JPC

OCTOBRE 1989

NUMERO 68

Le numéro 40 F

A PROPOS DU CLUB

HP-28

HP-71

EDITORIAL

Chers Amis,

Le duo infernal a encore frappé ! Après FINPUT, l'indispensable STRUC2 et autres irremplacables Lex, Pierre David et Janick Taillandier ont enfin terminé EDIT, sûrement le plus gros Lex réalisé hors des locaux de Hewlett Packard. Nous pouvons tous les féliciter de cet enorme travail. Comme vous le savez, notre devoir étant non seulement de vous permettre d'obtenir ces programmes, mais surtout de vous montrer comment ils sont faits, nous n'avons pas hésité à vous présenter l'intégralité du source. Celui ci faisant plus de 250 pages, cela représente à coup sûr une première dans la presse informatique. Pour des raisons évidentes, nous avons décidé de modifier notre mise en page, en compactant les listings et les notices, ainsi qu'en échelonnant la parution sur deux ou trois mois. Nous espérons ainsi que vous trouverez dans ce source les réponses que vous cherchez pour vous propres Lexs.

Deux nouvelles maintenant. La mauvaise d'abord : la HP-41 cessera d'exister courant 1990. La bonne nouvelle, venant de milieux bien informés, est que son successeur devrait sortir aux USA au début de l'année prochaine. D'après les rares renseignements dont nous disposons, il s'agirait d'une machine dotée d'entrées-sorties (grace à une RS-232), d'environ 120 Ko de mémoire, mais surtout équipée d'une mini tablette à digitaliser, pemettant d'entrer des formules directement en écrivant dessus. Son prix sera d'environ 4000 Frs. Bien entendu, nous vous donnerons d'autres précisions dès que possible. Zut ! J'allais oublier son numéro : ce sera le 48.

En attendant, nous vous souhaitons bonne lecture...

Le Bureau

COURRIER DU COEUR

Jean Pierre Karpinski
23 rue Rosenwald
75015 PARIS
Tel : (1) 45 30 25 77

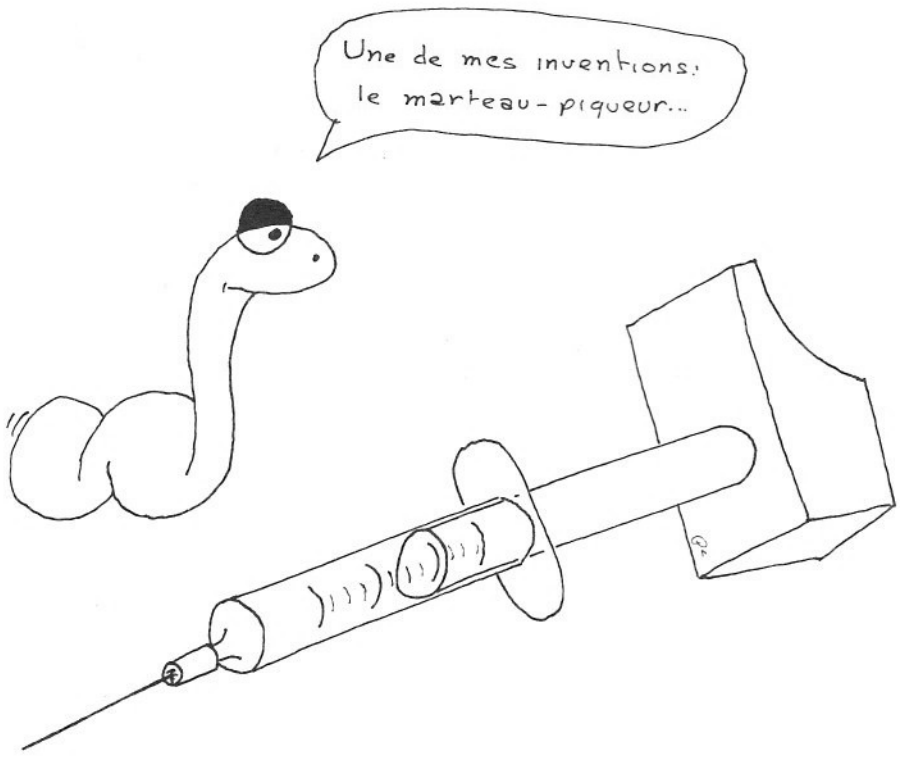
Vend :

HP-41CX avec accessoires. Prix à débatre.

Maubert Electronic
49 Bd St Germain
75005 Paris

Vend :

Accessoires pour HP-71 : Module éditeur de texte, Translator Pac, Lecteur de cartes ainsi que des logiciels pour HP-75. Materiel neuf.



HP-28

HP-28 S
PROGRAMMATION
ET APPLICATIONS

Ce livre de Jean-Michel Ferrard est un grand (235 p) recueil de programmes à vocation principalement mathématiques. On ne lui en veut pas, il est agrégé de Mathématiques et professeur de math. sup. à Toulouse (lycée Déodat de Séverac).

Le livre est composé de 14 groupes de programmes, au total 45 Ko, oui 45 Ko ce n'est pas une coquille, il vous faudra choisir. Voici un bref aperçu du contenu de cette corne d'abondance de programmes taupinaux. Tous les programmes sont agrémentés d'exemples, d'un diagramme de pile et d'un mode d'emploi.

Arithmétique

Dans ce répertoire on trouve des programmes de base. Décomposition en facteurs premiers, liste des diviseurs premiers, pgcd, ppcm, simplification de fraction, somme de fraction. On y trouve aussi des programmes plus complexes : fonction de Moebius, indicateur d'Euler et calcul des polynômes cyclotomiques

Nombres Réels et Complexes

Ce groupe de programmes tourne autour des sommes infinies, produits infinis, relations de récurences... SERIE permet le calcul des sommes partielles d'une série, PROD les produits partiels des produits infinis. FRCON calcule les fractions continues; R->Q passe d'un réel à sa forme fractionnaire. ITER permet les calculs récurrents (d'ordre 1 2 ou 3). XP et RACN calculent les racines nièmes. TRIG linéarise (le contraire) les puissances de sin et cos. Et enfin INTZ intègre sur un chemin du plan complexe.

Polynômes

Dans ce répertoire qui est un des plus gros, on trouve tout pour manipuler les polynomes : addition, multiplication, division selon les puissances croissantes ou décroissantes, d'élever à une puissance, de composer (utile pour les DL). Des programmes donnent les racines des polynomes de degré quelconque. On trouve aussi un générateur des polynomes de Tchébitchev (les deux espèces).

Calcul Matriciel

Vous pourrez trouver le rang, les valeurs, espaces propres et polynôme caractéristique d'une matrice carrée. SYST vous permetras de calculer la solution symbolique du systeme de n équations à p inconnues. Plus quelques programmes de manipulation de lignes et de colonnes.

Analyse

Non il n'y a pas toute l'analyse dans ce chapitre, seulement un programme d'approximation d'une fonction (moindre carrés), calcul du polynome, d'interpolation de Lagrange, résolution de systeme de deux ou trois fonctions (2 ou 3 variables), résolution de $y'=f(x,y)$ (Runge-Kutta) et un programme de calcul des sommes partielles de la série de Fourier d'une fonction.

Développements Limités

En codant les DL par des vecteurs on obtient un temps de calcul impressionant au prix il est vrai de 6 Ko de mémoire.

Géométrie

Quelques programmes qui résolvent des petits problèmes. Une application est de déterminer si le rayon laser touchera ou non la soucoupe volante, mais l'auteur du livre n'est apparament pas intéressé par ce sujet, dans ce livre.

Géométrie Différentielle

Pour serulement 2,6 Ko de programmes les possibilités sont impressionnantes. Rectification des longueur des courbes, integrales curviligne, calcul d'aire délimité par une courbe, rayon de courbure, divergence, rotationnel, gradient, laplacien, et enfin le calcul de la différentielle d'une fonction.

Graphisme

Tout pour le tracé paramétré, polaire, les deux, les fonctions implicites, l'enveloppe d'une famille de droites. Le tout agrémenté d'un programme de tracé automatique de familles et d'animation.

Entiers Longs

On devrait dire très longs, même très très longs car leur taille n'est limité que par la mémoire et le temps de calcul. Exemple :

148 472 100 632 * 911 048 399 582 500 002 =
135 265 269 663 435 487 083 084 201 264

Probabilités

Plus besoin des tables.

Statistiques simples

5,5 Ko pour 18 programmes : calcul des moments, des quantiles, des moyennes, et tracé des diverces courbes et histogrames.

Statistiques Simples

Etude des statistiques doubles discrètes, régression et corrélation.

Bases de Données

Les enrgistrements peuvent avoir jusqu'à 4 champs (chaînes de caractères). les fonctions sont courantes tri, recherche, modification, suppression, et ajout.

Ce livre est diffusé par *D2I DIFFUSION* pour un prix de 200F et pourra être consulté lors de nos prochaines réunions.

Laurent Chouraki (472)

METHODES DE TRI

Tri à bulles

Ce programme prend une liste au niveau 2, un programme au niveau 1 et renvoie la liste triée. Le programme doit opérer sur deux éléments de la liste A1 et A2 et donner 1 si A1 doit être placé avant A2 et 0 sinon.

Exemple : pour trier une liste de nombres par ordre croissant ou une liste de chaînes par ordre alphabétique, entrer la liste puis « < » et exécuter TRI.

Le premier élément constitue une sous-liste triée, on lui ajoute l'élément suivant de manière à obtenir une nouvelle sous-liste triée. On recommence jusqu'au dernier élément de la liste. Pour ajouter un élément E à une liste triée, on le compare au dernier élément de la liste, si E lui est inferieur, on les échange. On réitère ce test entre E et l'element le précédant jusqu'à ce qu'il ne soit pas vérifié ou qu'on arrive au début de la liste. On a alors obtenu une nouvelle liste triée.

```
TRI
«
  -> CLAUSE
  «
    DUP SIZE 2 SWAP
    FOR j j 1 - 1
      FOR k
        k GETI -> n1
        «
          GETI -> n2
          «
            DROP
            IF n1 n2 CLAUSE NOT THEN
              k n2 PUTI n1 PUT
            ELSE
              1 'k' STO
            END
          »
        »
      -1 STEP
    NEXT
  »
»
```

Tri par dichotomie

Ce programme s'utilise de la même manière que celui de tri à bulle mais fonctionne différemment. Il est plus rapide mais occupe plus de mémoire.

On prend les 2 premiers éléments, on les trie puis on insère l'élément suivant de manière à obtenir une nouvelle sous-liste triée. On recommence jusqu'au dernier élément de la liste. Pour insérer un élément E dans une liste triée, on le compare avec le premier et le dernier élément pour savoir si on peut l'ajouter à une des extrémités, sinon on le compare avec l'élément du milieu de la liste. On sait ainsi dans quelle moitié il se trouve, on recommence avec cette moitié jusqu'à ce que E soit ajouté (le procédé finit toujours par s'arrêter, car au pire on arrive à une liste de taille 1, E est donc soit avant soit après).

```
TRI2
«
  -> CLAUSE
  «
    DUP 1 GET 1 ->LIST OVER SIZE 2 SWAP
    FOR F OVER F GET -> X
      «
        IF X OVER 1 GET CLAUSE EVAL THEN
          X 1 ->LISTE SWAP +
        ELSE
          IF DUP F 1 - GET X CLAUSE EVAL THEN
            X 1 ->LIST +
          ELSE
            1 F 1 -
            WHILE DUP 2 - -1 == NOT
            REPEAT
              DUP2 + 2 / IP 4 PICK OVER GET X SWAP
              IF CLAUSE EVAL THEN
```

```
            SWAP DROP
          ELSE
            ROT DROP SWAP
          END
        END
        3 DUPN DROP 1 SWAP SUB X 1 ->LIST +
        4 ROLLD SWAP DROP F SUB +
      END
    END
  »
NEXT
SWAP DROP
»
»
```

Guillame Le Stum (432)

UNE FONCTION LINE

Source: La découverte du CBM 64 de D.J. David

L'objet de cette routine est de tracer une ligne entre deux points de l'écran de la HP-28.

Cette routine se décompose en deux subroutines. La première vise à restaurer des coordonées de pixels de manière à ce qu'ils aillent de 1 en 1. En effet, la HP-28 peut numéroter ses pixels de façon infinie selon l'échelle choisie par l'utilisateur. Or, la routine LINE - déjà extremmement lente - se ralentit en recalculant à chaque fois le pas de décompte lorsqu'il trace les points un à un à l'écran. Cette première subroutine, ECHLI (échelonne LINE) fixe ce pas un fois pour toutes à 1 (ou -1) au niveau du STEP et réajuste (avec PX et PY) les coordonées entrées et celles de l'écran. Votre ancien PPAR sera sauvegardé dans la variable NPAR.

Tâchez donc d'executer ECHLI toujours au moins *une* fois avant le premier LINE !

Quant au second programme:

On a des pointillés au lieu d'une ligne continue car on trace des points qui, s'ils correspondent à des X consécutifs, ne correspondent pas à des Y consécutifs ! (la pente est supérieure à 1) En fait la solution est de tracer tous les points $(X,Y(X))$; $(X,Y(X)+1)$; ...; $(X+1,Y(X+1)-1)$; $(X+1,Y(X+1))$ au lieu de $(X,Y(X))$ et $(X+1,Y(X+1))$, et ceci, par une bête formule d'interpolation...

Ainsi il faudra faire une boucle sur les X, soient $(X0,Y0)$ et $(X1,Y1)$ les points de départ et d'arrivée, et (X,Y) le point intermédiaire, la formule sera :

$X=X0+(Y-Y0)(X1-X0)/(Y1-Y0)$ avec Y allant du Y précédent au Y final.

Mais la pente peut aussi poser problème en Y ! En suivant le même principe qu'avant, on aurait deux boucles imbriquées...
Un truc de D.J. David :
On peut gagner une boucle en remarquant que si: $|\ X1-X0\ | > |\ Y1-Y0\ |$, les Y correspondant à deux X consécutifs seront à fortiori consécutifs. On pourra donc ne mettre que la boucle sur X. Et si $|\ X1-X0\ | < |\ Y1-Y0\ |$, il suffit d'échanger les rôles de Y et de X !

On peut faire de jolis pointillés en multipliant la pas par 2, 3, ... ce qui accélère le tracé. Le pied serait qu'un lecteur plus astucieux nous informe d'une autre méthode plus rapide, ou qu'il réécrive ce programme en code-machine, si c'est possible!

```
ECHLI
«
  'PPAR' DUP 'NPAR' STO RCL LIST-> 4 DROPN - C->R
  31 SWAP / ABS 'PY' STO PY H*
  136 SWAP / ABS 'PX' STO PX W*
»

LINE
«
  C->R ROT C->R PY * SWAP PX * ROT PY *
  4 ROLL PX * -> Y0 X0 Y1 X1
  «
    IF Y1 Y0 - ABS X1 X0 - ABS > THEN
      Y1 Y0 - SIGN 'ST' STO Y0 Y1
      FOR Y Y Y0 - X1 X0 - * Y1 Y0 - /
        X0 + Y R->C PIXEL
      ST STEP
    ELSE
      X1 X0 - SIGN 'ST' STO X0 X1
      FOR X X X0 - Y1 Y0 - * X1 X0 - /
        Y0 + X SWAP R->C PIXEL
      ST STEP
    END
  »
»
```

Mode d'emploi :

1 - Assurez-vous que l'échelle est bonne: faites ECHLI si vous ne l'avez pas déjà fait.
2 - Entrez les coordonnées des points:
- de départ (sous forme de nombres complexes).
- d'arrivée.
3 - Faites LINE

Michel Polski (531)

HP-71

P. David / J. Taillandier — Editeur de texte en assembleur — *8*

L'EDITEUR DE TEXTES

Nous sommes heureux et fiers de vous présenter aujourd'hui la dernière création des ateliers Taillandier / David, le fruit de plusieurs mois de travail, un des programmes dont nous sommes le plus fiers : l'éditeur de textes en assembleur.

Qu'est-ce qu'un éditeur de textes ?

Posons les bonnes questions... et apportons-y les bonnes réponses !

Un *éditeur de textes* est tout simplement un programme pour rentrer des caractères dans un fichier TEXT, et les modifier car vous savez bien qu'il y a toujours des corrections à faire : tout le monde fait des phôtes dauretaugraf ! Notez que Hewlett-Packard vend un tel programme (nommé EDTEXT) dans ses modules *Forth / Assembleur* et *Text Editor*.

Le module *Text Editor* contient, en outre, un *formateur de textes*. Il s'agit d'un programme pour aligner des paragraphes, centrer des lignes, etc. De tels programmes ont déjà été publiés dans *JPC*. Ne confondez donc pas *formatteur* et *éditeur*.

Pourquoi un éditeur ?

Tiens, oui, pourquoi ? Il en existe pourtant déjà un !

Certes, Hewlett-Packard vend EDTEXT dans deux de ses modules. Mais tous ceux qui ont essayé ce programme connaissent sa lenteur effroyable. Point n'est besoin d'avoir tapé tous les articles de *JPC* depuis les débuts pour s'en rendre compte ! EDTEXT met souvent quelques dizaines de secondes pour s'initialiser, et autant pour sortir. Le moindre ajout d'une ligne prend un temps intolérable en utilisation interactive.

Le problème est donc posé et de nombreuses personnes se sont penchées sur le problème. Des éditeurs en assembleur ont été écrits à divers endroits dans le monde.

Un des premiers essais sérieux a été réalisé il y a fort longtemps par notre ami Henrik Helnæs, de Copenhague au Danemark. Il était basé sur un principe voisin de l'éditeur du HP-75 c'est à dire que vous éditiez un fichier texte comme un programme Basic. Malheureusement, son éditeur n'a jamais dépassé le cadre expérimental.

Ensuite, Jean-Jacques Moreau s'est attelé à la tâche. Le résultat d'un été très studieux fut TEDIT, que certains d'entre vous connaissent peut être. Son éditeur, à l'opposé de celui d'Henrik, était strictement compatible avec EDTEXT d'HP. Cet éditeur était une belle réussite, mais il souffrait de quelques problèmes majeurs (par exemple, certaines fonctions provoquaient un Memory Lost). De plus, il n'a jamais été terminé et est depuis resté à l'état de chantier.

C'est vers le milieu de l'année dernière que nous nous sommes à notre tour penchés sur le problème. Il a fallu 30 jours complets de programmation répartis sur onze mois pour le résoudre.

Le cahier des charges

Nous nous étions fixés trois objectifs :
- rapidité maximum,
- totalement compatible avec EDTEXT d'HP, et
- ajouter des extensions telles que les chaînes génériques compatibles Unix.

La rapidité est la raison d'être de cet éditeur. La programmation en assembleur ne suffit pas, comme l'a montré l'éditeur de Jean-Jacques. En effet, son éditeur devenait très lent dès que le fichier prenait du poids. Dès qu'il dépassait les 500 lignes, le ralentissement était très net, pour la simple raison qu'il devait parcourir toutes les lignes à chaque opération. Aujourd'hui, avec notre éditeur, vous ne prenez pas plus de temps pour vous rendre à la ligne 5000 qu'à la première du fichier. Le résultat est toujours *immédiat*. La rapidité était notre principal objectif. Nous avons tout fait pour y arriver.

La compatibilité avec EDTEXT était un élément important. En effet, la majorité d'entre vous ont déjà pratiqué et pratiquent encore cet éditeur. Vous êtes donc habitués à son jeu de commandes, à ses caractères génériques, etc. Seuls quelques points mineurs ont été modifiés à la lumière de notre expérience. Les différences sont listées dans la page de manuel consacrée à TEDIT.

Enfin, il était indispensable d'ajouter des extensions. Par exemple, EDTEXT dans le module Forth Assembleur n'a pas de fonction d'aide. Nous en avons intégré une, plus utile d'ailleurs que celle du module Text Editor. Une autre extension est l'accès à la pile de commandes, que beaucoup d'entre nous regrettaient de ne pouvoir utiliser avec EDTEXT. Une autre extension bien pratique est l'ajout des chaînes génériques et d'un mode de fonctionnement inspirés des éditeurs existant sous Unix. Eh oui, le HP-71 se met à ressembler à Unix !

Sous le capot

Pour les passionnés d'assembleur HP-71, voici quelques détails sur le fonctionnement interne de l'éditeur. Il ne s'agit pas de faire une revue des 13657 lignes ou des 351629 octets composant le source de XEDIT, mais simplement de donner un aperçu sur certains points qui nous semblent intéressants.

Tout d'abord, il faut savoir que la rapidité de XEDIT tient à l'utilisation d'une *table d'accès rapide*. Le gros problème des fichiers TEXT, qui influe sur les éditeurs classiques tels que EDTEXT ou l'éditeur de Jean-Jacques Moreau, est la nécessité de parcourir tout le fichier depuis le début jusqu'à la ligne spécifiée. Ainsi, pour éditer la ligne 3562, il faut parcourir 3561 lignes pour obtenir l'adresse de la ligne demandée. C'est un problème majeur, car même en assembleur, cela devient très lent. En revanche, lorsque vous rentrez sous XEDIT, l'éditeur parcourt le fichier spécifié, et mémorise l'adresse des lignes 200, 400, 600, etc. (c'est en fait l'adresse relative par rapport à la deux-centaine précédente). Lors des accès ultérieurs, il suffit de parcourir la table. Ainsi, pour accéder à la ligne 3562, il faut parcourir 17 entrées dans la table d'accès rapide pour arriver à la ligne 3400, puis parcourir 162 lignes dans le fichier. L'accès à n'importe quelle ligne devient alors extrêmement rapide.

Lorsque vous détruisez ou ajoutez des lignes, cette table est actualisée, mais pas recalculée : si vous détruisez la ligne 350, la deuxième entrée dans la table pointera sur la ligne 399, ce qui évite un recalcul long. Il y a ainsi une tolérance dans la table, qui peut aller jusqu'à plus ou moins 50 lignes. Lorsque cette tolérance est dépassée, la table est recalculée. Concrètement, cela signifie que si vous éditez un gros fichier et que vous faites des gros ajouts ou destructions, vous noterez peut-être de temps à autre un ralentissement ponctuel.

Cette table est mémorisée dans un buffer temporaire, qui est détruit lorsque vous sortez de XEDIT. Chaque entrée occupant 10 quartets, cela autorise au maximum environ 80000 lignes de texte. Si vous avez un fichier plus gros, les accès aux dernières lignes ne seront plus aussi rapides. Mais si vous avez un tel fichier, contactez-nous !

Un autre problème intéressant est celui des chaînes génériques. L'algorithme de HP tel que décrit dans les I.D.S. Volume I ressemble plus à une astuce de programmeur des années 60 qu'à un algorithme ! Nous avons préféré nous référer à l'excellent ouvrage *Software Tools* de Brian W. Kernighan et P. J. Plauger, aux éditions Addison Wesley. Brièvement, cet algorithme est divisé en deux parties :

- compilation de la chaîne générique en une succession de tokens, puis

- recherche de la chaîne compilée dans le fichier.

Cette chaîne compilée est placée dans un buffer, lui aussi temporaire. La recherche de la chaîne compilée est hautement récursive. Mais, heureusement, depuis un certain article dévoilant tous les mystères de la récursivité en assembleur (?), la programmation fut extrêmement simple !

Un autre point intéressant de XEDIT est que les affichages et les noms des comandes sont dans une table de message. Cela donne un code plus complexe, mais la traduction des messages est maintenant possible. Nous avons fait, à titre expérimental, un traducteur. Et il marche ! Si, si ! Pour les vieux de la vieille, cela fonctionne comme le Lex FRALEX avec EDTEXT !

