

LA GESTION DES FICHES SOUS SEDORIC

par Yann Legrand et André Chéramy
- Basé sur un article de Ray MacLaughlin -

Depuis l'apparition du Sédoric, la gestion des fiches a été considérablement allégée et simplifiée pour les utilisateurs de l'Oric. Bon nombre d'auteurs ont ainsi pu mettre à profit des commandes faciles à manier pour créer des programmes de gestion de logithèque, de carnet d'adresses etc...

Dans cet article, nous allons examiner le fonctionnement intrinsèque des commandes Sédoric relatives aux fichiers à accès direct dits de type **R** (pour **R**andom). Ces commandes permettent de concevoir des programmes dignes des ordinateurs les plus performants...

En fait, pour tous les fichiers créés par la commande OPEN (qu'ils soient de type **R**, **S** ou **D**), Sédoric génère automatiquement une zone spéciale que Ray appelle "File Array" et qui se comporte en fait comme un "Pseudo Tableau" BASIC de type entier et de nom **FI**. Il est à noter que le manuel Sédoric ne souffle mot de ce "tableau" **FI** et que si par malheur vous aviez l'idée d'en créer un pour votre propre usage, le système plantera. De même, à partir du moment où un OPEN a été utilisé et ce jusqu'au dernier CLOSE, il est absolument interdit d'utiliser la commande BASIC DIM! (ce n'est pas non plus dans le manuel).

Dès que le premier n logique (NL) est attribué à un fichier par la commande OPEN (voir page 76 du manuel pour plus d'informations), le "Pseudo Tableau" **FI** est créé au début de la zone des tableaux BASIC (dont l'adresse est indiquée en 9E/9F) et a une longueur initiale totale de #288 octets. Au fur et à mesure de l'ouverture des fichiers et donc de l'attribution d'autres NL, **FI** est étendu par construction d'un buffer (tampon), le "Channel Buffer" pour chaque fichier (donc chaque NL). De plus lorsque la commande FIELD est utilisée pour la 1ère fois, un buffer spécial, le "Field Buffer" (tampon relatif aux champs) est initialisé juste à la fin de **FI** afin de prévoir le stockage des informations en provenance de la dite commande FIELD. Ce "Field Buffer" sera étendu au fur et à mesure des besoins afin de recueillir davantage d'informations. Toutes ces créations (**FI**, "Channel Buffer" et "Field Buffer") se font par insertions impliquant un décalage des (vrais) tableaux BASIC vers le haut de la RAM.

Passons maintenant à la description de la structure de FI:

1) "Entête" 8 octets de rang #00 à #07:

Les 2 premiers octets ont pour valeur respective #C6 et #C9 qui sont les lettres du nom du tableau (les bits 7 à 1 indiquent le type "entier").

La seconde paire d'octets donne la longueur totale de **FI** (lien utilisé par le BASIC pour trouver le début du tableau suivant). Ces 4 premiers octets constituent la partie usuelle et nécessaire d'un tableau BASIC.

La troisième paire d'octets donne la différence d'adresse (offset) entre le début de **FI** et le début du "Field Buffer" vu précédemment et sur lequel nous aurons l'occasion de revenir plus en détails.

La quatrième paire d'octets donne le nombre total de champs déclarés avec les commandes FIELD pour tous les fichiers ouverts.

2) "Table NL" #80 octets de rang #08 à #87:

Cette table des NL est formée de #40 paires d'octets (une pour chaque NL possible de #00 à #3F soit de 0 à 64). Chaque paire donne l'offset (différence d'adresses) entre le début de **FI** et le début du "Channel Buffer" correspondant à ce NL. Si l'octet de poids fort est à zéro, alors le NL n'a pas été attribué (fichier fermé). Il est donc possible en théorie d'ouvrir simultanément en mémoire 64 fichiers (qu'ils soient de type **R**, **S** ou **D**)...

3) "General Buffer" #200 octets de rang #88 à #287:

Comme son nom l'indique, c'est un tampon d'usage général. Nous ne parlerons que de son utilisation avec les fichiers de type **R** qui seuls nous intéressent ici. Sur la disquette, un fichier **R** donné est une série de fiches de longueur fixe qui sont stockées bout à bout. La plupart des fiches se trouvent donc à cheval sur 2 secteurs. Le 1er secteur est celui qui contient toujours le début de la fiche. Le 2ème secteur contient parfois la fin de la fiche. Ces 2 secteurs (#200 octets) sont lus/écrits dans le "General Buffer". La fiche proprement dite (longueur maximum #FF octets) est "reconstituée" dans le "Channel Buffer" (#100 octets, voir plus loin). Ray utilise (à juste titre) les termes "Data Buffer" et "Record Buffer" que nous avons essayé d'éviter car ils varient

selon le type de fichier (**R**, **S** ou **D**) considéré.

