

LES TRANSFERTS L.S.E. PAR LA VOIE V24

R. BARRAU

C'est pour des équipes de recherche INRP dirigées par P. Muller et s'occupant d'EAO en latin et en français que j'ai été amené à entreprendre sur cette question un gros travail qui m'a demandé, tout compris, des dizaines et peut-être même plusieurs centaines d'heures. Comme ces transferts intéressent actuellement beaucoup de collègues, j'ai demandé à l'EPI de bien vouloir publier les résultats auxquels je suis parvenu.

Le problème

Les didacticiels de lettres, on le sait, se composent de gros programmes et d'énormes fichiers de données. Un grand nombre de ces didacticiels existent actuellement sur micros "8 bits" qui ne figurent pas dans la bibliothèque officielle du CNDP, et qu'il fallait donc pouvoir transférer commodément sans son aide. Car, au moment de l'arrivée des "16 bits", il n'était pas question de retaper au clavier cette masse énorme d'information. Je dirai plus loin un mot des programmes. Venons-en aux fichiers de données, dont la transmission est plus complexe, et sur lesquels a porté l'essentiel de mon travail.

Nous disposions des programmes déjà anciens ECHS et ECHB de J.-J. Equoy, diffusés par le CNDP, et qui ont beaucoup servi pour les transferts entre "8 bits". Or, ces logiciels ne marchent pas bien pour le transfert "8 bits" - "16 bits", du moins quand le récepteur est un Persona , LSE version 4.065, sauf à augmenter considérablement les temporisations (il faut multiplier au moins par 3) avant chaque émission sur la voie V24, ce qui a pour résultat des temps d'exécution prohibitifs.

Mon problème était donc d'abord d'accélérer les transferts. Et il n'était pas question (on me l'a suggéré et même proposé) d'utiliser un binaire. Car, en l'espèce, on passe sans cesse d'un appareil et d'un système à un autre, ce qui aurait obligé à réécrire à chaque fois les P.B.I.

Les moyens

Je ne m'embarquais pas sans biscuit dans cette affaire, Dieu merci ! D'abord, ECHS et ECHB déblaient considérablement le terrain. J'ai conservé leur philosophie d'ensemble, la seule possible, me semble-t-il. Par ailleurs, d'autres collègues, ayant travaillé sur le même sujet, fournissaient ici et là des solutions intéressantes, dont je ne me suis évidemment pas privé. On sait, par exemple, que la fonction NES (numéro d'enregistrement suivant) n'est pas "implémentée" sur les "8 bits", or, ici, elle est indispensable. Supposons, cela existe, un fichier qui a des enregistrements contigus de 1 à 20, puis qui saute de 20 à 32000. Il n'est pas question d'exécuter 31979 fois pour rien l'instruction CHARGER pour être sûr de ne rien oublier ! 2 solutions existaient, que j'ai retenues selon l'appareil émetteur

- sur Micral, un binaire NES de J.C. Chabrilat, qui fait exactement la même chose.
- sur LX 529 et Sil'z 8 bits, le CHARGER à 2ème paramètre nul ou négatif, qui, par des voies différentes, aboutit au même résultat.

La solution

Ce qui restait à revoir, et à revoir entièrement, c'était la procédure d'échange elle-même. Ici, je dois apporter quelques détails techniques :

Quand et comment risque-t-on de perdre des informations en route ? Les échanges sur V24 se font à 9600 bauds. Quand on sait (on me l'a dit récemment) que les transmissions des nanoréseaux se font à 500000 bauds, soit 52 fois plus vite, et que le Persona 1600 est utilisé comme tête de réseau, on comprend que le problème n'est pas d'ordre matériel, mais logiciel (sauf, évidemment, mauvais réglage ou panne de la voie V24, qui relève de la maintenance des appareils !). On perd des informations, c'est-à-dire des caractères, quand l'ordinateur source commence à émettre avant que l'ordinateur récepteur ne soit en lecture, auquel cas ce dernier "ignore" les caractères reçus.

