FF88- JMP **DC6C** **XLIBSE** cherche un secteur libre dans la bitmap dans BUF2, retourne avec A = n° de piste et Y = n° de secteur (sinon "DISK FULL ERROR")

FF8B JMP **DBC0** **écriture descripteur** du fichier à sauver. Revient avec nombre secteurs à sauver dans NSSAV (C05A/5B), coordonnées du 1er secteur descripteur dans PSDESC (C05C/5D), le nombre de descripteurs utilisés dans NSDESC (C05E) et 1er descripteur en place

FF8E JMP **DB59** **XTRVCA** cherche une place libre dans le catalogue. A la sortie, POSNMX, POSNMP et POSNMS indiquent la position de la place réservée

FF91 JMP **DBA5** **cherche** POSNMX de **1ère place libre** dans le directory

FF94 JMP **DB41** **ajuste POSNMX sur entrée suivante** du catalogue et reprend la recherche dans le catalogue du fichier indiqué dans BUFNOM (Z = 1 si fini)

FF97 JMP **DB30** **XTVNM** cherche nom contenu dans BUFNOM. A la sortie, POSNMX, POSNMP et POSNMS contiennent position du nom dans catalogue et Z = 1 si fichier pas trouvé

FF9A JMP **DB2D** vérifie disquette en place est disquette Sédoric, cherche fichier dans le catalogue, revient avec secteur de catalogue en place (coordonnées POSNMP et POSNMS) et avec X = POSNMX, pointant sur "l'entrée" cherchée ou avec Z = 1 si fichier pas été trouvé

FF9D JMP **DB07** **XCABU** transfère dans BUFNOM le nom de fichier contenu dans le secteur de catalogue placé dans BUF3, à la position POSNMX

FFA0 JMP **DAFE** **ROUTINE** DOS à identifier

FFA3 JMP **DAEE** **XBUCA** transfère le nom de fichier contenu dans BUFNOM dans le secteur de catalogue contenu dans BUF3, à la position POSNMX

FFA6 JMP **DACE** **XVBUF1** rempli de 0 le BUFFER1

FFA9 JMP **DAA4** **XSVSEC** écrit un secteur selon DRIVE, PISTE, SECTEUR et RWBUF

FFAC JMP **DA9E** **XSAY** sauve le secteur visé par RWBUF au secteur Y de la piste A

FFAF JMP **DA91** **XSBUF1** sauve BUF1 à la piste A et le secteur Y

FFB2 JMP **DA82** **XSCAT** sauve secteur de catalogue contenu dans BUF3, selon POSNMP et POSNMS

FFB5 JMP **DA8A** **XSMAP** sauve le secteur de bitmap sur la disquette

FFB8 JMP **DA73** **XPRSEC** lit un secteur selon DRIVE, PISTE, SECTEUR et RWBUF

FFBB JMP **DA6D** **XPAY** charge dans RWBUF le secteur Y de piste A

FFBE JMP **DA5D** **XPBUF1** charge dans BUF1 le secteur Y de piste A

FFC1 JMP **DA4C** **XPMAP** prend secteur de bitmap dans BUF2, vérifie le format

FFC4 JMP **CFCD** **XRWTS** gestion lecteurs. X = commande. En sortie, Z = 1 si pas d'erreur. V = 1 si disquette prot écrit. DRIVE, PISTE, SECTEUR, et RWBUF doivent être à jour.

André Chéramy, 54 rue de Sours, 28000 CHARTRES