Il reste encore beaucoup de choses à dire sur les entrailles de XEDIT. Mais, après tout, vous avez l'intégralité des sources de XEDIT, et vous pouvez partir à la découverte vous-mêmes ! De toute façon, nous nous tenons à votre disposition pour en parler lors des réunions mensuelles.

Comment cela s'utilise-t-il ?

Après les détails mécaniques, passons à la manière de conduire l'engin...

Vous trouverez à la suite de cet article les pages de manuel que nous avons rédigées pour le manuel de JPC Rom. Si vous savez déjà vous servir de EDTEXT d'HP, nous vous conseillons de lire la page de TEDIT, puis celle de XEDIT. Si vous n'avez jamais utilisé EDTEXT, passez directement à XEDIT.

Et les fonctions auxiliaires ?

Il aurait été dommage de ne pas en profiter...

Lorsque nous avons entamé la conception de cet éditeur, nous avions comme projet de fournir des fonctions auxiliaires permettant d'exploiter les chaînes génériques directement en Basic, dans la plus pure tradition du Basic HP-71. Il aurait été dommage de ne pas profiter de l'occasion !

Nous avons donc écrit trois fonctions de recherche et de remplacement dans une chaîne (GENPOS, GENLEN et GENRPLC$), et une fonction de recherche dans un fichier (FILEPOS). Notons que le remplacement dans le fichier est assuré simplement par XEDIT lui-même !

Nous en avons profité également pour réécrire un mot-clef qui n'arrêtait pas de nous embêter depuis un certain temps, malgré les corrections de bug successives : il s'agit du très récalcitrant `FIND`. Il a profité de ce lifting pour accepter lui aussi les caractères génériques, ainsi que deux paramètres supplémentaires (et optionnels) pour préciser les lignes de début et fin de recherche. Cet ajout supprime le handicap de `FIND` pour trouver l'occurrence lorsque celle-ci est tapie dans la première ligne du programme.

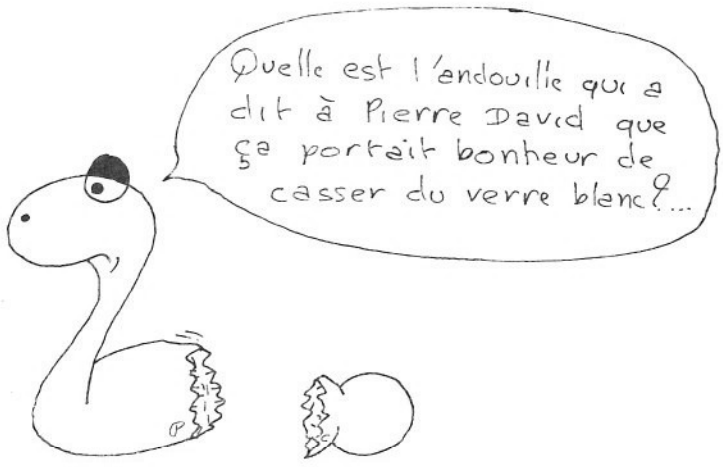
Cet article est, nous en sommes conscients, bien trop bref pour décrire le sujet. Il faudrait au moins un *JPC* complet pour expliquer en détail, progressivement et complètement le fonctionnement et l'utilisation de l'éditeur. Nous espérons toutefois que les pages de manuel vous suffiront pour l'apprendre et l'apprécier.

Enfin, sachez que toutes vos remarques, que ce soit sur l'utilisation, sur le manuel ou sur le code, seront toujours les bienvenues. Les colonnes de *JPC* vous sont plus que jamais ouvertes !

A bientôt,

Pierre David (37) (SIG1)
Janick Taillandier (246) (SIG7)

NDLR : Etant donné la taille de ce lex (10136 octets !) nous avons jugé inutile de le présenter dans le coin des Lex. Vous pouvez cependant l'obtenir en nous envoyant une disquette 3½", que nous vous renverrons après copie.



TEDIT

TEDIT (Text EDITor) est un éditeur de textes extrêmement rapide, compatible avec l'éditeur de textes Hewlett-Packard.

- ■ Ordre
- O Fonction
- O Opérateur
- ■ Exécution au clavier
- O Mode CALC
- ■ IF..THEN..ELSE
- ■ Opération d'unité

TEDIT *fichier*
TEDIT *fichier* , *chaîne de commandes*

Exemples

`TEDIT ESSAI`

Crée le fichier ESSAI et rentre sous l'éditeur de textes en mode commandes.

`TEDIT ESSAI,"1#R/TOTO/TATA/;E"`

Appelle l'éditeur de textes sur le ficher ESSAI, substitue toutes les occurrences de la chaîne TOTO par la chaîne TATA et sort de l'éditeur.

`TEDIT ESSAI:TAPE`

Copie le fichier ESSAI en mémoire et rentre sous l'éditeur de textes en mode commandes.

Paramètres d'entrée

Elément	Description	Restrictions
fichier	Expression alphanumérique ou chaîne sans guillemets.	Spécificateur de périphérique en option.
chaîne de commandes	Expression alphanumérique. Défaut : Aucune commande n'est exécutée.	Aucune.

Opération

TEDIT n'est inclus dans JPC Rom que pour des raisons de compatibilité, et son utilisation n'est conseillée que dans ce cas.

TEDIT est un éditeur de textes compatible avec l'éditeur de textes EDTEXT inclus dans les modules Forth / Assembleur et Text Editor de Hewlett-Packard. Il est cependant conseillé d'utiliser XEDIT, plus agréable et plus puissant.

Le lecteur désirant utiliser TEDIT pour des raisons de compatibilité avec EDTEXT est invité à se reporter au manuel de ce dernier pour référence, puis au paragraphe **différences entre TEDIT et EDTEXT**.

TEDIT (suite)

Le lecteur désirant utiliser XEDIT et connaissant déjà EDTEXT ou TEDIT est invité à se reporter au paragraphe **différences entre TEDIT et XEDIT**, puis au manuel de XEDIT pour la référence.

Différences entre TEDIT et EDTEXT

Ce paragraphe suppose que vous connaissez déjà le fonctionnement de EDTEXT. Si ce n'est pas le cas, nous vous conseillons d'apprendre directement le fonctionnement de XEDIT.

La principale différence entre TEDIT et EDTEXT est la rapidité. TEDIT est entièrement en assembleur et optimisé spécialement pour les grands fichiers, EDTEXT est pour une grande partie en Basic, et est donc très lent.

Les autres différences sont résumées ci-dessous.

Commandes ajoutées

TEDIT dispose de commandes supplémentaires :
- H (*Help*) : affiche une aide,
- J (*Join*) : réunit plusieurs lignes consécutives en une seule,
- Q (*Quit*) : équivalent à E (*Exit*), et
- X (*eXchange file*) : édite un autre fichier sans sortir de TEDIT.

Pour plus de détails sur ces commandes, voir le manuel de XEDIT.

Commandes différentes

Les commandes C (*Copy*) et M (*Move*) de EDTEXT n'autorisent pas la copie ou le déplacement d'un bloc de texte incluant la ligne courante, sauf si cette ligne est la première ou la dernière du bloc. Les commandes C et M de TEDIT ne sont sujettes à aucune restriction.

Fonctionnalités supplémentaires

TEDIT diffère de EDTEXT par quelques détails de comportement, qui sont le plus souvent des ajouts motivés par une longue utilisation de EDTEXT :

- La touche [g][CMDS] active la pile de commandes, et permet d'utiliser les touches de curseur verticales ([↑], [↓], [g][↑] et [g][↓]), que ce soit en mode *commandes* ou en mode *saisie de texte*. EDTEXT ne le permet pas, ou ne le permet qu'au prix d'une manipulation hasardeuse.

- La touche [f][CONT] affiche le nom du fichier, la ligne courante et le nombre total de lignes du fichier. EDTEXT n'affiche que le nom du fichier.

- En mode commandes, les touches de curseur verticales ([↑], [↓], [g][↑] et [g][↓]) n'affichent pas la nouvelle ligne. EDTEXT affiche la nouvelle ligne. TEDIT permet ainsi de se déplacer plus vite dans le fichier, l'affichage d'une ligne étant maintenant demandé explicitement par l'utilisateur en appuyant sur [ENDLINE]

- La touche [RUN] est désactivée. Avec EDTEXT, elle agit comme la touche [ATTN], ce qui provoque souvent la perte d'une ligne de texte en cours de frappe.

- La touche [f][USER] est maintenant active, ce qui permet de valider ou non le mode *user*. EDTEXT, à cause de sa programmation, interdit de passer en dehors du mode *user* (ou ne le permet qu'au prix d'une manipulation hasardeuse).

- La sortie de TEDIT ne provoque pas l'affichage du message Done, car la sortie est instantanée.

- La touche [ATTN][OFF] (ou une inactivité de plus de 10 minutes) sort de TEDIT. Avec EDTEXT, cela ne provoque que la perte de la ligne courante avec retour au mode *commandes*, sans possibilité ultérieure d'extinction.

Différences entre TEDIT et XEDIT

Les éditeurs TEDIT et XEDIT sont très proches l'un de l'autre. L'objectif de TEDIT est de rester compatible avec EDTEXT, alors que l'objectif de XEDIT est d'offrir un environnement d'édition proche de celui de l'éditeur ed sous le système Unix (Unix est une marque déposée par AT&T). Ces deux objectifs ont conduit à des divergences entre TEDIT et XEDIT lorsqu'ils étaient incompatibles.

Différences dans les chaînes génériques

Le principe des chaînes génériques sous EDTEXT est d'offrir deux modes : un mode *normal* et un mode *générique*. Par exemple, la chaîne A\@1\$ signifie :
- chercher un caractère A en mode normal, puis
- passer en mode générique, puis
- chercher un caractère quelconque de caractères quelconques (caractère spécial @), puis
- chercher un caractère 1 (le caractère 1 n'est pas spécial), puis
- repasser en mode normal, et enfin
- chercher un caractère $ (le caractère $ est spécial, mais nous ne sommes plus en mode générique).

Sous XEDIT, on est en permanence en mode *générique*. Si on veut utiliser un caractère spécial comme un caractère normal (exemple du $ ci-dessous), il faut le faire précéder par le caractère \. On passe donc en mode *normal* pour un et un seul caractère.

Hormis cette différence liée au mode, le caractère @ n'a pas de signification spéciale sous XEDIT (il est remplacé par la séquence . .*). La table ci-dessous représente les caractères spéciaux utilisés lors des recherches avec XEDIT :

Caractère	Signification
\	annule la signification du caractère suivant
^	début de ligne
$	fin de ligne
.	caractère quelconque
[ ]	ensemble de caractères
[^]	complémentaire de l'ensemble
*	répétition du motif précédent 0 ou n fois

L'exemple ci-dessus est donc A. .*\$ avec XEDIT.

Différences dans le format des paramètres

Les paramètres des commandes de TEDIT sont de simples numéros de lignes. Avec XEDIT, ce sont de véritables expressions contenant :
- des numéros de lignes classiques,
- le symbole . pour désigner la ligne courante,
- le symbole $ pour désigner la dernière ligne du fichier,
- des chaînes génériques.

Par exemple, pour détruire toutes les lignes entre la suivante et celle juste avant la prochaine occurrence d'une majuscule en début de ligne, on fera sous XEDIT :

```
.+1 , /^[A-Z]/-1 D
```

Les espaces ont été mis ici pour plus de clarté, ils ne sont bien évidemment pas nécessaires lorsque vous tapez la ligne.

On comprend aisément, au vu de cet exemple, la différence de puissance entre TEDIT et XEDIT.

Références

JPC ?? (page ??) Première version de TEDIT par Pierre David et Janick Taillandier.

Manuel du module Editeur de Textes par Hewlett-Packard.

Manuel du module Forth / Assembleur par Hewlett-Packard.

Mots-clés associés

XEDIT

Auteurs

Pierre David et Janick Taillandier.

XEDIT

XEDIT (eXtended EDITor) est un éditeur de textes étendu.

■ Ordre	■ Exécution au clavier
O Fonction	O Mode CALC
O Opérateur	■ IF..THEN..ELSE
	■ Opération d'unité

```
XEDIT fichier
XEDIT fichier , chaîne de commandes
```

Exemples

`XEDIT ESSAI`	Crée le fichier ESSAI et rentre sous l'éditeur de textes en mode commandes.
`XEDIT ESSAI,"1,$R/TOTO/TATA/;E"`	Appelle l'éditeur de textes sur le fichier ESSAI, substitue toutes les occurrences de la chaîne TOTO par la chaîne TATA et sort de l'éditeur.
`XEDIT ESSAI:TAPE`	Copie le fichier ESSAI en mémoire et rentre sous l'éditeur de textes en mode commandes.

Paramètres d'entrée

Elément	Description	Restrictions
fichier	Expression alphanumérique ou chaîne sans guillemets.	Spécificateur de périphérique en option.
chaîne de commandes	Expression alphanumérique. Défaut : Aucune commande n'est exécutée.	Aucune.

Opération

Qu'est-ce qu'un éditeur de textes ?

Un *éditeur de textes* est un programme permettant de rentrer des caractères dans un fichier TEXT du HP-71. Hewlett-Packard fournit un éditeur de textes dans les modules *Forth/Assembleur* d'une part, et *Text Editor* d'autre part.

Le module *Text Editor* comprend en outre un *formatteur de textes*, c'est à dire un programme pour aligner des paragraphes, centrer des lignes, etc. De tels programmes ont été publiés dans *JPC*. Un formatteur de textes opère sur des fichiers qu'il a donc fallu introduire dans le HP-71 à l'aide d'un éditeur de textes.

EDTEXT, l'éditeur de textes présent dans les deux modules de Hewlett-Packard, est rédigé en majeure partie en Basic, et souffre d'une lenteur caractérisée.

XEDIT (suite)

JPC Rom contient un éditeur de textes entièrement en assembleur, extrêmement rapide et dont les performances ne se dégradent pas, même avec de très grands fichiers.

Entrée sous l'éditeur

Lorsque vous exécutez l'ordre XEDIT suivi d'un nom de fichier, le fichier est soit :
- créé s'il n'existait pas,
- copié en mémoire s'il réside sur un support magnétique,
- ou simplement cherché en mémoire,

puis il est analysé pour vérifier s'il est bien du type TEXT, enfin l'éditeur passe en mode *commandes* (voir chapitre suivant).

Notez que l'indicateur 1 disparaît de l'affichage (le drapeau 1 conserve toujours son état en mémoire, seul l'affichage est affecté).

L'éditeur détermine en outre si votre fichier est modifiable, c'est à dire s'il n'est ni en Rom ou Eprom, pas sécurisé par SECURE, et pas déjà ouvert par ASSIGN #. Si votre fichier n'est pas modifiable, vous pouvez le visualiser, mais toute tentative de modification déclenchera l'erreur correspondant au type de protection.

Vous sortez de l'éditeur par la commande E ou Q. Si vous étiez en mode *saisie*, vous sortez en appuyant sur la touche [ATTN], puis en tapant une de ces deux commandes. Si vous ne savez pas en quel mode vous êtes, appuyez sur [ATTN], puis tapez la commande E ou Q.

Si vous aviez spécifié une chaîne de commandes, celle-ci aurait été exécutée automatiquement, sans passer en mode *commandes*. Lorsque vous utilisez une chaîne de commandes, faites attention toutefois à la terminer par la commande E ou Q pour sortir de l'éditeur, faute de quoi l'éditeur attendra une nouvelle commande.

Les modes d'opération

L'éditeur possède deux modes, le mode *commandes* et le mode *saisie*.

Le mode commandes

Lorsque vous rentrez sous l'éditeur, vous êtes placés en mode *commandes*. L'éditeur vous le signale en affichant un message tel que :

```
Line nnn, Cmd:        ou bien
Eof, Cmd:
```

et en plaçant le curseur à droite des caractères.

L'éditeur attend alors que vous tapiez une commande selon le format décrit dans le chapitre suivant.

Si vous laissez votre doigt appuyé sur la touche :
- [ENDLINE], la ligne est affichée, sans changer la ligne courante,
- [f][CONT], l'éditeur affiche le nom du fichier courant, le numéro de la ligne courante et le nombre de lignes dans le fichier.

Vous pouvez, en outre, utiliser les touches de curseur verticales pour passer à la ligne suivante (touche [↓]), revenir à la ligne précédente (touche [↑]), aller directement à la dernière (touche [g][↓]) ou à la première (touche [g][↑]) ligne du fichier.

Vous pouvez aussi choisir d'utiliser la pile de commandes en appuyant sur [g][CMDS], de la même manière qu'en utilisation normale.

XEDIT (suite)

Toutes les touches d'édition (touches de curseur horizontales, [f][BACK], [f][CHAR], etc.) sont utilisables pour faciliter l'introduction de vos commandes.

Le mode saisie

Vous pouvez choisir de passer en mode *saisie* (saisie de texte avec les commandes T (*text*) ou I (*insert*). Tout ce que vous introduisez est alors rentré dans votre fichier, jusqu'à ce que vous appuyiez sur la touche [ATTN] qui vous replace en mode commandes.

La touche [f][CONT] est là aussi active, ce qui vous permet de connaître à tout moment le nom du fichier en cours d'édition, le numéro de la ligne courante et le nombre de lignes dans le fichier.

La touche [f][VIEW] est aussi active (en mode USER seulement) et vous donne la position du curseur dans la ligne.

La pile de commandes est accessible à tout moment, toujours en appuyant sur [g][CMDS].

Inactivité de plus de dix minutes

L'éditeur détecte les périodes d'inactivité de plus de dix minutes et :
- si vous étiez en mode *saisie*, repasse en mode *commandes*,
- si vous étiez en mode *commandes*, sort de l'éditeur,

Ceci assure l'extinction automatique de votre HP-71B au bout de 20 ou 30 minutes suivant le mode dans lequel vous étiez.

Syntaxe des commandes

Les commandes que vous introduisez ont la syntaxe suivante :

[*début* [[,]*fin*]][?] [*commande*] [*paramètre*]

où les éléments entre crochets sont optionnels. Certaines commandes peuvent restreindre leur syntaxe : par exemple, la commande T n'admet pas de *fin*, pas de point d'interrogation, pas de *paramètre*.

Les éléments *début* et *fin* sont des numéros de ligne spécifiant un intervalle bornes comprises (si les deux sont présents), une simple ligne (si *début* est seul) ou la ligne courante (si aucun n'est présent). Pour plus de précision, voir les valeurs par défaut pour chacune des commandes.
Ces éléments sont en fait de véritables expressions pouvant faire jouer :
- la ligne courante (symbole `.`),
- la dernière ligne du fichier (symbole `$`),
- une constante numérique, ou
- la prochaine occurrence d'un certain motif (chaîne encadrée par `/`).

Par exemple, l'expression `.+5,/exemple/-3` signifie : de la ligne courante plus cinq lignes jusqu'à la prochaine occurrence de la chaîne *exemple* moins trois lignes.

Le point d'interrogation (utilisé par les commandes S, R et J) spécifie qu'une commande doit être exécutée en mode *confirmation*, c'est à dire qu'une question vous est posée, pour la commande R par exemple, à chaque remplacement.

XEDIT (suite)

La *commande* est une simple lettre. Il n'y a pas de distinction entre majuscule et minuscule. Le cas spécial où la commande est vide implique le positionnement sur une ligne. Par exemple, pour vous positionner sur la ligne 523, vous taperez simplement 523, puis [ENDLINE]. Vous pouvez aussi vous rendre directement à la prochaine occurrence de la chaîne *exemple* en tapant `/exemple` (notez que le / final peut être omis ici), puis [ENDLINE].

Le *paramètre* est enfin un paramètre optionnel de la commande. Par exemple, il permet à la commande H (Help) de savoir si vous désirez l'aide sur toutes les commandes, ou sur une commande particulière.

Vous pouvez placer plusieurs commandes sur la même ligne, séparées par le caractère `;` (point-virgule). Elles seront alors exécutées en séquence, sauf si l'une d'entre-elles s'interrompt à cause d'une erreur. Cette facilité de regrouper des commandes est très intéressante lorsque vous appelez XEDIT avec une chaîne de commandes.

Chaînes génériques

Le terme *chaîne générique* est utilisé pour définir les chaînes avec ou sans caractère spécial utilisées dans les recherches ou les remplacements avec XEDIT et les fonctions associées (FILEPOS, FIND, GENLEN, GENPOS et GENRPLC$).

Ces chaînes génériques sont compatibles avec les chaînes génériques sous Unix (Unix est une marque déposée par AT&T), et plus particulièrement avec l'éditeur `ed` d'Unix.

Chaînes génériques de recherche

Les chaînes génériques sont le plus souvent utilisées lors des recherches de chaînes dans le fichier.

Pour chercher une suite de caractères standards, il n'y a aucun problème. Par contre, les choses se compliquent dès que vous avez des besoins un peu plus compliqués, comme par exemple chercher un chiffre quelconque, une majuscule en début de ligne, un `a` et un `b` sur la même ligne, etc. Il faut alors que vous insériez dans votre chaîne de recherche des caractères spéciaux pour représenter ces cas.

La table ci-dessous décrit les caractères spéciaux utilisés dans les chaînes de recherche :

Caractère	Signification
`\`	annule la signification du caractère suivant
`^`	début de ligne
`$`	fin de ligne
`.`	caractère quelconque
`[ ]`	ensemble de caractères
`[^ ]`	complémentaire de l'ensemble
`*`	répétition du motif précédent 0 ou n fois

- Le caractère `\` (back-slash) est le caractère spécial des caractères spéciaux ! Si vous voulez chercher un point, par exemple, cela est impossible puisque c'est un caractère spécial et XEDIT le prendra comme tel. Il faut donc annuler la signification spéciale du point, pour ne plus le considérer que comme un caractère standard. Pour chercher un point, la chaîne générique sera donc : "`\.`" (les guillemets ne font pas partie de la chaîne, et sont là simplement pour la lisibilité).

- Le caractère `^` signifie, lorsqu'il est placé au début d'une chaîne de recherche, *début de ligne*. Ceci vous permet, par exemple, de chercher `toto` en début de ligne : vous utiliserez la chaîne générique : "`^toto`".

- Le caractère `$` signifie, lorsqu'il est placé à la fin d'une chaîne de recherche, *fin de ligne*. Ceci vous permet, par exemple, de chercher `tata` en fin de ligne : vous utiliserez la chaîne générique : "`tata$`".

- Le caractère . signifie *n'importe quel caractère*. Pour chercher un X en deuxième position sur une ligne, vous utilisez la chaîne générique : "^.X".

- Les caractères [et] servent à chercher un caractère dans un ensemble. Un ensemble ne peut trouver qu'un et un seul caractère. Il existe plusieurs types d'ensembles : les ensembles par énumération et les intervalles.
Un ensemble par énumération est, par exemple, la chaîne générique "[0123456789]" : elle représente un (et un seul) chiffre. L'ensemble est défini comme l'ensemble de tous les caractères entre les crochets.
Un ensemble de type intervalle est, par exemple, la chaîne générique "[0-9]", qui représente la même chose que ci-dessus, mais de manière plus condensée. Un tel ensemble est défini par tous les caractères de code ASCII compris entre les deux bornes.
La séparation entre ces deux types d'ensembles n'est pas si rigide, et les deux types peuvent cohabiter dans un seul ensemble. Par exemple, pour identifier un caractère qui peut être une lettre (minuscule ou majuscule) ou un chiffre impair, on utilisera la chaîne générique suivante : "[A-Za-z13579]". Il y a là deux intervalles et une énumération dans le même ensemble.