4) "Channel Buffer" #121 octets pour chaque fichier ouvert (qu'ils soient de type **R**, **S** ou **D**). Le rang de ces #121 octets dans **FI** dépend entre autre de leur ordre d'ouverture. Pour s'y retrouver, on dispose de la "Table NL" qui indique pour chaque NL en usage l'offset du "Channel Buffer" correspondant. Cependant ces #121 octets, que nous allons repérer par leur rang dans le "Channel Buffer", ont un rôle particulier. Afin de ne pas allonger inutilement cet article, nous nous en tiendrons aux fichiers de type **R**:

Rang: Usage:

- #00 Type de fichier ouvert, ici #00 (rappelé en 0B pour le fichier courant)
- #01 Longueur d'une fiche (rappelé en C083 en RAMOV pour le fichier courant)
- #02 n du dernier descripteur du fichier ouvert (le 1er porte le n #00)
- #03 Index dans le descripteur courant situé dans le "Descriptor Buffer", vise les coordonnées piste/secteur d'un des secteurs du fichier ouvert
- #04 n du descripteur courant dans le "Descriptor Buffer"
- #05 Index dans le "General Buffer" lu ou à écrire sur la disquette (offset entre début du "General Buffer" et début de la fiche)
- #06 n du lecteur sur lequel le fichier a été ouvert (le NL correspondant au fichier courant est indiqué en 0A)
- #07 à #16 copie de la ligne de directory correspondant au fichier ouvert
- #17 à #116 ces #100 octets constituent le "Channel's own Data Buffer" (tampon propre au NL), utilisé pour mettre une fiche donnée (le "General Buffer" sert d'intermédiaire entre la disquette et le "Channel's own Data Buffer" où les data propres à une fiche donnée sont assemblés)
- #117 à #120 ces 10 octets ne sont pas utilisés mais repoussés vers le haut lors de la création au point de rang #117 du "Descriptor Buffer". Dans ce cas, les octets #117 à #216 contiennent le premier descripteur du fichier, les octets #217 à #316 contiennent le descripteur suivant s'il existe etc... La micro zone située originalement de #117 à #120, ainsi que les autres "Channel Buffers", le "Field Buffer et les tableaux BASIC s'ils existent sont ainsi repoussés vers le haut pour charger tous les descripteurs du fichier.

5) "Field Buffer":

A la première utilisation de la commande FIELD, un tampon de 100 octets est généré à la fin de **FI** pour 10 entrées de 10 octets. Le "Field Buffer" sert à garder les informations fournies par les commandes FIELD. Il est étendu si nécessaire (par multiples de 10 entrées) pour recevoir toutes les informations supplémentaires fournies par d'autres commandes FIELD.

Les 10 octets d'une entrée, correspondant à la description d'un champ, ont les significations suivantes:

Rang: Usage:

- #00 à #04 nom du champ 5 caractères significatifs, les suivants sont négligés
- #05 index du champ (uniquement pour les éléments de pseudo tableau)
- #06 NL (fichier) pour lequel ce champ a été déclaré
- #07 offset séparant le début de la fiche du début de ce champ
- #08 longueur du champ en octets (1 si champ octet **O**, 2 si champ entier **%**, 5 si champ réel **!** ou longueur réelle pour un champ alphanumérique **\$**)
- #09 type de champ (#00 pour **!**, #01 pour **%**, #40 pour **O** et #80 pour **\$**)

Pointeurs utilisés pour les buffers

Les octets 00/07 en page zéro de la RAM sont utilisés comme suit:

- 00/01 adresse réelle du début du "Channel Buffer" du fichier courant
- 02/03 idem + #17 = adresse du début du "Channel's own Data Buffer" (#100 octets) qui est utilisé comme "Fiche Buffer" pour un fichier **"R"**
- 04/05 idem + #117 = adresse du début du "Descriptor Buffer"
- 06/07 adresse du début du "Buffer Général", c'est à dire de la zone tampon vraie, utilisée pour les échanges directs avec la disquette. Cette adresse qui est mise à jour sans arrêt est la seule qui ne change pas!

Nous vous donnons rendez-vous pour la suite dans un prochain article.