D'où les nombreuses temporisations de ECHS avant affichage. D'où aussi l'idée, excellente et que j'ai conservée, de J.-J. Equoy, de faire envoyer du récepteur à l'émetteur des tops de synchronisation. Dommage seulement qu'il n'ait pas suivi son idée jusqu'au bout, notamment dans la transmission des chaînes de caractères, ce qui nous aurait épargné de nombreuses attentes.

On voit tout de suite que les caractères perdus sont toujours les premiers caractères émis (je laisse à part le cas particulier de certains "8 bits" qui ont un tampon d'entrée de 250 caractères, cas déjà traité d'ailleurs, et fort bien, par J.-J. Equoy). D'où l'idée qu'en réduisant le nombre des affichages quitte à augmenter leur longueur, on réduit d'autant les temporisations nécessaires.

A la limite, en envoyant tout un enregistrement d'un coup, on n'a besoin que d'une seule temporisation par enregistrement. Et d'ailleurs, cette temporisation n'est même pas nécessaire, car, avant d'envoyer un enregistrement, l'émetteur doit le CHARGER à partir de la disquette et le traiter, ce qui laisse largement au récepteur le temps de se mettre en lecture. Mais, dira-t-on, comment envoyer d'un seul trait tous les renseignements concernant un enregistrement ? La réponse est simple : il suffit de tout concaténer dans une seule chaîne de caractères, en séparant par un délimiteur spécial (j'ai choisi EOT, .4.) les divers éléments. Nombres et booléens sont convertis en chaînes avant l'émission, et reconvertis en leur type d'origine après réception.

Sur ma lancée, j'ai voulu augmenter aussi la sécurité des échanges, en m'assurant à la réception, autant que faire se peut, que l'information reçue est correcte. 2 moyens :

- je fais précéder toute chaîne émise sur V24 du caractère XON (.17.), et je vérifie à la réception que ce premier caractère n'a pas été perdu,
- je fais suivre cet XON de la longueur de la chaîne émise et vérifie, à la réception, que le compte est bon.

N.B. Bien entendu, EOT et XON sont ajoutés à la liste des "caractères à problèmes", ceux qui sont transformés en EQN avant émission, et retransformés en EQC après réception (encore une idée excellente de J.-J. Equoy, que j'ai conservée).

Le transfert de programmes

Le problème d'échange est exactement le même que pour les fichiers de données. On perd des caractères, les premiers, lorsque l'émetteur, ayant fini de décompiler la ligne courante, l'envoie sur V24, alors que le récepteur, n'ayant pas achevé la compilation de la ligne précédente, n'est pas encore en lecture (supposons une ligne longue, suivie d'une ligne courte) D'où l'utilisation de la commande ATtendre qui, si je ne m'abuse, ne sert qu'à cela.

Or c'est une mauvaise solution, car on est obligé de demander l'attente maximum (pour éviter un "plantage") laquelle attente sera inutile pour 99% des lignes du programme, ou du moins inutilement importante. Je me permets d'en suggérer une autre aux "implémenteurs" du LSE (j'ai moi-même autrefois bricolé sur les systèmes, mais je m'en abstiens désormais, laissant cela aux spécialistes). C'est celle-là même qui existait à l'époque héroïque où les programmes étaient chargés par une bande perforée lue sur télétype de service. A la fin de chaque ligne de programme, un XOFF arrêta le lecteur de bandes, et un XON le relançait, après compilation de la ligne; par analogie, on pourrait faire émettre un XON après compilation par le récepteur sur V24, l'émetteur attendant cet XON pour envoyer la ligne suivante.

Il me reste, en conclusion, à remercier le CREFIAP (centre de formation lourde) de Montpellier, sans l'aide duquel mon travail n'aurait pas été possible. Je remercie en particulier mes amis R. WALD, qui m'a ouvert très libéralement l'accès aux appareils, L. Manara et R. Piquemal dont les avis m'ont été extrêmement précieux.

R. BARRAU

Les collègues intéressés peuvent s'adresser à : R. Barrau, professeur de lettres classiques - Lycée Joffre Montpellier - 8 Mai 1985