- Lorsqu'un caractère ^ est le premier caractère dans un ensemble, cela signifie que cet ensemble ne trouvera pas un caractère appartenant à cet ensemble, mais au contraire un caractère *n'appartenant pas* à cet ensemble. Par exemple, pour trouver un caractère qui ne soit pas une lettre, on utilisera la chaîne générique : "[^A-Za-z]".

- Le caractère *, quant à lui, ne représente pas un caractère, mais la répétition 0 ou n fois du *motif précédent*. Par exemple, la combinaison "[A-Z]*" représente 0 ou n occurrences d'une majuscule. Le caractère * est indissociable du motif précédent.
D'autre part, remarquez que la répétition est faite *0 ou n fois*. Ceci signifie, par exemple, que "X*" sera trouvé, même dans une chaîne ne contenant pas de X. Si vous désirez trouver au moins un X, il faudra utiliser "XX*", ce qui signifiera : trouver le caractère X suivi du caractère X 0 ou n fois, donc trouver le caractère X 1 ou n fois.

Vous pouvez combiner ces caractères spéciaux entre eux ou avec des caractères normaux comme bon vous semble, pour arriver au résultat.

Voici quelques exemples, pour vous familiariser avec ces chaînes génériques :
- une référence à TEDIT ou XEDIT :
```
[TX]EDIT
```
- une ligne vide :
```
^$
```
- une ligne se terminant par des espaces :
```
*$                    note : la chaîne commence par deux espaces
```
- une phrase ne commençant pas par une majuscule :
```
\. *[^A-Z]            note : ne traite pas les phrase en début de ligne
```
- un nombre négatif :
```
-[0-9][0-9]*          note : au moins un chiffre
```
- un nombre hexadécimal en assembleur HP-71 :
```
#[0-9A-F][0-9A-F]*
```
- une affectation à un tableau numérique en basic HP-71 :
```
[^$]([^)]*)=
```

Chaînes génériques de remplacement

Lorsque vous désirez effectuer un remplacement, vous spécifiez une chaîne de recherche, délimitant le ou les motifs que vous voulez remplacer, puis une chaîne de remplacement qui spécifie par quoi vous allez remplacer votre motif. La chaîne de recherche est une de celles vues plus haut.

Ceci permet, par exemple, de remplacer toutes les suites d'espaces par un seul (c'est à dire remplacer toutes les occurrences de " *" (deux espaces) par " " (un seul espace)), et ainsi de simuler le comportement de la fonction REDUCE$ de JPC Rom).

Mais la chaîne de remplacement peut également contenir des caractères spéciaux, résumés dans la table ci-dessous :

Caractère	Signification
\	annule la signification du caractère suivant
&	substitue l'occurrence trouvée

- Le caractère \ a la même signification que précédemment,

- Le caractère & (*et commercial*) signifie : substituer par l'occurrence trouvée.

Ainsi, si on veut mettre un signe - devant tous les nombres, on remplacera "[0-9][0-9]*" par "-&".

Si on veut remplacer toutes les occurrences d'une majuscule par la même répétée deux fois, on remplacera "[A-Z]" par "&&".

Affectation du back-slash

Comme vous avez pu le constater, le caractère \ (*back-slash*) a une certaine importance sous XEDIT. Nous vous conseillons de l'affecter définitivement à une touche. L'éditeur de textes EDTEXT de Hewlett-Packard l'affectait à la touche [f][/] ce qui nous semble un bon choix. Vous pouvez donc faire `DEF KEY "f/",CHR$(92);` pour l'affecter à cette touche.

Limitations des chaînes génériques

Les chaînes génériques de recherche sont sujettes à des limitations du fait de la ressource interne utilisée (les buffers du système sont limités à 2 Ko) : il ne peut y avoir plus de 14 ensembles dans un chaîne générique. Sachez en outre qu'un ensemble occupe 128 octets.

Une autre limitation vient de l'algorithme utilisé pour traiter les *. Le cas pathologique est la recherche, par exemple, du motif .*XX (où XX n'existe pas dans votre fichier), qui peut devenir nettement plus longue que prévu.

Traduction des messages et commandes

Vous pouvez utiliser un traducteur de messages pour traduire les messages de XEDIT ainsi que ses commandes.

Utilisation de la mémoire

XEDIT, pour ses besoins propres, nécessite de la mémoire disponible dans votre HP 71. Cette mémoire est libérée dès que vous sortez de l'éditeur.

XEDIT, à l'initialisation, cherche à créer une table d'accès rapide aux lignes de votre fichier. Pour cela, l'éditeur peut consommer jusqu'à 2 Ko si votre fichier est très gros. Très exactement, il occupe 3,5 octets + 5 octets toutes les 500 lignes. Si votre fichier fait plus de 50000 lignes (environ), XEDIT s'arrêtera à 2 Ko. Si vous ne disposez pas de la place suffisante en mémoire, XEDIT prendra ce qui est disponible.

A chaque fois que vous faites une recherche, XEDIT alloue de la mémoire pour stocker la chaîne générique de recherche. La place occupée est 4 octets + 0,5 octets pour les caractères ^, $, . et *, 1,5 octets pour les caractères normaux et 128,5 octets pour les ensembles. S'il n'y a pas assez de mémoire, XEDIT affiche l'erreur `No Room for Pattern`.

Lors d'une commande C, M ou D dans un fichier externe, 256 octets au moins sont requis pour mener à bien l'opération dans des conditions de rapidité convenables. Plus de mémoire disponible accélère notablement le transfert.

Note à propos des grands fichiers

XEDIT crée une table d'accès rapide aux lignes de votre fichier, ce qui lui évite de parcourir tout le fichier depuis le début à chaque fois que vous faites la moindre opération. Cette création est faite lorsque vous rentrez sous XEDIT.

Toutefois, il peut arriver qu'au cours d'une séance d'édition, votre fichier s'amenuise ou grandisse. Lorsque XEDIT le jugera bon, il décidera de recalculer cette table d'accès rapide, ce qui peut se traduire par un ralentissement d'autant plus important que votre fichier est gros. Ceci dit, cette opération est rarement perceptible, le recalcul dépassant rarement l'ordre de la seconde.

Commande nulle

La commande nulle a comme seul effet de déplacer la ligne courante.

Syntaxe
ligne

Valeurs par défaut
aucune

Opération

La commande nulle a comme seul effet de déplacer la ligne courante. En profitant de l'utilisation des chaînes génériques dans les numéros de lignes, cela permet de faire une recherche très simplement. Par exemple, pour se déplacer à la prochaine occurrence de la chaîne `toto`, faire `/toto`, puis `[ENDLINE]`.

Commandes Exit et Quit

Les commandes E (*Exit*) et Q (*Quit*) sortent de XEDIT.

Syntaxes
E
Q

Valeurs par défaut
Aucune

Opération

Les commandes E et Q sont deux synonymes servant à sortir de l'éditeur. Aucun message n'est affiché, le curseur Basic revient à l'affichage, et l'état initial du drapeau 1 est restauré.

Exemples

E	sort de l'éditeur
Q	sort de l'éditeur

Commande Help

La commande H (*Help*) affiche la syntaxe d'une ou de plusieurs commandes.

Syntaxe
H [*commande*]

Valeurs par défaut
commande = toutes les commandes

Opération

Si vous spécifiez une commande derrière H, XEDIT affiche la syntaxe de la commande spécifiée. Toute pression sur une touche ramène l'affichage en mode *commandes*.

Si vous ne spécifiez aucune commande, XEDIT affiche la syntaxe de toutes les commandes par ordre alphabétique. Vous passez d'une commande à l'autre par pression sur les touches de curseur verticales (`[↓]`, `[↑]`, `[g][↓]` et `[g][↑]`). Toute autre touche ramène l'affichage en mode *commandes*.

Exemples

H	affiche la syntaxe de toutes les commandes, une par une
H H	affiche la syntaxe de la commande H

Commandes Text et Insert

Les commandes T (*Text*) et I (*Insert*) rentrent en mode *saisie*).

Syntaxes
[*ligne*] T
[*ligne*] I

Valeurs par défaut
ligne = ligne courante

Opération

La commande T est la commande d'introduction de texte. Lorsque vous entrez en mode *saisie* avec la commande T, la ligne courante est affichée, et vous pouvez la modifier. Lorsque vous appuyez sur la touche `[ENDLINE]`, cette ligne est introduite dans le fichier.

La commande I est la commande d'insertion de lignes de texte. Lorsque vous entrez en mode *saisie* avec la commande I, l'indicateur 1 s'allume pour signaler que vous êtes en mode insertion, et la ligne courante est affichée. Lorsque vous taperez du texte et que vous appuierez sur la touche `[ENDLINE]`, le texte sera inséré dans le fichier juste avant la ligne que vous avez vue.

Lorsque vous êtes en mode *saisie*, vous pouvez vous déplacer avec les touches de curseur horizontales sur la même ligne, ou sur une autre ligne avec les touches verticales.

Pour sortir du mode *saisie*, appuyez sur la touche [ATTN].

Exemples

`T`	rentre en mode *saisie* sur la ligne courante
`$I`	rentre en mode *saisie* avant la dernière ligne
`/toto/T`	rentre en mode *saisie* sur la prochaine occurrence de `toto`

Commandes List et Print

Les commandes L (*List*) et P (*Print*) listent les lignes sur l'appareil de visualisation ou d'impression.

Syntaxes

[*ligne début* [*ligne fin*]] L [*nombre de lignes*] [N]
[*ligne début* [*ligne fin*]] P [*nombre de lignes*] [N]

Valeurs par défaut

ligne début = ligne courante
ligne fin = dernière ligne

Opération

Les commandes L et P sont très similaires (autant que `LIST` et `PLIST` en Basic).

Après le listage, la ligne courante devient la dernière ligne listée plus une. Le *nombre de lignes* est optionnel (et est incompatible avec *ligne fin*). Il indique le nombre de lignes à lister. Le caractère N optionnel provoque l'apposition du numéro de ligne devant chaque ligne listée.

Exemples

`3,5L`	liste les lignes 3 à 5 et la ligne 6 devient la ligne courante
`LN`	liste toutes les lignes avec leur numéro jusqu'à la fin
`.+3,$-5LN`	liste avec numéros de lignes
`5L10N`	liste 10 lignes à partir de la ligne 5

Commande Delete

La commande D (*Delete*) détruit des lignes dans le fichier.

Syntaxe

[*ligne début* [*ligne fin*]] D [*fichier* [+]]

Valeurs par défaut

ligne début = ligne courante
ligne fin = *ligne début*

Opération

La commande D détruit une ou plusieurs lignes. Si plus d'une ligne doit être détruite, confirmation est demandée avec le message `Ok to Delete? Y/N:`. Toute pression sur une touche autre que [Y] annule l'opération.

Il est possible de spécifier que les lignes détruites doivent être placées dans le fichier *fichier*. Si celui-ci n'existe pas, il est créé. S'il existe, une erreur est générée. Si vous voulez ajouter les lignes détruites à un fichier existant, faites suivre son nom par +.

Exemples

`D`	détruit la ligne courante
`$-5,$D`	détruit les 6 dernières lignes du fichier
`D EXEMPLE1`	détruit la ligne courante et la met dans le fichier EXEMPLE1
`D EXEMPLE2 +`	détruit la ligne courante et l'ajoute au fichier EXEMPLE2

Commandes Copy et Move

Les commandes C (*Copy*) et M (*Move*) copient ou déplacent une ou plusieurs lignes du fichier.

Syntaxes

[*ligne début* [*ligne fin*]] C [*fichier*]
[*ligne début* [*ligne fin*]] M [*fichier*]

Valeurs par défaut (pas de *fichier* spécifié)

ligne début = ligne courante
ligne fin = *ligne début*

Valeurs par défaut (*fichier* spécifié)

ligne début = première ligne du fichier spécifié
ligne fin = dernière ligne du fichier spécifié

Opération

La commande C copie une plusieurs lignes et les insère avant la ligne courante. La commande M est similaire à la commande C, mais détruit les lignes originales.

Si aucun fichier n'est spécifié, les lignes insérées viennent du fichier édité. Si un fichier est spécifié et qu'il existe, ces lignes viendront de ce fichier.

Il faut un minimum de 256 octets disponibles pour déplacer des lignes de texte. De manière générale, plus grande est la mémoire disponible et plus rapide est le déplacement.

Attention à l'utilisation des expressions (`.`, `$`, recherches) dans les paramètres de ces commandes lorsqu'elles sont utilisées avec un fichier externe : les numéros de lignes font références au fichier externe, et les expressions sont évaluées avec le fichier courant. Pour cette raison, n'utilisez que des nombres pour vos numéros de lignes.

Exemples

`C`	duplique la ligne courante
`5,7C`	duplique les lignes 5 à 7 et les insère avant la ligne courante
`5,$-1M`	déplace les lignes 5 à l'avant-dernière ligne du fichier
`5,7C EXEMPLE1`	copie les lignes 5 à 7 du fichier EXEMPLE1 devant la ligne courante
`20M EXEMPLE2`	copie les lignes 20 jusqu'à la fin de EXEMPLE2 et les insère dans le fichier courant

Commande Join

La commande J (*Join*) réunit deux (ou plus) lignes du fichier en une seule.

Syntaxe

[*ligne début* [*ligne fin*]] [?] J [*nombre de lignes*]

Valeurs par défaut

ligne début = ligne courante
ligne fin = *ligne début* + 1

Opération

La commande J réunit les lignes spécifiées en une seule. Attention toutefois : le HP-71 ne permet pas l'édition de lignes supérieures à 96 caractères !

Un espace est ajouté avant chaque ligne jointe.

Si vous spécifiez un *nombre de lignes*, vous ne pouvez pas spécifier de *ligne fin*.

Vous pouvez contrôler finement la réunion des lignes en utilisant le caractère ? avant J. Chaque tentative de réunion provoquera l'affichage du numéro de ligne, d'un `:`, du numéro de colonne auquel s'effectue la réunion, d'un \, de la ligne au point précis de la réunion, d'un / et d'un point d'interrogation indiquant que XEDIT attend une réponse.
Vous pouvez alors répondre par :
- `[Y]` pour accepter la réunion et passer éventuellement à la suivante,
- `[N]` ou `[Q]` pour refuser la réunion et revenir en mode *commandes*.

Pour faire l'opération inverse de la commande J, c'est à dire séparer une ligne en deux, vous pouvez copier la ligne courante par C, puis éditer la ligne courante avec T, détruire du point de césure jusqu'à la fin, puis descendre d'une ligne et détruire jusqu'au point de césure.

Exemples

`J`	réunit la ligne courante et la suivante en une seule
`5,7J`	réunit les lignes 5 à 7 en une seule ligne 5
`J5`	réunit 5 lignes à partir de la ligne courante
`.,/toto/?J`	réunit les lignes spécifiées en demandant confirmation

Commande Search

La commande S (*Search*) cherche un motif dans le fichier.

Syntaxe

[*ligne début* [*ligne fin*]] [?] S/ *motif* [/]

Valeurs par défaut

ligne début = ligne courante + 1
ligne fin = dernière ligne

Opération

La commande S cherche dans le fichier le motif spécifié à l'aide d'une chaîne générique (voir le paragraphe *chaînes génériques de recherche*). Si le motif est trouvé, la ligne trouvée devient la ligne courante.

Le caractère délimiteur de fin est optionnel.

Si vous désirez chercher une chaîne contenant un caractère /, spécifiez un autre caractère délimiteur après le S. Tout caractère non blanc peut convenir.

Vous pouvez contrôler finement la recherche en utilisant le caractère ? avant S. Chaque occurrence trouvée provoquera l'affichage du numéro de ligne, d'un `:`, du numéro de colonne trouvé, d'un \, de la partie de la ligne contenant l'occurrence, d'un / et d'un point d'interrogation indiquant que XEDIT attend une réponse.
Vous pouvez alors répondre par :
- `[Y]` pour accepter l'occurrence trouvée et revenir en mode commande avec cette ligne comme ligne courante,
- `[N]` pour refuser cette occurrence et continuer la recherche, en affichant un message d'erreur si toutes les occurrences ont été passées,
- `[Q]` pour quitter la recherche et revenir en mode *commandes*.

Note : pour effectuer une recherche simple, il est plus facile de taper directement */motif* puis `[ENDLINE]` que d'utiliser la commande S, réservée pour les utilisations « lourdes ».

Exemples

`1S/toto`	recherche la chaîne `toto` dans tout le fichier
`1,$-10S/^$`	recherche une ligne vide avant la dixième avant la fin
`?S:1/4:`	recherche `1/4` dans le fichier (caractère délimiteur = `:`)

Commande Replace

La commande R (*Replace*) remplace une chaîne par une autre dans le fichier.

Syntaxe

[*ligne début* [*ligne fin*]] [?] R/ *motif* / *remplacement* [/]

Valeurs par défaut

ligne début = ligne courante
ligne fin = *ligne début*

Opération

La commande R cherche dans le fichier le motif spécifié à l'aide de chaîne génériques (voir le paragraphe *chaînes génériques de recherche*). Si le motif est trouvé, l'occurrence trouvée est remplacée par la chaîne de remplacement (voir le paragraphe *chaînes génériques de remplacement*).

Le caractère délimiteur de fin est optionnel.

Si vous désirez chercher ou remplacer une chaîne contenant un caractère /, spécifiez un autre caractère délimiteur après le R. Tout caractère non blanc peut convenir.

Vous pouvez contrôler finement le remplacement en utilisant le caractère ? avant R. Chaque occurrence trouvée provoquera l'affichage du numéro de ligne, d'un `:`, du numéro de colonne trouvé, d'un \, de la partie de la ligne contenant l'occurrence substituée par *remplacement*, d'un / et d'un point d'interrogation indiquant que XEDIT attend une réponse.
Vous pouvez alors répondre par :
- `[Y]` pour accepter le remplacement et passer éventuellement au suivant,
- `[N]` pour refuser ce remplacement et continuer la recherche,
- `[Q]` pour quitter la recherche et revenir en mode *commandes*.

Exemples

`R/toto/tata/`	remplace `toto` par `tata` sur la ligne courante
`.,$R/[A-Z]/&&`	remplace les majuscules par deux majuscules jusqu'à la fin du fichier
`Rx/x\\x`	remplace tous les / par des \ sur la ligne courante
`?R/a/b`	remplace avec confirmation tous les a par des b

Commande eXchange

La commande X (*eXchange file*) édite un autre fichier sans sortir de XEDIT.

Syntaxe

X *fichier*

Valeurs par défaut

Aucune

Opération

La commande X édite un autre fichier sans sortir de XEDIT. Si le fichier nommé n'existe pas, il est créé. S'il existe, il est simplement cherché en mémoire.

Exemples

`X EXEMPLE`	édite le fichier EXEMPLE sans sortir de XEDIT

Références

JPC ?? (page ??) Première version de XEDIT par Pierre David et Janick Taillandier.

Manuel du module Editeur de Textes par Hewlett-Packard.

Manuel du module Forth / Assembleur par Hewlett-Packard.

Unix User's Reference Manual par AT&T.

Mots-clés associés

FILEPOS, FIND, GENLEN, GENPOS, GENRPLC, TEDIT

Auteurs

Pierre David et Janick Taillandier.

FILEPOS

FILEPOS (FILE POSition) cherche une chaîne générique dans un fichier Texte.

O Ordre	■ Exécution au clavier
■ Fonction	O Mode CALC
O Opérateur	■ IF..THEN..ELSE
	■ Opération d'unité

```
FILEPOS ( fichier , motif )
FILEPOS ( fichier , motif , début )
FILEPOS ( fichier , motif , début , fin )
FILEPOS ( fichier , motif , début , fin , colonne )
FILEPOS ( canal , motif )
FILEPOS ( canal , motif , début )
FILEPOS ( canal , motif , début , fin )
FILEPOS ( canal , motif , début , fin , colonne )
```

Exemples

```
DISP FILEPOS ("ESSAI","JPC Rom")
```

Cherche la chaîne "JPC Rom" dans le fichier ESSAI et retourne le numéro de l'enregistrement trouvé, ou -1 si la recherche a échoué.

```
L=FILEPOS (5,"^[0-9]",25,50)
```

Recherche dans le fichier correspondant au canal #5, la première occurrence d'un chiffre en début d'enregistrement, entre les lignes 25 et 50.

Paramètres d'entrée

Elément	Description	Restrictions
fichier	Expression alphanumérique.	Le fichier doit être en mémoire vive.
canal	Expression numérique arrondie à un entier.	1 à 255, et le fichier doit être en mémoire vive.
motif	Expression alphanumérique.	Expression générique valide.
début	Expression numérique arrondie à un entier. Défaut : 0	0 à 1048575.
fin	Expression numérique arrondie à un entier. Défaut : dernier enregistrement du fichier.	début à 1048575.
colonne	Expression numérique arrondie à un entier. Défaut : 1	1 à 65535.

Opération

FILEPOS (suite)

FILEPOS recherche la première occurrence de *motif* dans le fichier spécifié. Si une occurrence est trouvée, le numéro de l'enregistrement est retourné. Si aucune occurrence n'est trouvée, la valeur -1 est renvoyée. Les paramètres *début* et *fin* permettent de restreindre la recherche dans une partie du fichier.

Attention : les paramètres *début* et *fin* de FILEPOS sont exprimés en *numéros d'enregistrement*. L'enregistrement 0 correspond à la ligne 1 du fichier, de manière à être compatible avec les ordres standard du HP-71 (READ #, RESTORE #, etc.).

La recherche tient compte des caractères génériques étendus (voir XEDIT pour plus de précision sur les caractères génériques). Le tableau ci-dessous résume les caractères spéciaux :

Caractère	Signification
\	annule la signification du caractère suivant
^	début de ligne
$	fin de ligne
.	caractère quelconque
[ ]	ensemble de caractères
[^]	complémentaire de l'ensemble
*	répétition du motif précédent 0 ou n fois

Le paramètre *colonne* spécifie le numéro de la colonne à partir de laquelle la recherche commence. La recherche commencera à la ligne et à la colonne spécifiées, et continuera sur les lignes suivantes à partir de la première colonne. Ceci est utile lorsque vous désirez reprendre une recherche sur une ligne alors qu'une occurrence a déjà été trouvée. Le paramètre *colonne* permet alors de « sauter » cette occurrence, et de reprendre la recherche après.

Note : les modules *Forth / Assembler* et *Text Editor* offrent la fonction SEARCH. La fonctionnalité de FILEPOS est équivalente, mais se distingue par trois points essentiels :
- la recherche utilise les chaînes génériques étendues (celles de XEDIT et pas celles de TEDIT),
- la syntaxe est bien plus simple, claire et mnémotechnique, et enfin,
- la valeur de retour est, elle aussi, plus simple.

Si vous désirez chercher une chaîne dans un fichier, et que cette chaîne est susceptible de contenir des caractères spéciaux, il faut annulerla signification de ces caractères. Il suffit de les faire précéder par un \, ce qui donne la commande suivante :

```
FILEPOS (fichier, GENRPLC$ (motif, "[\\\^\$\.\[\]\*]", "\\&"))
```

Références

JPC ?? (page ??) Première version de FILEPOS par Pierre David et Janick Taillandier.

Unix User's Reference Manual par AT&T.

Mots-clés associés

ASSIGN #, CREATE, FIND, GENLEN, GENPOS, GENRPLC, PRINT #, READ #, XEDIT

Auteurs

Pierre David et Janick Taillandier.

GENLEN

GENLEN (GENeric LENgth) cherche une chaîne générique dans une chaîne alphanumérique et retourne la longueur de l'occurrence trouvée.

O Ordre	■ Exécution au clavier
■ Fonction	O Mode CALC
O Opérateur	■ IF...THEN...ELSE
	■ Opération d'unité

```
GENLEN ( chaîne , motif )
GENLEN ( chaîne , motif , début )
```

Exemples

```
DISP GENLEN ("JPC Rom", "Rom")
```

Affiche 3, c'est à dire la longueur de la chaîne "Rom" à l'intérieur de la chaîne "JPC Rom".

```
A=GENLEN (L$,"[0-9][0-9]*",X+1)
```

Place dans la variable A la longueur du premier nombre trouvé dans la chaîne L$ à partir du caractère X+1.

Paramètres d'entrée

Elément	Description	Restrictions
chaîne	Expression alphanumérique.	Aucune.
motif	Expression alphanumérique.	Expression générique valide.
début	Expression numérique arrondie à un entier. Défaut : 0	0 à 1048575

Opération

La fonction GENLEN recherche la chaîne générique *motif* dans la chaîne *chaîne*. Si une occurrence est trouvée, sa longueur (éventuellement nulle) est renvoyée, sinon la valeur 0 est renvoyée.

Attention : la valeur 0 est ambiguë, car elle peut indiquer à la fois une occurrence de taille nulle (ce qui peut arriver lorsque vous utilisez le caractère générique * par exemple), et à la fois une occurrence non trouvée. Pour lever l'ambiguïté, nous vous conseillons de tester au préalable la présence de l'occurrence avec GENPOS, puis de n'utiliser GENLEN que lorsque vous êtes sûr qu'il y a une occurrence. Cette ambiguïté est préférable, car ainsi les deux fonctions sœurs GENPOS et GENLEN renvoie la même valeur en cas d'occurrence non trouvée, de même que la fonction standard POS.

Les caractères génériques utilisables dans la chaîne *motif* sont résumés dans le tableau ci-dessous. Pour plus de détails, voir XEDIT.

GENLEN (suite)

Caractère	Signification
\	annule la signification du caractère suivant
^	début de ligne
$	fin de ligne
.	caractère quelconque
[ ]	ensemble de caractères
[^]	complémentaire de l'ensemble
*	répétition du motif précédent 0 ou n fois

Si le paramètre numérique *début* est présent, il spécifie à partir de quel caractère de *chaîne* la recherche doit commencer. Par défaut, *motif* est cherché dans toute la chaîne *chaîne*. Attention : le caractère générique ^ (début de chaîne) correspond au début de la chaîne et non à la position indiquée par *début*. Ainsi, la commande `GENLEN("ABCD","^B",2)` retournera 0.

Références

JPC ?? (page ??) Première version de GENPOS par Pierre David et Janick Taillandier.

Unix User's Reference Manual par AT&T.

Mots-clés associés

XEDIT, FILEPOS, FIND, GENPOS, GENRPLC, LEN

Auteurs

Pierre David et Janick Taillandier.

GENPOS

GENPOS (GENeric POSition) cherche une chaîne générique dans une chaîne alphanumérique et retourne la position de l'occurrence trouvée.

O Ordre	■ Exécution au clavier
■ Fonction	O Mode CALC
O Opérateur	■ IF...THEN...ELSE
	■ Opération d'unité

```
GENPOS ( chaîne , motif )
GENPOS ( chaîne , motif , début )
```

Exemples

```
DISP GENPOS ("JPC Rom", "Rom")
```

Affiche 4, c'est à dire la position de la chaîne "Rom" à l'intérieur de la chaîne "JPC Rom".

```
A=GENPOS (L$,"[0-9]",X+1)
```

Place dans la variable A la position du premier chiffre dans la chaîne L$ à partir du caractère X+1.

Paramètres d'entrée

Elément	Description	Restrictions
chaîne	Expression alphanumérique.	Aucune.
motif	Expression alphanumérique.	Expression générique valide.
début	Expression numérique arrondie à un entier. Défaut : 0	0 à 1048575

Opération

La fonction GENPOS recherche la chaîne générique *motif* dans la chaîne *chaîne*. Si une occurrence est trouvée, sa position dans la chaîne *chaîne* est renvoyée, sinon la valeur 0 est renvoyée.

Les caractères génériques utilisables dans la chaîne *motif* sont résumés dans le tableau ci-dessous. Pour plus de détails, voir XEDIT.

```
Caractère       Signification
---------       ---------------------------------------------
    \           annule la signification du caractère suivant
    ^           début de ligne
    $           fin de ligne
    .           caractère quelconque
   [ ]          ensemble de caractères
   [^ ]         complémentaire de l'ensemble
    *           répétition du motif précédent 0 ou n fois
```


GENPOS (suite)

Si le paramètre numérique *début* est présent, il spécifie à partir de quel caractère de *chaîne* la recherche doit commencer. Par défaut, *motif* est cherché dans toute la chaîne *chaîne*. Attention : le caractère générique ^ (début de chaîne) correspond au début de la chaîne et non à la position indiquée par *début*. Ainsi, la commande `GENPOS("ABCD","^B",2) retournera 0.`

Références

JPC ?? (page ??) Première version de GENPOS par Pierre David et Janick Taillandier.

Unix User's Reference Manual par AT&T.

Mots-clés associés

XEDIT, FILEPOS, FIND, GENLEN, GENRPLC, POS

Auteurs

Pierre David et Janick Taillandier.

GENRPLC$

GENRPLC$ (GENeric RePLaCe) cherche un motif générique dans une chaîne alphanumérique, et substitue la sous-chaîne trouvée par la chaîne de remplacement.

○ Ordre	■ Exécution au clavier
■ Fonction	○ Mode CALC
○ Opérateur	■ IF...THEN...ELSE
	■ Opération d'unité

```
GENRPLC$ ( chaîne , motif , remplacement )
GENRPLC$ ( chaîne , motif , remplacement , début )
```

Exemples

```
DISP GENRPLC$ ("Math Rom", "Math", "JPC")
```
Affiche la chaîne "JPC Rom", c'est à dire la chaîne d'origine avec toutes les occurrences de "Math" remplacées par "JPC".

```
A$=GENRPLC$ (L$,"[0-9][0-9]*","-&",X+1)
```
Place dans la variable A$ la chaîne origine avec tous les nombres rendus positifs : un signe "-" est placé devant chaque nombre. La chaîne de remplacement signifie : un signe "-" et l'occurrence trouvée.

Paramètres d'entrée

Elément	Description	Restrictions
chaîne	Expression alphanumérique.	Aucune.
motif	Expression alphanumérique.	Expression générique valide.
remplacement	Expression alphanumérique.	Aucune.
début	Expression numérique arrondie à un entier. Défaut : 0	0 à 1048575

Opération

La fonction GENRPLC$ substitue, dans la chaîne *chaîne*, toutes les occurrences de la chaîne *motif* par la chaîne *remplacement*.

La fonction GENRPLC$ est similaire à la fonction REPLACE$. La fonction REPLACE$ est, elle-même, le regroupement de deux fonctions plus anciennes, REPLACE$ et RPLC$. Cette fonction REPLACE$ n'est laissée dans JPC Rom que pour des raisons de compatibilité ascendante, et vous êtes plutôt encouragés à utiliser GENRPLC$. Le tableau ci-dessous résume les différences entre ces trois fonctions :

GENRPLC$ (suite)

Fonction	Ancien nom	Caractéristiques
REPLACE$	REPLACE$	Remplacement simple avec caractère joker
REPLACE$	RPLC$	Chaînes génériques compatibles avec EDTEXT
GENRPLC$	GENRPLC$	Chaînes génériques compatibles avec XEDIT

Note : l'ancienne fonction REPLACE$ correspond au cas où l'actuelle fonction REPLACE$ est appelée avec un quatrième paramètre alphanumérique représentant le joker. Dans les autres cas d'appel, c'est l'ancienne fonction RPLC$ qui est exécutée.

Les chaînes *motif* et *remplacement* peuvent contenir des caractères génériques. Le tableau ci-dessous résume ces caractères. Pour une explication détaillée, voir XEDIT.

Les caractères génériques de la chaîne *motif* sont :

Caractère	Signification
\	annule la signification du caractère suivant
^	début de ligne
$	fin de ligne
.	caractère quelconque
[ ]	ensemble de caractères
[^]	complémentaire de l'ensemble
*	répétition du motif précédent 0 ou n fois

Dans la chaîne *remplacement*, les caractères génériques sont :

Caractère	Signification
\	annule la signification du caractère suivant
&	substitue l'occurrence trouvée

Si le paramètre numérique *début* est présent, il spécifie à partir de quel caractère de *chaîne* la recherche et la substitution doivent commencer. Par défaut, *motif* est cherché dans toute la chaîne *chaîne*. Attention : le caractère générique ^ (début de chaîne) correspond au début de la chaîne et non à la position indiquée par *début*. Ainsi, la commande GENRPLC$("ABCD","^B","X",2) retournera la chaîne "ABCD".

Si vous désirez remplacer une chaîne susceptible de contenir des caractères spéciaux dont vous voulez annuler la signification, c'est à dire si vous désirez opérer un remplacement non générique, vous ferez :

```
GENRPLC$ (chaîne, GENRPLC$ (motif, "[\\\^\$\.\[\]\*]", "\\&"),
                  GENRPLC$ (remplacement, "[\\\&]","\\&"))
```

Références

JPC ?? (page ??) Première version de GENRPLC$ par Pierre David et Janick Taillandier.

Unix User's Reference Manual par AT&T.

Mots-clés associés

XEDIT, GENLEN, GENPOS, REPLACE$

```
Page 001             Module principal <main.as>
AREUH ASS. V2.4

00000                 TITLE   Module principal <main.as>
00000
00000            *
00000            * Allocation du bit restant dans le ram-réservée
00000            * A mettre dans def.as lors de l'intégration dans JPC Rom
00000            *
00000            =RESJPC EQU     #2F991
00000            =INEDIT EQU     %1000   bit de T/XEDIT
00000            =ASFMSK EQU     %0010
00000
00000            *
00000            * Version A : 88/../.. -> 88/10/16
00000            * - distribution à Buitenhuis
00000            * - L, P, I, T, S, R marchent convenablement
00000            * - Le mode HP-UX n'a plus besoin de "," pour délimiter les
00000            *   deux paramètres
00000            * Version A1 : 88/10/22 -> 88/10/..
00000            * - distribution à Laurent Istria
00000            * - correction de l'accès au message "Y/N/Q"
00000            * - le flag 1 signale quand on est en insertion
00000            * - dernier bit de la Reserved Ram utilisé par T/XEDIT
00000            * Version A2 : 88/10/.. -> 88/11/05
00000            * - distribution à la réunion PPC de Novembre 1988
00000            * - les commandes J et H ont été écrites
00000            * - les commandes sont accessibles par un message
00000            * - suppression du buffer en cas d'erreur "chaine générique
00000            *   invalide"
00000            * - la commande "240" posait un problème (ne reconnaissait
00000            *   pas EOL)
00000            * - D fonctionne en interne (i.e. sans fichier externe)
00000            * - J et R n'acceptent plus les lignes > 64 K
00000            * Version A3 : 88/11/05 -> 88/11/13
00000            * - distribution à Jean-Jacques Dhénin
00000            * - D fonctionne en interne et externe
00000            * - correction des accès aux fichiers en Rom ou Secure
00000            * - amélioration des traitements d'erreur
00000            * - la commande X a été écrite
00000            * - les commandes C et M fonctionnent !
00000            * - utilisation de l'id #E1
00000            * - traduction des touches avec "KEYWT"
00000            * - introduction du message "Insufficient Memory"
00000            * - écriture du traducteur de messages.
00000            * Version A4 : 88/11/13 -> 88/11/20
00000            * - distribution à Laurent Istria, Jean-Jacques Dhénin,
00000            *   Eric Gengoux
00000            * - il n'y a plus de sortie brutale sur "eMEM"
00000            * - visualisation de la ligne courante avec [ENDLINE]
00000            * Version A5 : 88/11/20 -> 88/11/26
00000            * - distribution à Jean-Jacques Dhénin
00000            * - correction de ...C (début dest = début source)
```

```
Page 002             Module principal <main.as>
AREUH ASS. V2.4

00000            * - ajout du nb total de lignes dans [f][CONT]
00000            * Version A6 : 88/11/27 -> 89/06/10
00000            * - distribution à
00000            * Version A7 : 89/06/1? -> 89/06/17 (10128 octets)
00000            * - réécriture et intégration de FIND
00000            * - fin du codage de FILEPOS
00000            * - retrait des deux paramètres (largeur/hauteur) de XEDIT
00000            * - retrait de FILERPLC
00000            * - changement du test d'écriture dans le fichier
00000            * - changement dans les sources (main de find intégré ici)
00000            * - séparation de xutil.as et jpcutil.as
00000            * - séparation des messages en deux fichiers (FIND / EDIT)
00000            * - intégration dans JPC Rom
00000            *
00000
00000 54443445          NIBASC  'EDIT    '
00008 02020202
00010 802E              NIBHEX  802E
00014 00                CON(2)  0
00016 0000              CON(4)  0
0001A 000000            CON(6)  0
00020 00000             REL(5)  =FileNd
00025
00025           =id     EQU     #E1
00025            t      EQU     112
00025
00025 1E                CON(2)  =id
00027 07                CON(2)  t       Lowest Token
00029 57                CON(2)  5+t     Highest Token
0002B 37200             REL(5)  find    Next Lex
00030 F                 NIBHEX  F
00031 4400              REL(4)  1+TxTbSt
00035 0000              REL(4)  =MSGTBL1 Offset to Message Table
00039 70100             REL(5)  POLHND  Poll
0003E
0003E 000               CON(3)  (TxEn01)-(TxTbSt)
00041 00000             REL(5)  =FILEPOSe
00046 F                 NIBHEX  F
00047
00047 110               CON(3)  (TxEn02)-(TxTbSt)
0004A 00000             REL(5)  =GENLENe
0004F F                 NIBHEX  F
00050
00050 020               CON(3)  (TxEn03)-(TxTbSt)
00053 00000             REL(5)  =GENPOSe
00058 F                 NIBHEX  F
00059
00059 F20               CON(3)  (TxEn04)-(TxTbSt)
0005C 00000             REL(5)  =GENRPLCe
00061 F                 NIBHEX  F
```

```
Page 003             Module principal <main.as>
AREUH ASS. V2.4

00062
00062 240               CON(3)  (TxEn05)-(TxTbSt)
00065 00000             REL(5)  =TEDITe
0006A D                 NIBHEX  D
0006B
0006B F40               CON(3)  (TxEn06)-(TxTbSt)
0006E 00000             REL(5)  =XEDITe
00073 D                 NIBHEX  D
00074
00074           TxTbSt
00074 D         TxEn01  CON(1)  13
00075 6494C454          NIBASC  'FILEPOS'
0007D 05F435
00083 07                CON(2)  0+t
00085 B         TxEn02  CON(1)  11
00086 7454E4C4          NIBASC  'GENLEN'
0008E 54E4
00092 17                CON(2)  1+t
00094 B         TxEn03  CON(1)  11
00095 7454E405          NIBASC  'GENPOS'
0009D F435
000A1 27                CON(2)  2+t
000A3 F         TxEn04  CON(1)  15
000A4 7454E425          NIBASC  'GENRPLC$'
000AC 05C43442
000B4 37                CON(2)  3+t
000B6 9         TxEn05  CON(1)  9
000B7 45544494          NIBASC  'TEDIT'
000BF 45
000C1 47                CON(2)  4+t
000C3 9         TxEn06  CON(1)  9
000C4 85544494          NIBASC  'XEDIT'
000CC 45
000CE 57                CON(2)  5+t
000D0 1FF               NIBHEX  1FF
000D3
000D3           =TEDITd
000D3 8F00000   =XEDITd GOSBVL  =FILDC*
000DA 8D00000           GOVLNG  =DROPDC
000E1
000E1 8F00000    numck  GOSBVL  =NUMCK
000E8 8D00000    resptr GOVLNG  =RESPTR
000EF
000EF           =TEDITp
000EF           =XEDITp
000EF           *
000EF           * Note : Lors de l'intégration dans JPC Rom, les parses de
000EF           * XEDIT et TEDIT sont les memes que celle de ADCREATE.
000EF           *
000EF 7D10              GOSUB   fichip
```

```
Page 004             Module principal <main.as>
AREUH ASS. V2.4

000F3
000F3 8F00000           GOSBVL =EOLCK
000FA 4DE               GOC    resptr
000FD 77EF              GOSUB  resptr
00101 7C10              GOSUB  virgup
00105 8F00000           GOSBVL =STRGCK
0010C 6BDF              GOTO   resptr
00110
00110 8F00000    fichip GOSBVL =FSPECp
00117 500               RTNNC
0011A 8D00000           GOVLNG =FSPECe
00121
00121 8F00000    virgup GOSBVL =NTOKEN
00128 8F00000           GOSBVL =COMCK+
0012F 400               RTNC             Cy = 1 ==> tCOMMA trouvé
00132 8D00000           GOVLNG =MSPARe
00139
00139            *
00139            * Fin de la partie extraite de ADCREATE
00139            *
00139
00139 8D00000    errp   GOVLNG  =SYNTXe
00140
00140
00140
00140
00140 969        POLHND ?B=0    B
00143 D0                GOYES   POL010
00145 3100              LC(2)   =pKYDF
00149 961               ?B=C    B
0014C 83                GOYES   hKYDEF
0014E 00                RTNSXM
00150 11B        POL010 C=R3
00153 135               D1=C
00156 112               A=R2
00159 1CD               D1=D1-  (VER$en)-(VER$st)-2
0015C 137               CD1EX
0015F 8B6               ?A>C    A
00162 C1                GOYES   POL050
00164 135               D1=C
00167 10B               R3=C
0016A 3D7314A3   VER$st LCASC   ' XED:A7'
00172 44548502
0017A 15DD       VER$en DAT1=C  (VER$en)-(VER$st)-2
0017E 00         POL050 RTNSXM
00180
00180            MNEMF  EQU     8
00180            MODF   EQU     15
00180            COMF   EQU     24
00180                   *
```

```
Page 005          Module principal <main.as>
AREUH ASS. V2.4

00180                ***************************************************
00180                * Auteur: Stephane Barizien 03/85 - Interception
00180                *         de pKYDF et champs assembleur.
00180                * 1ere modification: M.MARTINET et P.DAVID 10/85 -
00180                *       transformation en FIELD ON | OFF.
00180                * 2eme modification: M.MARTINET 12/85 - mise en
00180                *       place de la bascule touche CALC et suppres-
00180                *       sion token FIELD ON | OFF.
00180                * 3eme modification: J.Taillandier et P.David
00180                *       08/86, reunion de JPC-LEX
00180                ***************************************************
00180
00180 6A10            hASF1  GOTO   hASF
00184
00184 110             hKYDEF A=R0            A(A) = code logique
00187 3102                   LC(2)  32
0018B 962                    ?A=C   B
0018E 2F                     GOYES  hASF1
00190                *       LC(2)  =kcVIEW
00190                *       ?A=C   B
00190                *       GOYES  hCURS_1
00190                *       LC(2)  =kcLFT
00190                *       ?A=C   B
00190                *       GOYES  hCALC_1
00190 3100                   LC(2)  =kcCALC
00194 962                    ?A=C   B
00197 9E                     GOYES  hASF1
00199 00                     RTNSXM
0019B
0019B                 hASF
0019B                *
0019B                * Est-on sous editeur ? Test modifie le 88/10/23 par PD/JT
0019B                *
0019B 1B199F2                D0=(5)  =RESJPC
001A2 1520                   A=DAT0  P
001A6 308                    LC(1)   =INEDIT
001A9 0E06                   A=A&C   P
001AD 908                    ?A=0    P
001B0 24                     GOYES   rtn
001B2                *
001B2                * R0(A) = touche pressee
001B2                *
001B2 110                    A=R0
001B5 3100                   LC(2)  =kcCALC  Est-ce
001B9 966                    ?A#C   B        la touche [f][CALC] ?
001BC 83                     GOYES  FXQ      Non: execution suite prgm.
001BE                * Touche [f][CALC] : inversion de l'etat du bit ASF
001BE 1A199F                 D0=(4) =RESJPC
001C4 1520                   A=DAT0 P
001C8 302                    LC(1)  =ASFMSK
```

```
Page 006          Module principal <main.as>
AREUH ASS. V2.4

001CB D8                     B=A    A
001CD 0EF2                   C=C&A  A        C(0) := ASMMSK ou 0
001D1 90A                    ?C=0   P
001D4 C0                     GOYES  asf10    Il faut le mettre à 1
001D6                * Il faut mettre le bit à 0
001D6 30D                    LC(1)  `=ASFMSK
001D9 0E06                   A=A&C  P        A(0) := bit ASF à 0
001DD 590                    GONC   asf20    B.E.T.
001E0                * Il faut mettre le bit à 1
001E0 302             asf10  LC(1)  =ASFMSK
001E3 0E0E                   A=A!C  P        A(0) := bit ASF à 1
[illegible]           asf20  DAT0=A P
001EB 840                    ST=0   0        Touche tombee, pas de definition
001EE 6AA0                   GOTO   rtnxm0   et fin d'execution
001F2
001F2 00              rtn    RTNSXM
001F4 1A199F          FXQ    D0=(4) =RESJPC  Execution du programme
001FA 15A0                   A=DAT0 1        Est-on en mode field on ?
001FE 302                    LC(1)  =ASFMSK
00201 0E06                   A=A&C  P
00205 908                    ?A=0   P
00208 AE                     GOYES  rtn      Non: fin de programme
0020A 1A0000                 D0=(4) =DSPBFS  Oui: continu
00210 32000                  LC(3)  (=DSPBFE)-(=DSPBFS)-1
00215 AB5                    B=C    X
00218 14A             SP05   A=DAT0 B
0021B 968                    ?A=0   B
0021E F1                     GOYES  SP08
00220 3102                   LCASC  ' '
00224 966                    ?A#C   B
00227 D0                     GOYES  SP06
00229 161                    D0=D0+ 2
0022C A3D                    B=B-1  X
0022F 58E                    GONC   SP05
00232 00              rtn1   RTNSXM
00234 31A2            SP06   LCASC  '*'      Y a-t-il des remarques ?
00238 962                    ?A=C   B
0023B 7B                     GOYES  rtn      Oui: fin de programme
0023D 1A0000          SP08   D0=(4) =CURSOR   Verification
00243 14A                    A=DAT0 B        position du curseur
00246 3170                   LC(2)  (MNEMF)-1 en fonction
0024A 9EA                    ?A<=C  B        des
0024D 41                     GOYES  SP10     differents
0024F 31E0                   LC(2)  (MODF)-1 champs:
00253 9EA                    ?A<=C  B        MNEMF: mnemo-field
00256 B0                     GOYES  SP10     MODF : modification-field
00258 3171                   LC(2)  (COMF)-1 COMF : comment-field
0025C 9E6                    ?A>C   B
0025F 3D                     GOYES  rtn1
00261 E2              SP10   C=C-A  A        Redefinition
```

```
Page 007          Module principal <main.as>
AREUH ASS. V2.4

00263 E6                     C=C+1  A        de la touche
00265 1A0000                 D0=(4) =DEFADR   espace
0026B 14C                    DAT0=C B        avec
0026E 161                    D0=D0+ 2        le
00271 304                    LCHEX  4        bon
00274 15C0                   DAT0=C 1        nombre
00278 7210                   GOSUB  SKPTBL   de
0027C 02020202               NIBASC '        '  CARCTERES BLANCS
00284 02020202
0028C 02                     NIBASC ' '
0028E 07              SKPTBL C=RSTK          Sortie de table
00290 160                    D0=D0+ 1
00293 144                    DAT0=C A
00296 850                    ST=1   0        ST[0]=1 => definit. touche
00299 821             rtnxm0 XM=0            ST[0]=0 => touche inhibee
0029C 03                     RTNCC           Puis interception du Poll.
0029E
0029E                 find
0029E
0029E                 tFIND  EQU    75
0029E
0029E 1E                     CON(2) =id
002A0 B4                     CON(2) tFIND
002A2 B4                     CON(2) tFIND
002A4 00000                  CON(5) 0        Next Lex
002A9
002A9 F                      NIBHEX F        Pas de speed table
002AA 7100                   REL(4) 1+FindTxTbSt
002AE 0000                   REL(4) =MSGTBL2 Offset to Message Table
002B2 00000                  CON(5) 0        Poll handler
002B7
002B7                *
002B7                * Main Table
002B7                *
002B7 000                    CON(3) (FindTxEn01)-(FindTxTbSt)
002BA 00000                  REL(5) =FINDe
002BF 1                      CON(1) #1       Stmt. Non programmable
002C0
002C0                *
002C0                * Text Table
002C0                *
002C0                 FindTxTbSt
002C0                 FindTxEn01
002C0 7                      CON(1) 7
002C1 6494E444               NIBASC 'FIND'
002C9 B4                     CON(2) tFIND
002CB
002CB 1FF                    NIBHEX 1FF
002CE
```

```
Page 008          Module principal <main.as>
AREUH ASS. V2.4   Parse

002CE
002CE                ***************************************************
002CE                * FINDp
002CE                *
002CE                * But: analyser et reconnaitre la syntaxe de FIND
002CE                * Syntaxe: FIND <chaine> [.<debut> [. <fin>]]
002CE                * Entrée:
002CE                *    - D1 = ^ chaîne Ascii
002CE                *    - D0 = ^ tokens
002CE                *    - D(A) = AVMEME
002CE                * Sortie:
002CE                *    [illegible]
002CE                * Abime: [illegible]
002CE                *    [illegible]
002CE                * Niveaux: 5 (STRGCK)
002CE                * Appelle: STRGCK, COMCK+, ck2li#, RESPTR
002CE                * Historique:
002CE                *    89/06/09: P.D. reconception & recodage
002CE                ***************************************************
002CE
002CE                 respTr
002CE 8D00000                GOVLNG =RESPTR
002D5
002D5                =FINDp
002D5 8F00000                GOSBVL =STRGCK
002DC                *
002DC                * A = token de ce qui suit l'expression
002DC                * D1 = ^ derrière le texte correspondant à A
002DC                *
002DC 8F00000                GOSBVL =COMCK+
002E3 5AE                    GONC   respTr   Pas de ',' : c'est fini !
002E6 181                    D0=D0- 2        ck2li# sortira une virgule
002E9                *
002E9                * Une virgule a été reconnue, donc il faut un ou
002E9                * deux numéros de ligne.
002E9                * C'est le rôle de ck2li#, empruntée à JPC Rom et
002E9                * plus précisément à D/PBLIST/RENUMREM
002E9                *
002E9
002E9                *
002E9                * ATTENTION ! LE CODE CONTINUE !
002E9                *
002E9
002E9                ***************************************************
002E9                * ck2li#
002E9                *
002E9                * But : parse   "  <line#1> [ . <line#2> ]  "
002E9                * Entrée :
002E9                *    - D1 = ^ blancs optionnels avant
002E9                * Sortie :
```

```
Page 009              Module principal  main.as
AREUH ASS. V2.4       Parse

0392 002E9            *   - S9 = 1 si les deux <line#> ont ete reconnus
0393 002E9            * Appelle : ck1li#, COMCK, NTOKNL, OUT3TK
0394 002E9            * Niveaux : 4 (NTOKNL)
0395 002E9            * Utilise : A-C, P, D0, D1, R0, S0-S3, S11
0396 002E9            * Historique :
0397 002E9            *   86/05/..: JPB     reecriture d'apres I.D.S.
0398 002E9            *   88/01/10: FD & JT separation & documentation
0399 002E9            ************************************************
0400 002E9
0401 002E9 849         ck2li# ST=0    9
0402 002EC 7D00               GOSUB   ck1li#    <line1>
0403 002F0 8F00000            GOSBVL  =COMCK    Comma ?
0404 002F7 56D                GONC    respTr    Non : sortir
0405 002FA 859                ST=1    9         On reconnait les 2 <line#>
0406 002FD
0407 002FD 8F00000     ck1li# GOSBVL  =NTOKNL   Idem NTOKEN, admet line#
0408 00304 3100               LC(2)   =tLINE#
0409 00308 966                ?A#C    B
0410 0030B 01                 GOYES   LSTPE     Pas <line#> : erreur
0411 0030D 3100               LC(2)   =tCOMMA
0412 00311 AEA                A=C     B
0413 00314 8D00000            GOVLNG  =OUT3TK
0414 0031B
0415 0031B 8D00000     LSTPE  GOVLNG  =IVPARe   INVALID (MISSING PARM)
0416 00322
0417 00322                    END
```

```
Page 010              Module principal <main.as>
AREUH ASS. V2.4       **** SYMBOL TABLE ****

=ASFMSK        00002 Abs  0009 -  0242 0248 0252 0261
=COMCK         Extrn Ukn       -  0403
=COMCK*        Extrn Ukn       -  0157 0371
=CURSOR        Extrn Ukn       -  0281
 COMF          00013 Abs  0191 -  0289
=DEFADR        Extrn Ukn       -  0294
=DROPDC        Extrn Ukn       -  0132
=DSPBFE        Extrn Ukn       -  0266
=DSPBFS        Extrn Ukn       -  0265 0266
=EOLCK         Extrn Ukn       -  0145
=FILDC*        Extrn Ukn       -  0131
=FILEPOSe      Extrn Ukn       -  0036
=FINDe         Extrn Ukn       -  0327
=FINDp         00205 Rel  0365 -
=FSPECe        Extrn Ukn       -  0154
=FSPECp        Extrn Ukn       -  0152
=FiLeNd        Extrn Ukn       -  0371
 FxQ           001F4 Rel  0259 -  0233
 FiLeNd        00322 Rel  0417 -
 FindTxEn01    002C0 Rel  0334 -  0328
 FindTxTbSt    002C0 Rel  0333 -  0319 0328
=GENLENe       Extrn Ukn       -  0090
=GENPOSe       Extrn Ukn       -  0094
=GENRPLCe      Extrn Ukn       -  0098
=INEDIT        00008 Abs  0008 -  0220
=IVPARe        Extrn Ukn       -  0415
 LSTPE         0031B Rel  0415 -  0410
=MSGTBL1       Extrn Ukn       -  0082
=MSGTBL2       Extrn Ukn       -  0320
=MSPARe        Extrn Ukn       -  0159
 MNEMF         00008 Abs  0189 -  0283
 MODF          0000F Abs  0190 -  0286
=NTOKEN        Extrn Ukn       -  0156
=NTOKNL        Extrn Ukn       -  0407
=NUMCK         Extrn Ukn       -  0134
=OUT3TK        Extrn Ukn       -  0413
 POL010        00150 Rel  0176 -  0171
 POL050        0017E Rel  0187 -  0182
 POLHND        00140 Rel  0170 -  0083
=PESJPC        2F991 Abs  0007 -  0226 0240 0259
=RESPTR        Extrn Ukn       -  0135 0363
=STRGCK        Extrn Ukn       -  0149 0366
=SYNTXe        Extrn Ukn       -  0165
 SKPTBL        0028E Rel  0302 -  0299
 SP05          00218 Rel  0268 -  0276
 SP06          00234 Rel  0273 -  0273
 SP08          0023D Rel  0281 -  0270
 SP10          00261 Rel  0292 -  0285 0288
=TEDITd        000D3 Rel  0130 -
=TEDITe        Extrn Ukn       -  0102
```

```
Page 011              Module principal <main.as>
AREUH ASS. V2.4       **** SYMBOL TABLE ****

=TEDITp        000EF Rel  0137 -
 TxEn01        00074 Rel  0110 -  0085
 TxEn02        00085 Rel  0113 -  0089
 TxEn03        00094 Rel  0116 -  0093
 TxEn04        000A3 Rel  0119 -  0097
 TxEn05        000B6 Rel  0122 -  0101
 TxEn06        000C3 Rel  0125 -  0105
 TxTbSt        00074 Rel  0109 -  0031 0085 0089 0093 0097 0101 0105
 VER$en        0017A Rel  0186 -  0179 0186
 VER$st        0016A Rel  0185 -  0179 0186
=XEDITd        000D3 Rel  0131 -
=XEDITe        Extrn Ukn       -  0106
=XEDITp        000EF Rel  0138 -
 asf10         001E0 Rel  0252 -  0246
 asf20         001E7 Rel  0254 -  0250
 ck1li#        002FD Rel  0407 -  0402
 ck2li#        002E9 Rel  0401 -
 errp          00139 Rel  0165 -
 fichip        00110 Rel  0152 -  0143
 find          0029E Rel  0309 -  0079
 hASF          0019B Rel  0222 -  0205
 hASF1         00180 Rel  0205 -  0210 0219
 hKYDEF        00184 Rel  0207 -  0174
=id            000E1 Abs  0073 -  0076 0313
=kcCALC        Extrn Ukn       -  0217 0236
 numck         000E1 Rel  0134 -
=pKYDF         Extrn Ukn       -  0172
 respTr        002CE Rel  0362 -  0372 0404
 resptr        000E8 Rel  0135 -  0146 0147 0150
 rtn           001F2 Rel  0258 -  0231 0264 0280
 rtn1          00232 Rel  0277 -  0291
 rtnsm0        00299 Rel  0306 -  0256
=tCOMMA        Extrn Ukn       -  0411
=tLINE#        Extrn Ukn       -  0408
 t             00070 Abs  0074 -  0077 0078 0112 0115 0118 0121 0124 0127
 tFIND         0004B Abs  0311 -  0314 0315 0337
 virgup        00121 Rel  0156 -  0148


Source  : main.as

Object  : obj/main.ao

Listing : list/main.al

Date    : Sat Aug 12 17:40:48 1989
```

```
Page 012              Module principal <main.as>
AREUH ASS. V2.4       **** SYMBOL TABLE ****

Errors  : 000


Areuh Assembler/Linker V2.4. (c) P. David & J. Taillandier 1986 Paris, France
```

```
Page 001
AREUH ASS. V2.4

0001 00000          •
0002 00000          • Fichier généré automatiquement par amg
0003 00000          • et modifié automatiquement par chmsg
0004 00000          •
0005 00000          MBASE   EQU     16
0006 00000          NULL    EQU     (MBASE)+0     EDIT
0007 00000          =eXMEM  EQU     (MBASE)+1     Insufficient Memory
0008 00000          =teICMD EQU     (MBASE)+2     Invalid Cmd
0009 00000          =teIPAT EQU     (MBASE)+3     Invalid Pattern
0010 00000          =teICMP EQU     (MBASE)+4     No Room for Pattern
0011 00000          =tePLIN EQU     (MBASE)+5     Line ', Cmd:
0012 00000          =tePEOF EQU     (MBASE)+6     Eof, Cmd:
0013 00000          =teEOF  EQU     (MBASE)+7     [Eof]
0014 00000          =teDELE EQU     (MBASE)+8     Ok to Delete? Y/N:
0015 00000          =teCONF EQU     (MBASE)+9     Yes/No/Quit ?
0016 00000          =teKEYS EQU     (MBASE)+10    YNQ
0017 00000          =teVCMD EQU     (MBASE)+11    CDEHIJLMPQRSTX
0018 00000          =teHL00 EQU     (MBASE)+12    Copy: [b[e]] C [<file>]
0019 00000          =teHL01 EQU     (MBASE)+13    Delete: [b[e]] D [<file>[*]]
0020 00000          =teHL02 EQU     (MBASE)+14    Exit: E
0021 00000          =teHL03 EQU     (MBASE)+15    Help: H [<cmd>]
0022 00000          =teHL04 EQU     (MBASE)+16    Insert: [l] I
0023 00000          =teHL05 EQU     (MBASE)+17    Join: [b[e]] [?] J [n]
0024 00000          =teHL0E EQU     (MBASE)+18    List: [b[e]] L [n][N]
0025 00000          =teHL07 EQU     (MBASE)+19    Move: [b[e]] M [<file>]
0026 00000          =teHL08 EQU     (MBASE)+20    Print: [b[e]] P [n][N]
0027 00000          =teHL09 EQU     (MBASE)+21    Quit: Q
0028 00000          =teHL10 EQU     (MBASE)+22    Replace: [b[e]] [?] R/str1/str2[
0029 00000          =teHL11 EQU     (MBASE)+23    Search: [b[e]] [?] S/str[/]
0030 00000          =teHL12 EQU     (MBASE)+24    Text: [l] T
0031 00000          =teHL13 EQU     (MBASE)+25    Exchange File: X <file>
0032 00000
0033 00000          BB42    EQU     42            EDIT
0034 00000          BB43    EQU     43            Insufficient Memory
0035 00000          BB44    EQU     44            Invalid Cmd
0036 00000          BB45    EQU     45            Invalid Pattern
0037 00000
0038 00000          =MSGTBL1
0039 00000 11               CON(2) (MBASE)+1    Lowest message #
0040 00002 92               CON(2) (MBASE)+25   Highest message #
0041 00004
0042 00004          • EDIT
0043 00004 01               CON(2) 16
0044 00006 01               CON(2) NULL     Message # 16
0045 00008 4                CON(1) 4
0046 00009 54449445         NIBASC 'EDIT '
     00011 02
0047 00013 C                CON(1) 12
0048 00014
0049 00014          • Insufficient Memory
```

```
Page 002
AREUH ASS. V2.4

0050 00014 E2               CON(2) 46
0051 00016 11               CON(2) =eXMEM   Message # 17
0052 00018 B                CON(1) 11
0053 00019 F                CON(1) 15
0054 0001A 94E63757         NIBASC 'Insuffic'
     00022 66669636
0055 0002A 9656E647         NIBASC 'ient Mem'
     00032 0CD456D6
0056 0003A 2                CON(1) 2
0057 0003B F62797           NIBASC 'ory'
0058 00041 C                CON(1) 12
0059 00042
0060 00042          • Invalid Cmd
0061 00042 F0               CON(2) 15
0062 00044 21               CON(2) =teICMD  Message # 18
0063 00046 E                CON(1) 14
0064 00047 CE               CON(2) 236
0065 00049 2                CON(1) 2
0066 0004A 34D646           NIBASC 'Cmd'
0067 00050 C                CON(1) 12
0068 00051
0069 00051          • Invalid Pattern
0070 00051 B0               CON(2) 11
0071 00053 31               CON(2) =teIPAT  Message # 19
0072 00055 E                CON(1) 14
0073 00056 CE               CON(2) 236
0074 00058 D                CON(1) 13
0075 00059 A2               CON(2) BB42
0076 0005B C                CON(1) 12
0077 0005C
0078 0005C          • No Room for Pattern
0079 0005C 22               CON(2) 34
0080 0005E 41               CON(2) =teICMP  Message # 20
0081 00060 B                CON(1) 11
0082 00061 B                CON(1) 11
0083 00062 E4FE0225         NIBASC 'No Room '
     0006A F6F6D602
0084 00072 66F62702         NIBASC 'for '
0085 0007A D                CON(1) 13
0086 0007B A2               CON(2) BB42
0087 0007D C                CON(1) 12
0088 0007E
0089 0007E          • Line ', Cmd:
0090 0007E F1               CON(2) 31
0091 00080 51               CON(2) =tePLIN  Message # 21
0092 00082 4                CON(1) 4
0093 00083 C496E656         NIBASC 'Line '
     0008B 02
0094 0008D F2               NIBHEX F2
0095 0008F 5                CON(1) 5
```

```
Page 003
AREUH ASS. V2.4

0096 00090 C20234D6         NIBASC ', Cmd:'
     00098 46A3
0097 0009C C                CON(1) 12
0098 0009D
0099 0009D          • Eof, Cmd:
0100 0009D 81               CON(2, 24
0101 0009F 61               CON(2) =tePEOF  Message # 22
0102 000A1 8                CON(1) 8
0103 000A2 54F666C2         NIBASC 'Eof, Cmd'
     000AA 0234D646
0104 000B2 A3               NIBASC ':'
0105 000B4 C                CON(1) 12
0106 000B5
0107 000B5          • [Eof]
0108 000B5 01               CON(2) 16
0109 000B7 71               CON(2) =teEOF   Message # 23
0110 000B9 4                CON(1) 4
0111 000BA B554F666         NIBASC '[Eof]'
     000C2 D5
0112 000C4 C                CON(1) 12
0113 000C5
0114 000C5          • Ok to Delete? Y/N:
0115 000C5 C2               CON(2) 44
0116 000C7 81               CON(2) =teDELE  Message # 24
0117 000C9 B                CON(1) 11
0118 000CA F                CON(1) 15
0119 000CB F4B60247         NIBASC 'Ok to De'
     000D3 F6024456
0120 000DB C6564756         NIBASC 'lete? Y/'
     000E3 F30295F2
0121 000EB 1                CON(1) 1
0122 000EC E4A3             NIBASC 'N:'
0123 000F0 C                CON(1) 12
0124 000F1
0125 000F1          • Yes/No/Quit ?
0126 000F1 12               CON(2) 33
0127 000F3 91               CON(2) =teCONF  Message # 25
0128 000F5 B                CON(1) 11
0129 000F6 C                CON(1) 12
0130 000F7 955637F2         NIBASC 'Yes/No/Q'
     000FF E4F8F215
0131 00107 57964702         NIBASC 'uit ?'
     0010F F3
0132 00111 C                CON(1) 12
0133 00112
0134 00112          • YNQ
0135 00112 C0               CON(2) 12
0136 00114 A1               CON(2) =teKEYS  Message # 26
0137 00116 2                CON(1) 2
0138 00117 95E415           NIBASC 'YNQ'
```

```
Page 004
AREUH ASS. V2.4

0139 00110 C                CON(1) 12
0140 0011E
0141 0011E          • CDEHIJLMPQRSTX
0142 0011E 32               CON(2) 35
0143 00120 B1               CON(2) =teVCMD  Message # 27
0144 00122 B                CON(1) 11
0145 00123 D                CON(1) 13
0146 00124 34445484         NIBASC 'CDEHIJLM'
     0012C 94A4C4D4
0147 00134 05152535         NIBASC 'PQRSTX'
     0013C 4585
0148 00140 C                CON(1) 12
0149 00141
0150 00141          • Copy: [b[e]] C [<file>]
0151 00141 22               CON(2) 34
0152 00143 C1               CON(2) =teHL00  Message # 28
0153 00145 5                CON(1) 5
0154 00146 34F60797         NIBASC 'Copy: '
     0014E A302
0155 00150 D                CON(1) 13
0156 00153 B2               CON(2) BB43
0157 00155 2                CON(1) 2
0158 00156 3402B5           NIBASC 'C ['
0159 0015C D                CON(1) 13
0160 0015D C2               CON(2) BB44
0161 0015F 0                CON(1) 0
0162 00160 D5               NIBASC ']'
0163 00162 C                CON(1) 12
0164 00163
0165 00163          • Delete: [b[e]] D [<file>[*]]
0166 00163 C2               CON(2) 44
0167 00165 D1               CON(2) =teHL01  Message # 29
0168 00167 7                CON(1) 7
0169 00168 4456C656         NIBASC 'Delete: '
     00170 4756A302
0170 00178 D                CON(1) 13
0171 00179 B2               CON(2) BB43
0172 0017B 2                CON(1) 2
0173 0017C 3402B5           NIBASC 'D ['
0174 00182 D                CON(1) 13
0175 00183 C2               CON(2) BB44
0176 00185 3                CON(1) 3
0177 00186 B5B2D5D5         NIBASC '[*]]'
0178 0018E C                CON(1) 12
0179 0018F
0180 0018F          • Exit: E
0181 0018F 41               CON(2) 20
0182 00191 E1               CON(2) =teHL0E  Message # 30
0183 00193 6                CON(1) 6
0184 00194 54879647         NIBASC 'Exit: E'
```

```
Page 005
AREUH ASS. V2.4

     0019C A30254
     001A2 C                     CON(1) 12
     001A3
     001A3                  • Help: H [<cmd>]
     001A3 52                    CON(2) 37
     001A5 F1                    CON(2) =teHL03  Message # 31
     001A7 B                     CON(1) 11
     001A8 E                     CON(1) 14
     001A9 8456C607              NIBASC 'Help: H '
     001B1 A3028402
     001B9 B5C336D6              NIBASC '[<cmd>]'
     001C1 46E3D5
     001C7 C                     CON(1) 12
     001C8
     001C8                  • Insert: [1] I
     001C8 12                    CON(2) 33
     001CA 02                    CON(2) =teHL04  Message # 32
     001CC B                     CON(1) 11
     001CD C                     CON(1) 12
     001CE 94E63756              NIBASC 'Insert: '
     001D6 2747A302
     001DE B5C6D502              NIBASC '[1] I'
     001E6 94
     001E8 C                     CON(1) 12
     001E9
     001E9                  • Join: [b[e]] [?] J [n]
     001E9 82                    CON(2) 40
     001EB 12                    CON(2) =teHL05  Message # 33
     001ED 5                     CON(1) 5
     001EE A4F696E6              NIBASC 'Join: '
     001F6 A302
     001FA D                     CON(1) 13
     001FB B2                    CON(2) BB43
     001FD 8                     CON(1) 8
     001FE B5F3D502              NIBASC '[?] J [n'
     00206 A402B5E6
     0020E D5                    NIBASC ']'
     00210 C                     CON(1) 12
     00211
     00211                  • List: [b[e]] L [n][N]
     00211 B1                    CON(2) 27
     00213 22                    CON(2) =teHL06  Message # 34
     00215 5                     CON(1) 5
     00216 C4963747              NIBASC 'List: '
     0021E A302
     00222 D                     CON(1) 13
     00223 B2                    CON(2) BB43
     00225 0                     CON(1) 0
     00226 C4                    NIBASC 'L'
     00228 D                     CON(1) 13
```

```
Page 006
AREUH ASS. V2.4

     00229 D2                    CON(2) BB45
     0022B C                     CON(1) 12
     0022C
     0022C                  • Move: [b[e]] M [<file>]
     0022C 22                    CON(2) 34
     0022E 32                    CON(2) =teHL07  Message # 35
     00230 5                     CON(1) 5
     00231 D4F56756              NIBASC 'Move: '
     00239 A302
     0023D D                     CON(1) 13
     [illegible]                 CON(2) BB4[illegible]
     [illegible]                 [illegible]
     [illegible]                 NIBASC 'M ['
     [illegible]                 CON(1) 13
     [illegible]                 CON(2) BB44
     0024A 0                     CON(1) 0
     0024B D5                    NIBASC ']'
     0024D C                     CON(1) 12
     0024E
     0024E                  • Print: [b[e]] P [n][N]
     0024E D1                    CON(2) 29
     00250 42                    CON(2) =teHL08  Message # 36
     00252 6                     CON(1) 6
     00253 052796E6              NIBASC 'Print: '
     0025B 47A302
     00261 D                     CON(1) 13
     00262 B2                    CON(2) BB43
     00264 0                     CON(1) 0
     00265 05                    NIBASC 'P'
     00267 D                     CON(1) 13
     00268 D2                    CON(2) BB45
     0026A C                     CON(1) 12
     0026B
     0026B                  • Quit: Q
     0026B 41                    CON(2) 20
     0026D 52                    CON(2) =teHL09  Message # 37
     0026F 6                     CON(1) 6
     00270 15573647              NIBASC 'Quit: Q'
     00278 A30215
     0027E C                     CON(1) 12
     0027F
     0027F                  • Replace: [b[e]] [?] R/str1/str2[/]
     0027F 24                    CON(2) 66
     00281 F2                    CON(2) =teHL10  Message # 38
     00283 8                     CON(1) 8
     00284 25560706              NIBASC 'Replace:'
     0028C 163656A3
     00294 02                    NIBASC ' '
     00296 D                     CON(1) 13
     00297 B2                    CON(2) BB43
```

```
Page 007
AREUH ASS. V2.4

     00299 B                     CON(1) 11
     0029A F                     CON(1) 15
     0029B B5F3D502              NIBASC '[?] R/st'
     002A3 25F23747
     002AB 2713F237              NIBASC 'r1/str2['
     002B3 472723B5
     002BB 1                     CON(1) 1
     002BC F2D5                  NIBASC '/]'
     002C0 C                     CON(1) 12
     002C1
     002C1                  • Search: [b[e]] [?] S/str[/]
     002C1 33                    CON(2) 51
     002C3 72                    CON(2) =teHL11  Message # 39
     002C5 7                     CON(1) 7
     002C6 35561627              NIBASC 'Search: '
     002CE 3686A302
     002D6 D                     CON(1) 13
     002D7 B2                    CON(2) BB43
     002D9 B                     CON(1) 11
     002DA B                     CON(1) 11
     002DB B5F3D502              NIBASC '[?] S/st'
     002E3 35F23747
     002EB 27B5F2D5              NIBASC 'r[/]'
     002F3 C                     CON(1) 12
     002F4
     002F4                  • Text: [1] T
     002F4 C1                    CON(2) 28
     002F6 82                    CON(2) =teHL12  Message # 40
     002F8 A                     CON(1) 10
     002F9 45568747              NIBASC 'Text: [1'
     00301 A302B5C6
     00309 D50245                NIBASC '] T'
     0030F C                     CON(1) 12
     00310
     00310                  • Exchange File: X <file>
     00310 D2                    CON(2) 45
     00312 92                    CON(2) =teHL13  Message # 41
     00314 B                     CON(1) 11
     00315 F                     CON(1) 15
     00316 54873686              NIBASC 'Exchange'
     0031E 16E67656
     00326 026496C6              NIBASC ' File: X'
     0032E 56A30285
     00336 0                     CON(1) 0
     00337 02                    NIBASC ' '
     00339 D                     CON(1) 13
     0033A C2                    CON(2) BB44
     0033C C                     CON(1) 12
     0033D
     0033D                  • Pattern
```

```
Page 008
AREUH ASS. V2.4

     0033D 41                    CON(2) 20
     0033F A2                    CON(2) BB42      Message # 42
     00341 6                     CON(1) 6
     00342 05164747              NIBASC 'Pattern'
     0034A 5627E6
     00350 C                     CON(1) 12
     00351
     00351                  • [b[e]]
     00351 41                    CON(2) 20
     00353 B2                    CON(2) BB43      Message # 43
     00355 6                     CON(1) 6
     00356 [illegible]           NIBASC '[b[e]] '
     0035E D5D502
     00364 C                     CON(1) 12
     00365
     00365                  • <file
     00365 21                    CON(2) 18
     00367 C2                    CON(2) BB44      Message # 44
     00369 5                     CON(1) 5
     0036A C366960C6             NIBASC '<file>'
     00372 56E3
     00376 C                     CON(1) 12
     00377
     00377                  •  [n][N]
     00377 41                    CON(2) 20
     00379 D2                    CON(2) BB45      Message # 45
     0037B 6                     CON(1) 6
     0037C 02B5E6D5              NIBASC ' [n][N]'
     00384 B5E4D5
     0038A C                     CON(1) 12
     0038B
     0038B FF                    NIBHEX FF        Table terminator
```

```
Page 009
AREUH ASS. V2.4    **** SYMBOL TABLE ****

 BB42          0002A Abs  0033 -  0075 0035 0317
 BB43          0002B Abs  0034 -  0158 0171 0211 0223 0236 0251 0272 0287 0304
 BB44          0002C Abs  0035 -  0160 0175 0240 0312 0331
 BB45          0002D Abs  0036 -  0027 0255 0308
 FiLeNd        0038D Rel  0344 -
=MSGTBL1       00000 Rel  0038 -
 MBASE         00010 Abs  0005 -  0006 0007 0008 0009 0010 0011 0012 0013 0014
                                +  0015 0016 0017 0018 0019 0020 0021 0022 0023
                                +  0024 0025 0026 0027 0028 0029 0030 0031 0039
                                +  0040
 NULL          00010 Abs  0006 -  0044
=eXMEM         00011 Abs  0007 -  0051
=teCONF        00019 Abs  0015 -  0127
=teDELE        00018 Abs  0014 -  0116
=teEOF         00017 Abs  0013 -  0109
=teHL00        0001C Abs  0018 -  0152
=teHL01        0001D Abs  0019 -  0167
=teHL02        0001E Abs  0020 -  0182
=teHL03        0001F Abs  0021 -  0189
=teHL04        00020 Abs  0022 -  0198
=teHL05        00021 Abs  0023 -  0207
=teHL06        00022 Abs  0024 -  0219
=teHL07        00023 Abs  0025 -  0232
=teHL08        00024 Abs  0026 -  0247
=teHL09        00025 Abs  0027 -  0260
=teHL10        00026 Abs  0028 -  0267
=teHL11        00027 Abs  0029 -  0283
=teHL12        00028 Abs  0030 -  0296
=teHL13        00029 Abs  0031 -  0304
=teICMD        00012 Abs  0008 -  0062
=teICMF        00014 Abs  0010 -  0080
=teIPAT        00013 Abs  0009 -  0071
=teKEYS        0001A Abs  0016 -  0136
=tePEOF        00016 Abs  0012 -  0101
=tePLIN        00015 Abs  0011 -  0091
=teVCMD        0001B Abs  0017 -  0143


Source  : msg1.as

Object  : obj/msg1.ao

Listing : list/msg1.al

Date    : Sat Aug 12 17:40:51 1989
```

```
Page 001
AREUH ASS. V2.4

0001 00000          •
0002 00000          • Fichier généré automatiquement par amg
0003 00000          • et modifié automatiquement par chmig
0004 00000          •
0005 00000          MBASE   EQU     1
0006 00000          NULL    EQU     (MBASE)+0       FIND
0007 00000          =eNFND  EQU     (MBASE)+1       Not Found
0008 00000
0009 00000          =MSGTBL2
0010 00000 20               CON(2) (MBASE)+1    Lowest message #
0011 00002 20               CON(2) (MBASE)+1    Highest message #
0012 00004
0013 00004          • FIND
0014 00004 01               CON(2) 16
0015 00006 10               CON(2) NULL     Message # 1
0016 00008 4                CON(1) 4
0017 00009 6494E444         NIBASC 'FIND '
     0001 02
0018 00013 C                CON(1) 12
0019 00014
0020 00014          • Not Found
0021 00014 81               CON(2) 24
0022 00016 20               CON(2) =eNFND   Message # 2
0023 00018 8                CON(1) 8
0024 00019 E4F64702         NIBASC 'Not Foun'
     0021 64F657E8
0025 00029 46               NIBASC 'd'
0026 0002B C                CON(1) 12
0027 0002C
0028 0002C FF               NIBHEx FF       Table terminator
```

```
Page 010
AREUH ASS. V2.4    **** SYMBOL TABLE ****

Errors  : 000


Areuh Assembler/Linker V2.4, (c) P. David & J. Taillandier 1986 Paris, France
```

```
Page 002
AREUH ASS. V2.4    **** SYMBOL TABLE ****

 FiLeNd        0002E Rel  0029 -
=MSGTBL2       00000 Rel  0009 -
 MBASE         00001 Abs  0005 -  0006 0007 0010 0011
 NULL          00001 Abs  0006 -  0015
=eNFND         00002 Abs  0007 -  0022


Source  : msg2.as

Object  : obj/msg2.ao

Listing : list/msg2.al

Date    : Sat Aug 12 17:40:53 1989

Errors  : 000


Areuh Assembler/Linker V2.4, (c) P. David & J. Taillandier 1986 Paris, France
```

```
Page 001            Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4

00000                     TITLE   Editeur, décomposition des cmd <xdec.as>
00000
00000             *
00000             * Variables temporaires
00000             *
00000              tmp    EQU     (=SCRTCH)-1
00000
00000              val    EQU     (tmp)-5         5 q dans expr
00000              signe  EQU     (tmp)-6         1 q dans expr
00000
00000              sauve  EQU     =STMTR0         20 q pendant recherche
00000              debut  EQU     (=STMTR0)+20    5 q pendant "expr"
00000             ****************************************************************
00000             * skip
00000             *
00000             * But: sauter les blancs
00000             * Entree:
00000             *    - D1 = ^ chaine
00000             *    - D(A) = nb de caractères
00000             * Sortie: -
00000             * Abime: A(B), C(B)
00000             * Appelle: lookah, getchr
00000             * Niveaux: 1
00000             * Historique:
00000             *    88/05/14: PD/JT conception & codage
00000             *    88/05/15: PD/JT isolement dans un module séparé
00000             ****************************************************************
00000
00000 3102        =skip   LCASC   ' '
00004 7E10         skip10 GOSUB   =lookah
00008 400                 RTNC              Plus de caractère
0000B 956                 ?A#C    B
0000E 00                  RTNYES
00010 7300                GOSUB   =getchr
00014 5FE                 GONC    skip10  B.E.T.
00017
00017             ****************************************************************
00017             * lookah, getchr
00017             *
00017             * But: lire un caractère, et passer au suivant (getchr)
00017             * Entree:
00017             *    - D1 = ^ chaine
00017             *    - D(A) = nb de caractères
00017             * Sortie:
00017             *    - Cy = 1 : fin de la chaine
00017             *    - Cy = 0, A(B) = caractère lu
00017             * Abime: A(B)
00017             * Appelle: -
00017             * Niveaux: 0
```

```
Page 002            Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4

00017             * Historique:
00017             *    88/05/14: PD/JT conception & codage
00017             *    88/05/15: PD/JT isolement dans un module séparé
00017             ****************************************************************
00017
00017 8AB         =getchr ?D=0    A
0001A 00                  RTNYES
0001C CF                  D=D-1   A
0001E 14B                 A=DAT1  B
00021 1C1                 D1=D1-  2
00024 01                  RTN               Cy = 0
00026
00026 8AB         =lookah ?D=0    A
00029 00                  RTNYES
0002B 14B                 A=DAT1  B
0002E 01                  RTN               Cy = 0
00030
00030             ****************************************************************
00030             * xeddec
00030             *
00030             * But: décomposer une ligne de commande
00030             * Entree:
00030             *    - LCMD, PCMD positonnés
00030             * Sortie:
00030             *    - Cy = 1 : rien trouve
00030             *    - Cy = 0 : trouve une commande (éventuellement nulle)
00030             *      A(B) = cmd (en majuscule, 0xff si pas de commande)
00030             *      D(A) = longueur de PARAM
00030             *      D1 = adresse de PARAM
00030             *      R0 = nblignes (0, 1 ou 2)
00030             *       R1 = ldebut
00030             *       R2 = lfin
00030             *      R3 = query (1 si '?' reconnu avant la commande)
00030             * Attention: PCMD et LCMD ne sont pas bien positionnés en
00030             *   sortie. Pour cela, il faut que chaque commande appelle
00030             *   setCMD pour stocker D1 et D(A) dans CMD.
00030             * Abime: A, C, D, D1
00030             * Appelle: skip, getchr, lookah, linenb, nbli++, expr
00030             * Niveaux: 3
00030             * Historique:
00030             *    88/05/15: PD/JT conception & codage
00030             *    88/10/16: PD/JT retrait de la "." obligatoire dans HP-UX
00030             ****************************************************************
00030
00030 1F00000     =xeddec D1=(5)  =MODE71
00037 1574                C=DAT1  S       C(S) := mode71 pour tout à l'heure
0003B             *
0003B             * Positionne D(A) et D1 pour utilisation de getchr et lookah
0003B             *
0003B 1D00                D1=(2)  =LCMD
```

```
Page 003            Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4

0003F 147                 C=DAT1  A
00042 D7                  D=C     A       D(A) := longueur
00044 1D00                D1=(2)  =PCMD
00048 147                 C=DAT1  A
0004B 135                 D1=C            D1 := ^ caractère à lire
0004E             *
0004E             * Y-a-t'il quelque chose dans la ligne de commande ?
0004E             *
0004E 7EAF                GOSUB   =skip
00052 70DF                GOSUB   =lookah
00056 400                 RTNC            Cy = 1 : rien trouvé
00059             *
00059             * Il y a quelque chose.
00059             *
00059 D2                  C=0     A
0005B 108                 R0=C            nblignes := 0
0005E             *
0005E             * Mode HP-71 ou HP-UX ?
0005E             *
0005E 94A                 ?C=0    S       Si si, ça tient dans C(S) !
00061 B4                  GOYES   dec100  Mode HP-UX
00063
00063             ****************************************************************
00063             * Mode HP-71
00063             ****************************************************************
00063             *
00063             * [ldebut [[.] lfin]] [?] [cmd] [param]
00063             *
00063             *
00063 7DF0                GOSUB   linenb
00067 404                 GOC     dec050  ldebut non reconnu
0006A             *
0006A             * [ldebut [[.] lfin]] [?] [cmd] [param]
0006A             *
0006A             *
0006A 101                 R1=A            R1 := ldebut
0006D 7623                GOSUB   nbli++  nblignes++
00071
00071 7B8F                GOSUB   =skip
00075 7DAF                GOSUB   =lookah
00079 4E2                 GOC     dec050  EOL
0007C 31C2                LC(2)   '.'
00080 966                 ?A#C    B
00083 71                  GOYES   dec020  virgule non reconnue
00085
00085 7E8F                GOSUB   =getchr passer la virgule
00089             *
00089             * [ldebut [[.] lfin]] [?] [cmd] [param]
00089             *
00089             *
```

```
Page 004            Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4

00089 77D0                GOSUB   linenb
0008D 531                 GONC    dec030  lfin reconnu
00090 330000              LC(4)   =eMSPAR "5."
00096 6BB2                GOTO    erreur
0009A
0009A 76C0         dec020 GOSUB   linenb
0009E 490                 GOC     dec050  lfin non reconnu
000A1             *
000A1             * [ldebut [[.] lfin]] [?] [cmd] [param]
000A1             *
000A1             *
000A1 102          dec030 R2=A            R2 := lfin
000A4 7FE2                GOSUB   nbli++  nblignes++
000A8 6560         dec050 GOTO    dec200
000AC
000AC             ****************************************************************
000AC             * Mode HP-UX
000AC             ****************************************************************
000AC
000AC              dec100
000AC             *
000AC             * [ldebut [[.] lfin]] [?] [cmd] [param]
000AC             *
000AC             *
000AC
000AC             *
000AC             * ajout du 88/10/07 :
000AC             * ligne de départ de la recherche = courante
000AC 1B00000             D0=(5)  =COUR
000B3 142                 A=DAT0  A
000B6 1B00000             D0=(5)  debut   debut := COUR
000BD 140                 DAT0=A  A
000C0             * fin de l'ajout
000C0             *
000C0 7631                GOSUB   expr
000C4 494                 GOC     dec200  ldebut non reconnu
000C7 101                 R1=A            ldebut
000CA 79C2                GOSUB   nbli++  nblignes++
000CE             *
000CE             * ajout du 88/10/07 :
000CE             * ligne de départ de la recherche = ldebut
000CE 111                 A=R1
000D1 1B00000             D0=(5)  debut   debut := R1
000D8 140                 DAT0=A  A
000DB             * fin de l'ajout
000DB
000DB             *
000DB             * [ldebut [[.] lfin]] [?] [cmd] [param]
000DB             *
000DB             *
```

```
Page 005          Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4

0201 000DB 774F             GOSUB   =lookah
0202 000DF 4E2              GOC     dec200
0203 000E2 31C2             LCASC   ':'
0204 000E6 966              ?A#C    B
0205 000E9 71               GOYES   dec120  virgule non reconnue
0206 000EB 782F             GOSUB   =getchr
0207 000EF                  *
0208 000EF                  * [ldebut [[.] lfin]] [?] [cmd] [param]
0209 000EF                  *
0210 000EF                  *
0211 000EF 7701             GOSUB   expr
0212 000F3 531              GONC    dec130
0213 000F6 330000           LC(4)   =eMSPAR Missing parameter
0214 000FC 6552             GOTO    erreur
0215 00100                  *
0216 00100 76F0     dec120  GOSUB   expr
0217 00104 490              GOC     dec200  nombre non reconnu
0218 00107                  *
0219 00107                  * [ldebut [[.] lfin]] [?] [cmd] [param]
0220 00107                  *
0221 00107                  *
0222 00107 102      dec130  R2=A            lfin
0223 0010A 7982             GOSUB   nbli++  nblignes++
0224 0010E                  *
0225 0010E                  * Attention ! le code continue !
0226 0010E                  *
0227 0010E                  *
0228 0010E                  ****************************************************
0229 0010E                  * Partie commune HP-71/HP-UX
0230 0010E                  ****************************************************
0231 0010E                  *
0232 0010E 7EEE     dec200  GOSUB   =skip
0233 00112                  *
0234 00112                  * [?] [cmd] [param]
0235 00112                  *
0236 00112                  *
0237 00112 AC2              C=0     S       query := 0
0238 00115                  *
0239 00115 7D0F             GOSUB   =lookah
0240 00119 421              GOC     dec300  EOL
0241 0011C 31F3             LCASC   '?'
0242 00120 966              ?A#C    B
0243 00123 90               GOYES   dec300
0244 00125 7EEE             GOSUB   =getchr pligne++
0245 00129 846              C=C+1   S       query := 1
0246 0012C 1WB      dec300  R=C             query := 0 ou 1
0247 0012F                  *
0248 0012F                  * [?] [cmd] [param]
0249 0012F                  *
0250 0012F                  *
```

```
Page 006          Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4

0251 0012F 7DCE             GOSUB   =skip
0252 00133 7FEE             GOSUB   =lookah
0253 00137 4B0              GOC     dec320  EOL
0254 0013A                  *
0255 0013A                  * Fin de commande (':')
0256 0013A                  *
0257 0013A 31B3             LCASC   ':'
0258 0013E 966              ?A#C    B
0259 00141 B0               GOYES   dec350
0260 00143 AE0      dec320  A=0     B
0261 00146 A0C              A=A-1   B       cmd := 0xff
0262 00149 481              GOC     dec300  fin de decomp (B.E.T.)
0263 0014C                  *
0264 0014C                  * cas général
0265 0014C                  *
0266 0014C 77CE     dec350  GOSUB   =getchr A(B) := cmd (caractère existe)
0267 00150 AE8              B=A     B       protéger A(B)
0268 00153 79AE             GOSUB   =skip
0269 00157 DA               A=C     A
0270 00159 DC               ABEX    A       A(A) := cmd ; B(A) := len(param)
0271 0015B 8F0000           GOSBVL  =UCONVUC A(B) := cmd en majuscule
0272 00162
0273 00162                  ****************************************************
0274 00162                  * fin de decomp
0275 00162                  ****************************************************
0276 00162
0277 00162 03       dec900  RTNCC           Un seul point de retour: c'est beau
0278 00164
0279 00164                  ****************************************************
0280 00164                  * linenb
0281 00164                  *
0282 00164                  * But: lire un numéro de ligne
0283 00164                  * Entree:
0284 00164                  *   - D1 =   chaine
0285 00164                  *   - D(A) = nb de caractères
0286 00164                  * Sortie:
0287 00164                  *   - Cy = 1 : non reconnu
0288 00164                  *   - Cy = 0, A(A) = nombre reconnu
0289 00164                  * Abime: A(A), C(A)
0290 00164                  * Appelle: skip, getchr, lookah, DRANGE, nombre (tombe deds)
0291 00164                  * Niveaux: 1
0292 00164                  * Historique:
0293 00164                  *   88/05/14: PD/JT conception & codage
0294 00164                  *   88/05/15: PD/JT isolement dans un module séparé
0295 00164                  ****************************************************
0296 00164
0297 00164 789E     linenb  GOSUB   =skip
0298 00168 7ABE             GOSUB   =lookah
0299 0016C 31E2             LCASC   ' '
0300 00170 966              ?A#C    B
```

```
Page 007          Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4

0301 00173 21               GOYES   lnb010
0302 00175 7E9E             GOSUB   =getchr
0303 00179 1B0000           D0=(5)  =COUR   return courante :
0304 00180 142              A=DAT0  A
0305 00183 01               RTN             Cy = 0
0306 00185 3132     lnb010  LCASC   '#'
0307 00189 966              ?A#C    B
0308 0018C 21               GOYES   lnb020
0309 0018E 798E             GOSUB   =getchr
0310 00192 1B0000           D0=(5)  =DEPN   return derniere :
0311 00193 142              A=DAT0  A
0312 0019C 01               RTN             Cy = 0
0313 0019E          lnb020
0314 0019E
0315 0019E                  *
0316 0019E                  * Attention ! le code continue !
0317 0019E                  *
0318 0019E
0319 0019E                  ****************************************************
0320 0019E                  * nombre
0321 0019E                  *
0322 0019E                  * But: lire un nombre
0323 0019E                  * Entree:
0324 0019E                  *   - D1 = ' chaine
0325 0019E                  *   - D(A) = nb de caractères
0326 0019E                  * Sortie:
0327 0019E                  *   - Cy = 1 : nombre non reconnu
0328 0019E                  *   - Cy = 0, A(A) = nombre reconnu
0329 0019E                  * Abime: A(A), B(A), C(A) (D(A) et D1 réactualisés)
0330 0019E                  * Appelle: skip, getchr, lookah, DRANGE
0331 0019E                  * Niveaux: 1
0332 0019E                  * Historique:
0333 0019E                  *   88/05/14: PD/JT conception & codage
0334 0019E                  *   88/10/29: PD/JT ajout du test de EOL
0335 0019E                  *   88/10/29: PD/JT documentation
0336 0019E                  ****************************************************
0337 0019E
0338 0019E                  *
0339 0019E                  * Attention ! Le code vient d'en haut...
0340 0019E                  *
0341 0019E 748E     =nombre GOSUB   =lookah
0342 001A2 400              RTNC            Non, il n'y a rien...
0343 001A5 8F0000           GOSBVL  =DRANGE
0344 001A9 400              RTNC            Non, il n'y a rien...
0345 001AF D1               B=0     A
0346 001B1 717E     nbr010  GOSUB   =lookah
0347 001B5 404              GOC     nbr090  EOL
0348 001B8 8F0000           GOSBVL  =DRANGE
0349 001BF 463              GOC     nbr090  not in [0..9]
0350 001C2 715E             GOSUB   =getchr
```

```
Page 008          Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4

0351 001C6
0352 001C6 D9               C=B     A       vérification que B < 65536
0353 001C8 24               P=      4
0354 001CA A82              C=0     P
0355 001CD 20               P=      0
0356 001CF 8A5              ?B#C    A
0357 001D2 A1               GOYES   ivarg
0358 001D4
0359 001D4 DC               C=0     A       C:= on peut encore multiplier par 10
0360 001D6 2103             LCASC   '0'
0361 001DA B2E              C=A-C   B       [ A ] := ATH(A(B))
0362 001DD CE               B=B+B   A
0363 001DF D4               A=B     A       A(A) := 2B
0364 001E1 CE               B=B+B   A
0365 001E3 C5               B=B+B   A       B(A) := 8B
0366 001E5 C8               B=A+B   A       B(A) := 10B
0367 001E7 C1               B=B+C   A       B(A) := 10B + C(A)
0368 001E9 57C              GONC    nbr010  B.E.T.
0369 001EC 330000   ivarg   LC(4)   =eIVARG
0370 001F2 6F51             GOTO    erreur
0371 001F6 D4       nbr090  A=B     A
0372 001F8 03               RTNCC           Cy = 0
0373 001FA
0374 001FA                  ****************************************************
0375 001FA                  * expr
0376 001FA                  *
0377 001FA                  * But: analyser une expression à la syntaxe HP-UX
0378 001FA                  * Entree:
0379 001FA                  *   - D1 = ' chaine
0380 001FA                  *   - D(A) = nb de caractères
0381 001FA                  *   - debut = numéro de la ligne de debut de recherche
0382 001FA                  * Sortie:
0383 001FA                  *   - Cy = 1 : non reconnu
0384 001FA                  *   - Cy = 0, A(A) = nombre reconnu
0385 001FA                  * Abime: val, signe, A(A), C(S), C(A), D0, D1
0386 001FA                  * Appelle: skip, expelm, lookah, getchr
0387 001FA                  * Niveaux:
0388 001FA                  * Historique:
0389 001FA                  *   88/05/24: PD/JT conception & codage
0390 001FA                  *   88/10/07: PD/JT ajout de "debut" dans la recherche
0391 001FA                  ****************************************************
0392 001FA
0393 001FA 7680     expr    GOSUB   expelm
0394 001FE 400              RTNC            nombre non reconnu
0395 00201 1B0000   expr00  D0=(5)  val
0396 00208 140              DAT0=A  A       val := A(A)
0397 0020B 71FD             GOSUB   =skip
0398 0020F 731E             GOSUB   =lookah
0399 00213 446              GOC     expr99  sortir enfin
0400 00216 AC2              C=0     S
```

```
Page 009          Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4

0401 00219 31B2             LCASC   '+'
0402 0021D 962              ?A=C    B
0403 00220 E0               GOYES   expr10  C(S) := #0 pour '+'
0404 00222 A4E              C=C-1   S       C(S) := #F pour '-'
0405 00225 31D2             LCASC   '-'
0406 00229 966              ?A#C    B
0407 0022C C4               GOYES   expr99  sortir si ni '+', ni '-'
0408 0022E 1B0000   expr10  D0=(5)  signe
0409 00235 1544             DAT0=C  S       signe := 0 si '+', #F si '-'
0410 00239 7ADD             GOSUB   =getchr passer l'opérateur
0411 0023D 7340             GOSUB   expeln
0412 00241 5C0              GONC    expr20  ok...
0413 00244 330000           LC(4)   =eSYNTX "5-"
0414 0024A E701             GOTO    erreur
0415 0024E 1B0000   expr20  D0=(5)  signe
0416 00255 1564             C=DAT0  S
0417 00259 1900             D0=(2)  val
0418 0025D 146              C=DAT0  A
0419 00260 84A              ?C=0    S
0420 00263 C4               GOYES   expr30  '+'
0421 00265 0E               ACEX    A
0422 00267 EA               A=A-C   A       '-'
0423 0026A 579              GONC    expr00  si tout va bien
0424 0026C 470              GOC     Ivarg   B.E.T.
0425 0026F CA       expr30  A=A+C   A
0426 00271 5F5              GONC    expr00  si tout va bien
0427 00274 E7FF     Ivarg   GOTO    ivarg   B.E.T.
0428 00278
0429 00278 1B0000   expr99  D0=(5)  val
0430 0027F 145              A=DAT0  A
0431 00282 03               RTNCC           Ok, ca a marché...
0432 00284
0433 00284          ********************************************************
0434 00284          * expelm
0435 00284          *
0436 00284          * But: analyser un élément d'expression à la syntaxe HP-UX
0437 00284          * Entree:
0438 00284          *   - D1 = ^ chaine
0439 00284          *   - D(A) = nb de caractères
0440 00284          *   - debut = numéro de la première ligne de la recherche
0441 00284          * Sortie:
0442 00284          *   - Cy = 1 : non reconnu
0443 00284          *   - Cy = 0, A(A) = nombre reconnu
0444 00284          * Abime:
0445 00284          * Appelle: skip, lookah, GETSRC, getchr
0446 00284          * Niveaux:
0447 00284          * Historique:
0448 00284          *   88/05/24: FD/JT conception & codage
0449 00284          *   88/10/07: FD/JT ajout de "debut" dans la recherche
0450 00284          *   88/10/15: FD/JT modification pour nouveau GETSRC
```

```
Page 011          Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4

0501 0030C
0502 0030C 1B0000           D0=(5)  debut
0503 00313 142              A=DAT0  A
0504 00316 E4               A=A+1   A       A(A) := debut + 1
0505 00318 1B0000           D0=(5)  =DERN
0506 0031F 146              C=DAT0  A       C(A) := dernière
0507 00322 886              ?A>C    A
0508 00326 B0               GOYES   ex1035
0509 00327 8E0000           GOSUBL  =SRCLIN
0510 0032D 482              GOC     ex1040  match found
0511 00330
0512 00330 D0       ex1035  A=0     A
0513 00332 E4               A=A+1   A       A(A) := 1
0514 00334 1B0000           D0=(5)  debut
0515 0033B 146              C=DAT0  A       C(A) := début
0516 0033E 8B6              ?A>C    A
0517 00341 B0               GOYES   notfnd
0518 00343 8E0000           GOSUBL  =SRCLIN
0519 00349 4C0              GOC     ex1040  match found
0520 0034C
0521 0034C 330000   notfnd  LC(4)   (=id)"(=eNFND)  "Not Found"
0522 00352 6000     erreur  GOTO    =xederr aie aie aie...
0523 00356
0524 00356          *
0525 00356          * Match found
0526 00356          *   A(A) = ligne trouvée
0527 00356          *
0528 00356 101      ex1040  R1=A             Sauvegarde de A(A)
0529 00359 8E0000           GOSUBL  =ENDPOS Achever le buffer (arghhh)
0530 0035F 111              A=R1            Restauration de A(A)
0531 00362
0532 00362 AF2              C=0     W       on en profite pour nettoyer
0533 00365 1B0000           D0=(5)  sauve   R0(A)
0534 0036C 146              C=DAT0  A
0535 0036F 103              R0=C
0536 00372 164              D0=D0+  5       R1(A)
0537 00375 146              C=DAT0  A
0538 00378 109              R1=C
0539 0037B 164              D0=D0+  5       D1
0540 0037E 146              C=DAT0  A
0541 00381 105              D1=C
0542 00384 164              D0=D0+  5       D(A)
0543 00387 146              C=DAT0  A
0544 0038A D7               D=C     A
0545 0038C
0546 0038C 24               P=      4       restaurer 5 niveaux
0547 0038E 8F0000           GOSBVL  =RSTK=R
0548 00395
0549 00395 03               RTNCC           on a trouvé !
0550 00397
```

```
Page 012          Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4

0551 00397          ********************************************************
0552 00397          * nbli++
0553 00397          *
0554 00397          * But: incrémenter R0
0555 00397          * Entree:
0556 00397          *   - R0
0557 00397          * Sortie:
0558 00397          *   - R0 := R0+1
0559 00397          * Abime: C(W)
0560 00397          * Niveaux: 0
0561 00397          * Historique:
0562 00397          *   88/05/15: FD/JT conception & codage
0563 00397          ********************************************************
0564 00397
0565 00397 128      nbli++  CR0EX
0566 0039A E6               C=C+1   A
0567 0039C 128              CR0EX
0568 0039F 01               RTN
0569 003A1
```

```
Page 010          Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4

0451 00284          ********************************************************
0452 00284
0453 00284 787D      expelm GOSUB   =skip
0454 00288 7A3D             GOSUB   =lookah
0455 0028C 31E2             LCASC   '('
0456 00290 966              ?A#C    B
0457 00293 21               GOYES   ex1010
0458 00295 7E7D             GOSUB   =getchr
0459 00299 1B0000           D0=(5)  =COUR   return courante :
0460 0029F 142              A=DAT0  A
0461 002A2 01               RTN             Cy = 0
0462 002A5 3142     ex1010  LCASC   ')'
0463 002A9 96F              ?A#C    B
0464 002AB 21               GOYES   ex1020
0465 002AE 7E6D             GOSUB   =getchr
0466 002B2 1B0000           D0=(5)  =DERN   return dernière :
0467 002B9 142              A=DAT0  A
0468 002BC 01               RTN             Cy = 0
0469 002BF 31F2     ex1020  LCASC   '/'
0470 002C3 962              ?A=C    B
0471 002C5 F0               GOYES   ex103P
0472 002C7 66DE             GOTO    =nombre "default"
0473 002CB
0474 002CB          ex1030
0475 002CB          *
0476 002CB          * Recherche : /.../
0477 002CB          *
0478 002CB 24               P=      4       sauver 5 niveaux
0479 002CD 8F0000           GOSBVL  =R>RSTK
0480 002D4
0481 002D4 1B0000           D0=(5)  sauve   qui peut !
0482 002DB 110              C=R0            R0(A)
0483 002DE 144              DAT0=C  A
0484 002E1 164              D0=D0+  5       R1(A)
0485 002E4 113              C=R1
0486 002E7 141              DAT0=C  A
0487 002EA
0488 002EA 8E0000           GOSUBL  =GETSRC compiler la chaine de recherche
0489 002F0 416              GOC     erreur
0490 002F3 702D             GOSUB   =getchr passer le délimiteur (88/10/15)
0491 002F7          *
0492 002F7          * R0^7:0: caractéristiques du buffer
0493 002F7          * D1 et D(A): réactualisés
0494 002F7          *
0495 002F7 1B0000           D0=(5)  =sauve+10
0496 002FE 15F              CD1EX
0497 00301 141              DAT0=C  A
0498 00304 164              D0=D0+  5
0499 00307 1F               C=C     A
0500 00309 141              DAT0=C  A
```

```
Page 013          Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4   Routines de support

0571 003A1
0572 003A1        ********************************************************
0573 003A1        • GETFIL
0574 003A1        •
0575 003A1        • But: lire un nom de fichier, chercher ce fichier ou le
0576 003A1        •   créer selon les flags.
0577 003A1        • Entree:
0578 003A1        •   - D1 = ' chaine
0579 003A1        •   - D(A) = nb de caractères
0580 003A1        •   - sDELET = 0 : cas C/M
0581 003A1        •   - sDELET = 1 : cas D ou X
0582 003A1        •       - sXCHG = 1 : cas X
0583 003A1        •       - sXCHG = 0 : cas D
0584 003A1        • Sortie:
0585 003A1        •   - Cy = 1 : erreur
0586 003A1        •       C(3-0) = numéro d'erreur
0587 003A1        •   - Cy = 0 : pas d'erreur
0588 003A1        •       EXTFIL = adresse du header du fichier
0589 003A1        •       sCREAT = 1 : fichier crée
0590 003A1        • Abime: A-D, R0-R3, D0, D1, S0-S9, STMTxx, FUNCxx
0591 003A1        • Appelle: ADHEAD, FILXQ$, FINDF+, CRETF+
0592 003A1        • Niveaux: 6 (CRETF+)
0593 003A1        • Algorithme:
0594 003A1        •   création d'une chaine sur la M.S. contenant le nom
0595 003A1        •     (la chaine s'arrete sur '+', ':' ou EOL)
0596 003A1        •   si DELET = 1
0597 003A1        •     alors
0598 003A1        •       si XCHG = 1
0599 003A1        •         alors plus := true
0600 003A1        •         sinon plus := (caractère lu == '+')
0601 003A1        •       fin si
0602 003A1        •   fin si
0603 003A1        •   FILXQ$
0604 003A1        •   FINDF+
0605 003A1        •   si il existe
0606 003A1        •     alors
0607 003A1        •       si DELET=1 et not plus
0608 003A1        •         alors erreur (File Exists)
0609 003A1        •       fin si
0610 003A1        •     sinon
0611 003A1        •       si DELET=0
0612 003A1        •         alors erreur (File Not Found)
0613 003A1        •         sinon CRETF+
0614 003A1        •       fin si
0615 003A1        •   fin si
0616 003A1        •   vérifier que le type est TEXT
0617 003A1        •   retourner l'adresse dans EXTFIL
0618 003A1        • Historique:
0619 003A1        •   88/10/31: PD/JT conception & codage
0620 003A1        •   88/11/11: PD/JT modification pour la commande 'X'
```

```
Page 014          Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4   Routines de support

0621 003A1        ********************************************************
0622 003A1
0623 003A1        =sDELET EQU     09      flag D (par opposition à C/M)
0624 003A1        =sXCHG  EQU     10      cas D ou X ? (apres : '+' reconnu)
0625 003A1        =sCREAT EQU     11      fichier créé par nos soins
0626 003A1
0627 003A1         sPLUS  EQU     =sXCHG  flag plus utilisé apres son test
0628 003A1
0629 003A1 84B    =GETFIL ST=0    =sCREAT fichier non encore créé
0630 003A4 1B0000         D0=(5)  =EXTFIL par défaut, EXTFIL := 0
0631 003AB DC             C=0     A
0632 003AD 144            DAT0=C  A
0633 003B0
0634 003B0 1B0000         D0=(5)  =AVMEMS
0635 003B7 14E            C=DAT0  A
0636 003BA D5             B=C     A       B(A) := AVMEMS
0637 003BC 164            D0=D0+  5       D0=(5)  =AVMEME
0638 003BF 146            C=DAT0  A
0639 003C2 134            D0=C            D0 := ' début de la M.S.
0640 003C5 103            R1=C            pour ADHEAD tout à l'heure
0641 003C8
0642 003C8 7A5C     GF010 GOSUB   =lookah
0643 003CC 483            GOC     GF100   EOL
0644 003CF 3582B302       LCASC   ' :+'   ensemble (espace, ':', '+')
0645 003D7 25             P=      3+2-1
0646 003D9 8F0000         GOSBVL  =MEMBER n'abime que C(WP)
0647 003E0 542            GONC    GF100   byte in set
0648 003E3        •
0649 003E3        • Mettre le caractère dans la M.S. (pointée par D0)
0650 003E3        •
0651 003E3 703C           GOSUB   =getchr passer le caractère
0652 003E7        • Ce qui suit est très fortement inspiré de STKCHR (#18505)
0653 003E7 181            D0=D0-  2
0654 003EA 136            CD0EX
0655 003ED 8B1            ?C#B    A
0656 003F0 B0             GOYES   GF090   Memerr
0657 003F2 136            CD0EX
0658 003F5 148            DAT0=A  B
0659 003F8 5FC            GONC    GF01#   B.E.T.
0660 003FB
0661 003FB 330000   GF090 LC(4)   =eMEM
0662 00401 02             RTNSC           Beeeeeeep !
0663 00403
0664 00403 03       rtncc RTNCC
0665 00405        •
0666 00405        • Fin de la chaine. Si la longueur est nulle, alors sortir
0667 00405        • sans avoir rien reconnu.
0668 00405        •
0669 00405 136      GF100 CD0EX           C(A) := ' haut de la M.S.
0670 00408 111            A=R1            A(A) :=  bas de la M.S.
```

```
Page 015          Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4   Routines de support

0671 0040B 8A2            ?A=C    A
0672 0040E 5F             GOYES   rtncc   sortie sans erreur
0673 00410 134            D0=C            sauvegarde temporaire dans D0
0674 00413        •
0675 00413        • Le nom du fichier a été reconnu et est dans la M.S.
0676 00413        •
0677 00413 79EB           GOSUB   =skip
0678 00417        •
0679 00417        • Reconnaitre le '+' seulement si 'D' (et pas C/M ou X)
0680 00417        • si sDELET = 1
0681 00417        •   alors
0682 00417        •     si sXCHG = 1
0683 00417        •       alors plus := true
0684 00417        •       sinon plus := '+' reconnu ? true : false
0685 00417        •     fin si
0686 00417        • fin si
0687 00417        •
0688 00417 869            ?ST=0   =sDELET cas 'D' ?
0689 0041A F1             GOYES   GF150   non
0690 0041C 87A            ?ST=1   =sXCHG  cas 'X' ?
0691 0041F A1             GOYES   GF150   oui, donc plus := true (déja vrai)
0692 00421        •
0693 00421        • On sait que XCHG = 0. Donc sPLUS = 0
0694 00421        • l'instruction "ST=0   sPLUS" est inutile"
0695 00421        •
0696 00421 710C           GOSUB   =lookah
0697 00425 31B2           LCASC   '+'
0698 00429 966            ?A#C    B       caractère suivant = '+' ?
0699 0042C D0             GOYES   GF150   non
0700 0042E        •
0701 0042E        • '+' reconnu
0702 0042E        •
0703 0042E 75EB           GOSUB   =getchr passer le '+'
0704 00432 7ACB           GOSUB   =skip
0705 00436 85A            ST=1    sPLUS   plus := true
0706 00439        •
0707 00439        • Sauver les meubles !!! (D1 et D(A))
0708 00439        •
0709 00439 137      GF150 CD1EX
0710 0043C 1F00000        D1=(5)  =STMTR0
0711 00443 145            DAT1=C  A       STMTR0+00 := D1
0712 00446 174            D1=D1+  5
0713 00449 DB             C=D     A
0714 0044B 145            DAT1=C  A       STMTR0+05 := D(A)
0715 0044E
0716 0044E        •
0717 0044E        • D0 = ' haut de la M.S.
0718 0044E        • R1 = ' bas de la M.S.
0719 0044E        • sDELET = 1 si commande 'D', 0 si commande 'C' ou 'M'
0720 0044E        • sPLUS = 1 si commande 'D' et '+' reconnu
```

```
Page 016          Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4   Routines de support

0721 0044E        • STMTR0(4-0) = sauvegarde de D1
0722 0044E        • STMTR0(9-5) = sauvegarde de D(A)
0723 0044E        •
0724 0044E
0725 0044E        •
0726 0044E        • Construire le M.S. header et chercher le fichier
0727 0044E        •
0728 0044E 136      .     CD0EX
0729 00451 135            D1=C            D1 = ' haut de la M.S.
0730 00454 1B0000         D0=(5)  =AVMEMS
0731 0045B 146            C=DAT0  A
0732 0045E D7             D=C     A       D(A) := AVMEMS
0733 00460 850            ST=1    0       rtn desired
0734 00463        •
0735 00463        • D1 = ' top of M.S.
0736 00463        • R1 = ' bottom of M.S.
0737 00463        • D(A) = AVMEMS
0738 00463        • S0=1
0739 00463        •
0740 00463 8F0000         GOSBVL  =ADHEAD D1 = ' string header
0741 00468
0742 0046A 8F0000         GOSBVL  =FILXQ$
0743 00471 4A0            GOC     GF160   pas d'erreur
0744 00474
0745 00474 330000   ivfspc LC(4)   =eFSPEC
0746 0047A 02             RTNSC
0747 0047C
0748 0047C          GF160
0749 0047C        •
0750 0047C        • Spécificateur valide.
0751 0047C        • A(W) = file name ou 0
0752 0047C        • D(S) = F if no device specified
0753 0047C        •        0 if :MAIN
0754 0047C        •        1 if :PORT
0755 0047C        •        7 if :CARD
0756 0047C        • La vérification du spécificateur sera faite par FINDF+
0757 0047C        • Maintenant, il faut chercher le fichier.
0758 0047C        • A(W) = le nom
0759 0047C        • D(S) et D(B) = device specifier
0760 0047C        •
0761 0047C
0762 0047C        •
0763 0047C        • Sauvegarde de A(W) et D(S)-D(B)
0764 0047C        • dans STMTR0(10-25) et STMTR0(28-2B)
0765 0047C        •
0766 0047C 1B0000         D0=(5)  10+=STMTR0
0767 00483 1507           DAT0=A  W
0768 00487 16F            D0=D0+  16
0769 0048A AFB            C=D     W
0770 0048D 812            CSLC            C(2-0) = D(S) + D(B)
```

```
Page 017              Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4       Routines de support

0771 00490 15C2              DAT0=C  3
0772 00494
0773 00494           *
0774 00494           * Chercher le fichier
0775 00494           *
0776 00494 8F00000           GOSBVL  =FINDF+
0777 0049B 402               GOC     GF300   fichier n'existe pas
0778 0049E           *
0779 0049E           * Le fichier existe
0780 0049E           *
0781 0049E           * D1 = ^ header du fichier
0782 0049E           *
0783 0049E 869        GF200  ?ST=0   =sDELET pour les commandes 'C' et 'M'
0784 004A1 70                GOYES   GF205   le fichier doit exister
0785 004A3 86A               ?ST=0   sPLUS   si D et pas +, il doit pas exister
0786 004A6 60                GOYES   fexist  pas plus => erreur
0787 004A8 6080       GF205  GOTO    GF900   rallonge
0788 004AC
0789 004AC 330000     fexist LC(4)   =eFEXST "File Exists"
0790 004B2 02                RTNSC
0791 004B4 330000     fnfnd  LC(4)   =eFnFND "File Not Found"
0792 004BA 02                RTNSC
0793 004BC
0794 004BC           *
0795 004BC           * Le fichier n'existe pas
0796 004BC           *
0797 004BC 869        GF300  ?ST=0   =sDELET commande 'C' ou 'M' ?
0798 004BF 5F                GOYES   fnfnd   oui : "File Not Found"
0799 004C1           *
0800 004C1           * Le fichier n'existe pas, mais il faut le créer pour la
0801 004C1           * commande 'D'
0802 004C1           *
0803 004C1 1B00000           D0=(5)  16+10+=STMTR0
0804 004C8 15E2              C=DAT0  3
0805 004CC 816               CSRC
0806 004CF AF7               D=C     W       D(W) := comme apres FILXQ$
0807 004D2
0808 004D2 D2                C=0     A
0809 004D4 3192              LC(2)   37+4    header + EOF mark
0810 004D8 8F00000           GOSBVL  =MEMCKL
0811 004DF 400               RTNC            Not Enough Memory
0812 004E2
0813 004E2 D9                C=B     A       C(A) := amount to check (MEMCKL)
0814 004E4           *
0815 004E4           * C(A) = taille du buffer en nibs
0816 004E4           * D = conditions de sortie de FILXQ$
0817 004E4           *
0818 004E4 8F00000           GOSBVL  =CRETF+ Théoriq., le Lex ne doit pas bouger
0819 004EB 400               RTNC            erreur !
0820 004EE
```

```
Page 018              Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4       Routines de support

0821 004EE 85B               ST=1    =sCREAT fichier cree !
0822 004F1           *
0823 004F1           * R1 = ^ début du nouveau fichier
0824 004F1           * manque le nom et le type
0825 004F1           *
0826 004F1 119               C=R1
0827 004F4 1F00000           D1=(5)  10+=STMTR0
0828 004FB 1537              A=DAT1  W
0829 004FF 135               D1=C
0830 00502 1517              DAT1=A  W       nom du fichier
0831 00506 17F               D1=D1+  16
0832 00509 35100004          LCHEX   400001  text + copy code
0833 00511 15D5              DAT1=C  6
0834 00515 17F               D1=D1+  16
0835 00518 174               D1=D1+  5
0836 0051B D2                C=0     A
0837 0051D CE                C=C-1   A       C(3-0) := #FFFF
0838 0051F 15D3              DAT1=C  4       EOF mark
0839 00523
0840 00523 119               C=R1            C(A) := ^ début du fichier
0841 00526 135               D1=C            D1 := ^ début du fichier
0842 00529
0843 00529           *
0844 00529           * D1 = ^ header du fichier trouvé ou créé
0845 00529           * STMTR0(9-0) = sauvegarde de D1 et D(A)
0846 00529           *
0847 00529            GF900
0848 00529 137               CD1EX
0849 0052C 1F00000           D1=(5)  =EXTFIL
0850 00533 145               DAT1=C  A
0851 00536           *
0852 00536           * Vérifier que le fichier est bien TEXT
0853 00536           *
0854 00536 135               D1=C
0855 00539 8E0000            GOSUBL  =CHKTXT
0856 0053F 400               RTNC            C(A) = eFTYPE
0857 00542           *
0858 00542           * Restaurer les meubles (D1 et D(A))
0859 00542           *
0860 00542 1B00000    GF910  D0=(5   =STMTR0
0861 00549 146               C=DAT0  A
0862 0054C 135               D1=C
0863 0054F 164               D0=D0+  5
0864 00552 146               C=DAT0  A
0865 00555 D7                D=C     A
0866 00557 03                RTNCC           sortie sans erreur
0867 00559
0868 00559                   END
```

```
Page 019              Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4       **** SYMBOL TABLE ****

=ADHEAD     Extrn Ukn        -  0740
=AVMEMS     Extrn Ukn        -  0634 0730
=CHKTXT     Extrn Ukn        -  0855
=CONVUC     Extrn Ukn        -  0271
=COUR       Extrn Ukn        -  0179 0303 0459
=CRETF+     Extrn Ukn        -  0818
=DERN       Extrn Ukn        -  0310 0466 0505
=DRANGE     Extrn Ukn        -  0343 0348
=ENDPOS     Extrn Ukn        -  0529
=EXTFIL     Extrn Ukn        -  0630 0849
=FILXQ$     Extrn Ukn        -  0742
=FINDF+     Extrn Ukn        -  0776
 FiLeNd     00559 Rel  0868  -
=GETFIL     003A1 Rel  0629  -
=GETSRC     Extrn Ukn        -  0488
 GF010      003C8 Rel  0642  -  0659
 GF090      003FB Rel  0661  -  0656
 GF100      00405 Rel  0669  -  0643 0647
 GF150      00439 Rel  0709  -  0689 0691 0699
 GF160      0047C Rel  0748  -  0743
 GF200      0049E Rel  0783  -
 GF205      004A8 Rel  0787  -  0784
 GF300      004BC Rel  0797  -  0777
 GF900      00529 Rel  0847  -  0787
 GF910      00542 Rel  0860  -
 Ivarg      00274 Rel  0427  -  0424
=LCMD       Extrn Ukn        -  0100
=MEMBER     Extrn Ukn        -  0646
=MEMCKL     Extrn Ukn        -  0810
=MODE71     Extrn Ukn        -  0095
=PCMD       Extrn Ukn        -  0103
=R<RSTK     Extrn Ukn        -  0479
=RSTK<R     Extrn Ukn        -  0547
=SCRTCH     Extrn Ukn        -  0006
=SRCLIN     Extrn Ukn        -  0509 0518
=STMTR0     Extrn Ukn        -  0011 0012 0710 0766 0803 0827 0860
 debut      Unkwn Ukn  0012  -  0181 0193 0502 0514
 dec020     0009A Rel  0156  -  0144
 dec030     000A1 Rel  0162  -  0152
 dec050     000A8 Rel  0164  -  0131 0141 0157
 dec100     000AC Rel  0170  -  0121
 dec120     00100 Rel  0216  -  0205
 dec130     00107 Rel  0222  -  0212
 dec200     0010E Rel  0232  -  0164 0186 0202 0217
 dec300     0012C Rel  0246  -  0240 0243
 dec320     00143 Rel  0260  -  0253
 dec350     0014C Rel  0266  -  0259
 dec900     00162 Rel  0277  -  0262
=eFEXST     Extrn Ukn        -  0789
=eFSPEC     Extrn Ukn        -  0745
```

```
Page 020              Editeur, décomposition des cmd <xdec.as>
AREUH ASS. V2.4       **** SYMBOL TABLE ****

=eFnFND     Extrn Ukn        -  0791
=eIVARG     Extrn Ukn        -  0369
=eMEM       Extrn Ukn        -  0661
=eMSPAR     Extrn Ukn        -  0153 0213
=eNFND      Extrn Ukn        -  0521
=eSYNTX     Extrn Ukn        -  0413
 erreur     00352 Rel  0522  -  0154 0214 0370 0414 0489
 ex1010     002A5 Rel  0462  -  0457
 ex1020     002BE Rel  0469  -  0464
 ex1030     002CB Rel  0474  -  0471
 ex1035     00330 Rel  0512  -  0508
 ex1040     00356 Rel  0528  -  0510 0519
 expelm     00284 Rel  0453  -  0393 0411
 expr       001FA Rel  0393  -  0185 0211 0216
 expr00     00201 Rel  0395  -  0423 0426
 expr10     0022E Rel  0408  -  0403
 expr20     0024E Rel  0415  -  0412
 expr30     0026F Rel  0425  -  0420
 expr99     00278 Rel  0429  -  0399 0407
 fexist     004AC Rel  0789  -  0786
 fnfnd      004B4 Rel  0791  -  0798
=getchr     00017 Rel  0056  -  0035 0146 0206 0244 0266 0302 0309 0350 0410
                             +  0458 0465 0490 0651 0703
=id         Extrn Ukn        -  0521
 ivarg      001EC Rel  0369  -  0357 0427
 ivfspc     00474 Rel  0745  -
=lookah     00026 Rel  0063  -  0031 0110 0140 0201 0239 0252 0298 0341 0346
                             +  0398 0454 0642 0696
 linenb     00164 Rel  0297  -  0130 0151 0156
 lnb010     00185 Rel  0306  -  0301
 lnb020     0019E Rel  0313  -  0308
=nombre     0019E Rel  0341  -  0472
 nbli++     00397 Rel  0565  -  0137 0163 0188 0223
 nbr010     001B1 Rel  0346  -  0368
 nbr090     001F6 Rel  0371  -  0347 0349
 notfnd     0034C Rel  0521  -  0517
 rtncc      00403 Rel  0664  -  0672
=sCREAT     0000B Abs  0625  -  0629 0821
=sDELET     00009 Abs  0623  -  0688 0783 0797
=sXCHG      0000A Abs  0624  -  0627 0690
=skip       00000 Rel  0030  -  0109 0139 0232 0251 0268 0297 0397 0453 0677
                             +  0704
 sPLUS      0000A Abs  0627  -  0705 0785
 sauve      Unkwn Ukn  0011  -  0481 0495 0533
 signe      Unkwn Ukn  0009  -  0403 0415
 skip10     00004 Rel  0031  -  0036
 tmp        Unkwn Ukn  0006  -  0008 0009
 val        Unkwn Ukn  0008  -  0395 0417 0429
=xeddec     00030 Rel  0095  -
=xederr     Extrn Ukn        -  0522
```

Source : xdec.as

Object : obj/xdec.ao

Listing : list/xdec.al

Date : Sat Aug 12 17:40:57 1989

Errors : 000

Areuh Assembler/Linker V2.4, (c) P. David & J. Taillandier 1986 Paris, France

```
Page 001          Editeur, boucle principale <xbcl.as>
AREUH ASS. V2.4

0001 00000                    TITLE   Editeur, boucle principale <xbcl.as>
0002 00000
0003 00000          •
0004 00000          • Variables utilisées par les fonctions et par XEDIT
0005 00000          • Tout le reste de TRFMBF est donc disponible
0006 00000          •
0007 00000          =MODE71 EQU     00+=TRFMBF      1q si mode = HP-71
0008 00000          =QUERY  EQU     01+=TRFMBF      1q si mode = "?" pour S/R
0009 00000          =FCNTMP EQU     02+=TRFMBF      58 quartets pour le scratch
0010 00000
0011 00000          •
0012 00000          • Variables utilisées par XEDIT seulement.
0013 00000          •
0014 00000          =BUFID  EQU     02+=TRFMBF      3q
0015 00000          =BUFADR EQU     05+=TRFMBF      5q
0016 00000
0017 00000          =FILADR EQU     10+=TRFMBF      5q
0018 00000          =COUR   EQU     15+=TRFMBF      5q no de ligne courante
0019 00000          =DERN   EQU     20+=TRFMBF      5q no de la dernière ligne
0020 00000          =PCMD   EQU     25+=TRFMBF      5q pteur dans la cmd
0021 00000          =LCMD   EQU     30+=TRFMBF      5q longueur de la cmd
0022 00000          =FLAG1  EQU     35+=TRFMBF      1q état du flag 1 à l'entrée
0023 00000          =RDONLY EQU     36+=TRFMBF      1q fichier read-only
0024 00000
0025 00000          •
0026 00000          • Tout le reste (à savoir 23 quartets) est disponible
0027 00000          • pour des variables temporaires dans XEDIT
0028 00000          •
0029 00000          =XEDTMP EQU     37+=TRFMBF      23q : variables temporaires
0030 00000
0031 00000          •
0032 00000          • En particulier, la fonction GENRPLC$ et la commande R
0033 00000          • utilisent 10 quartets pour stocker les coefficients "a" et
0034 00000          • "b" utilisés lors du remplacement. Voir la documentation
0035 00000          • CALCab.
0036 00000          • Cette zone peut être utilisée par toute routine
0037 00000          • n'utilisant pas REPLIN ou REPLFIL.
0038 00000          •
0039 00000          =coeffa EQU     (=SCRTCH)-10
0040 00000          =coeffb EQU     (=coeffa)+5
0041 00000
0042 00000          •
0043 00000          • Diverses zones de scratch
0044 00000          •
0045 00000          =TMP5   EQU     =LDCSPC
0046 00000          =TMP2   EQU     =STSAVE
0047 00000          =TMP1   EQU     (=SCRTCH)-1
0048 00000
0049 00000
0050 00000          •
```

```
Page 002          Editeur, boucle principale <xbcl.as>
AREUH ASS. V2.4

0051 00000          • Constantes globales :
0052 00000          •
0053 00000          =FLIMSK EQU     #0010   flag 1 = 1 dans les flags 3-0
0054 00000          =NCMDED EQU     13      nombres de commandes (0-13)
0055 00000
0056 00000          ********************************************************
0057 00000          • poparg
0058 00000          •
0059 00000          • But : dépiler un nombre de la M.S. et le vérifier
0060 00000          • Entree :
0061 00000          •   - D1 = ^ top of M.S.
0062 00000          • Sortie :
0063 00000          •   - A(A) = C(A) = nombre dépilé (0 < nb <= 255)
0064 00000          • Abime: A, B(S), B(A), C(A), D(A)
0065 00000          • Appelle: RNDAHX
0066 00000          • Niveaux: 4 (RNDAHX)
0067 00000          • Historique:
0068 00000          •   88/05/13: PD/JT conception & codage
0069 00000          ********************************************************
0070 00000
0071 00000 8F00000  poparg GOSBVL  =RNDAHX
0072 00027 511              GONC    ivarg   L < 0
0073 0002A 2A8              ?A=0    A
0074 0002D C0               GOYES   ivarg
0075 0002F D2               C=0     A
0076 00011 AE6              C=A     B
0077 00014 8A2              ?A=C    A
0078 00017 00               RTNYES
0079 00019 8D00000  ivarg   GOVLNG  =IVAERR
0080 00020
0081 00020          ********************************************************
0082 00020          • nxtexp
0083 00020          •
0084 00020          • But: évaluer la prochaine expression
0085 00020          • Entree:
0086 00020          •   - D0 = ^ chaine tokenisée (sur tCOMMA ou tEOL)
0087 00020          • Sortie:
0088 00020          •     cy = 1 si tEOL reconnu
0089 00020          •   - cy = 0 et expression sur la M.S. (A(W) = top 16 nibs)
0090 00020          • Abime: tous les registres, FUNction scratch
0091 00020          • Appelle: EOLxC?, EXPEXC
0092 00020          • Niveaux: 4 (EXPEXC)
0093 00020          • Historique:
0094 00020          •   88/05/13: PD/JT conception & codage
0095 00020          ********************************************************
0096 00020
0097 00020 14A      nxtexp A=DAT0  B
0098 00023 161              D0=D0+  2
0099 00026 8F00000          GOSBVL  =EOLxC?
0100 00029 400              RTNC            tEOL trouvé
```

```
Page 003          Editeur, boucle principale <xbcl.as>
AREUH ASS. V2.4

0101 00030 8D00000          GOVLNG  =EXPEXC Cy = 0 en sortie
0102 00037
0103 00037          ********************************************************
0104 00037          • XEDITe, TEDITe
0105 00037          •
0106 00037          • But: éditeur de texte
0107 00037          • Entree:
0108 00037          •   - le nom du fichier
0109 00037          •   - une chaine de commande optionnelle
0110 00037          • Abime: tout
0111 00037          • Appelle: ????????
0112 00037          • Niveaux: 7 niveaux sont autorisés pour les statements
0113 00037          • Historique:
0114 00037          •   88/05/13: PD/JT conception & codage
0115 00037          •   89/06/17: PD/JT retrait des paramètres largeur/hauteur
0116 00037          ********************************************************
0117 00037
0118 00037 00000            REL(5)  =TEDITd
0119 0003C 00000            REL(5)  =TEDITp
0120 00041 850      =TEDITe ST=1    0       Mode HP-71 := true
0121 00044 6010             GOTO    XED000
0122 00048
0123 00048 00000            REL(5)  =XEDITd
0124 0004D 00000            REL(5)  =XEDITp
0125 00052 840      =XEDITe ST=0    0       Mode HP-71 := false
0126 00055 1F00000  XED000  D1=(5)  =TRFMBF
0127 0005C 34C3000          LC(5)   60      60 quartets
0128 00063 8F00000          GOSBVL  =WIPOUT TRFMBF rempli avec des 0
0129 0006A
0130 0006A 860              ?ST=0   0       mode HP-71 ?
0131 0006D 01               GOYES   XED010  non : on saute
0132 0006F 301              LC(1)   1
0133 00072 1F00000          D1=(5)  =MODE71
0134 00079 1550             DAT1=C  P       MODE71 := 1
0135 0007D          XED010
0136 0007D          •
0137 0007D          • Analyse du fichier
0138 0007D          •
0139 0007D 8E0000           GOSUBL  =FILE1  Evalue la spécification
0140 00083
0141 00083          •
0142 00083          • Analyse de la chaine de commande
0143 00083          •
0144 00083 799F             GOSUB   nxtexp  Expression suivante
0145 00087 443              GOC     XED050  tEOL trouvé
0146 0008A 8F00000          GOSBVL  =POP1S
0147 00091 137              CD1EX
0148 00094 1F00000          D1=(5)  =AVMEME
0149 0009B 145              DAT1=C  A       AVMEME := ^ bas de la M.S.
0150 0009E C2               C=C+A   A       C(A) := ^ début de la chaine
```

```
Page 004          Editeur, boucle principale <xbcl.as>
AREUH ASS. V2.4

0151 000A0 CE               C=C-1   A
0152 000A2 CE               C=C-1   A
0153 000A4 1E0000           D1=(4)  =PCMD
0154 000AA 145              DAT1=C  A       PCMD := ^ premier caractère
0155 000AD AF2              C=0     W
0156 000B0 D6               C=A     A
0157 000B2 81E              CSRB            C(A) := longueur en octets
0158 000B5 1D00             D1=(2)  =LCMD
0159 000B9 145              DAT1=C  A       LCMD := longueur en octets
0160 000BC          •
0161 000BC          • Recherche du fichier, et copie éventuelle
0162 000BC          •
0163 000BC          XED050
0164 000BC 851              ST=1    1       Copie éventuelle
0165 000BF 852              ST=1    2       Création éventuelle
0166 000C2 8E0000           GOSUBL  =FILE2
0167 000C8          •
0168 000C8          • Le fichier est trouvé. On a son adresse dans C(A)
0169 000C8          •
0170 000C8 8E0000           GOSUBL  =CHKWRT Modifiable ?
0171 000CE 1F00000          D1=(5)  =RDONLY
0172 000D5 1554             DAT1=C  S       RDONLY := 0 (write) 1..3 (read-only)
0173 000D9
0174 000D9 1E0000           D1=(4)  =FILADR Adresse de l'en-tete
0175 000DF 145              DAT1=C  A       FILADR := adresse de l'en-tete
0176 000E2
0177 000E2 135              D1=C            D1 := ^ file header
0178 000E5
0179 000E5 8E0000           GOSUBL  =CHKTXT Est-ce un fichier TEXT ?
0180 000EB 530              GONC    XED080  Oui
0181 000EE
0182 000EE 8D00000  bserr   GOVLNG  =BSERR
0183 000F5
0184 000F5          XED080
0185 000F5          •
0186 000F5          • inibuf initialise le tableau de pointeurs dans le texte.
0187 000F5          • Cette routine commence par désallouer le buffer dont le
0188 000F5          • numéro est dans BUFID. Cela ne nous gène pas car nous
0189 000F5          • avons fait un WIPOUT sur toutes les variables au début.
0190 000F5          •
0191 000F5 8E0000           GOSUBL  =inibuf
0192 000FB
0193 000FB D2               C=0     A
0194 000FD E6               C=C+1   A
0195 000FF 1F00000          D1=(5)  =COUR   courante := 1
0196 00106 145              DAT1=C  A
0197 00109
0198 00109          •
0199 00109          • Mettre à 0 le flag 1 et sauvegarder son ancien état.
0200 00109          •
```

```
Page 005          Editeur, boucle principale <xbcl.as>
AREUH ASS. V2.4

0201 00109 1E0000           D1=(4)  =FLGREG
0202 0010F 1530             A=DAT1  P
0203 00113 A88              B=A     P       B(0) := ancienne valeur du flag 1
0204 00116 30D              LC(1)   '=FL1MSK
0205 00119 0E0E             A=A&C   P       A(0) := flags 0-3 sans le 1
0206 0011D 1510             DAT1=A  P
0207 00121
0208 00121 302              LC(1)   =FL1MSK
0209 00124 0E05             C=C&B   P       C(0) := flag 1 seulement
0210 00128 1E0000           D1=(4)  =FLAG1
0211 0012E 1550             DAT1=C  P
0212 00132
0213 00132 6A80             GOTO    =BOUCLE
0214 00136
0215 00136
0216 00136          ********************************************************
0217 00136          * viewln
0218 00136          *
0219 00136          * But: affiche la ligne en cours ou "[EOf]" tant que la
0220 00136          *   touche [ENDLINE] est appuyée.
0221 00136          * Entree:
0222 00136          *   - COUR = numéro de la ligne courante
0223 00136          *   - DERN = numéro de la dernière ligne
0224 00136          * Sortie: -
0225 00136          * Abime: A-D, R0-R3, D0, D1, STMTR0, AVMEMS, TMP5
0226 00136          * Appelle: OBCOLL, Num2D1, AVS2DS, seek, dsplin, CRLFND,
0227 00136          *   dspmsg, KYDN?
0228 00136          * Niveaux: 6 (CRLFND)
0229 00136          * Historique:
0230 00136          *   88/11/13: PD/JT conception & codage
0231 00136          ********************************************************
0232 00136
0233 00136           viewln
0234 00136          *
0235 00136          * "<numero>: ligne" ou "[EOf]" ?
0236 00136          *
0237 00136 1B00000          D0=(5)  =DERN
0238 0013D 142              A=DAT0  A       A(A) := dernière
0239 00140 1900             D0=(2)  =COUR
0240 00144 146              C=DAT0  A       C(A) := courante
0241 00147 8B2              ?C#A    A
0242 0014A 75               GOYES   nul800  attendre le relachement de [ENDLINE]
0243 0014C          *
0244 0014C          * Afficher le numéro de ligne suivi d'un ":"
0245 0014C          *
0246 0014C 8F00000          GOSBVL  =OBCOLL C(A) := OUTBS
0247 00153 135              D1=C            D1 := ^ AVMEMS
0248 00156
0249 00156 1B00000          D0=(5)  =COUR
0250 0015D 142              A=DAT0  A       A(A) := courante
```

```
Page 007          Editeur, boucle principale <xbcl.as>
AREUH ASS. V2.4

0301 001C0          *
0302 001C0 8E0000           GOSUBL  =xeddec
0303 001C6 581              GONC    BCL200  Commande non nulle
0304 001C9          *
0305 001C9          * Commande nulle. Si on vient d'afficher notre prompt,
0306 001C9          * c'est le cas spécial de l'utilisateur qui appuie sur
0307 001C9          * [ENDLINE] pour avoir la ligne courante à l'affichage.
0308 001C9          *
0309 001C9 86A              ?ST=0   10      Ce n'est pas le cas
0310 001CC 60               GOYES   BCL100  donc affichage du prompt
0311 001CE          *
0312 001CE          * C'est le cas. Afficher la ligne tant que [ENDLINE]
0313 001CE          * est appuyée.
0314 001CE          *
0315 001CE 746F             GOSUB   viewln
0316 001D2
0317 001D2          *
0318 001D2          * Afficher un prompt
0319 001D2          *
0320 001D2 8E0000    BCL100 GOSUBL  =prompt
0321 001D8 85A              ST=1    10
0322 001DB 64EF             GOTO    BCL000
0323 001DF
0324 001DF 31FF      BCL200 LC(2)   -1      commande nulle ?
0325 001E3 962              ?A=C    B
0326 001E6 1D               GOYES   cmNULL
0327 001E8          *
0328 001E8          * Une commande a été fournie. La chercher dans le message
0329 001E8          * listant les commandes valides.
0330 001E8          *
0331 001E8 1B00000          D0=(5)  =FUNCR0
0332 001EF DB               C=D     A
0333 001F1 144              DAT0=C  A       FUNCR0 + 0 := D(A)
0334 001F4 164              D0=D0+  5
0335 001F7 137              CD1EX
0336 001FA 144              DAT0=C  A       FUNCR0 + 5 := D1
0337 001FD 164              D0=D0+  5
0338 00200 118              C=R0
0339 00203 144              DAT0=C  A       FUNCR0 + 10 := R0
0340 00206 164              D0=D0+  5
0341 00209 119              C=R1
0342 0020C 144              DAT0=C  A       FUNCR0 + 15 := R1
0343 0020F
0344 0020F 8E0000           GOSUBL  =poscmd
0345 00215 546              GONC    invcmd  "Invalid Cmd"
0346 00218          *
0347 00218          * La commande a été reconnue et acceptée.
0348 00218          * Ce qui suit est un "hack" basé sur le fait qu'il y a
0349 00218          * moins de 16 commandes dans l'éditeur. Tout tient donc
0350 00218          * sur un seul quartet.
```

```
Page 006          Editeur, boucle principale <xbcl.as>
AREUH ASS. V2.4

0251 00160
0252 00160 8E0000           GOSUBL  =Num2D1 place A(A) en FUNCR0
0253 00166 31A3             LCASC   ':'
0254 0016A 14D              DAT1=C  R
0255 0016D 171              D1=D1+  2       *D1++ := ':'
0256 00170 AE0              A=0     B
0257 00173 A6C              A=A-1   B       A(A):= #FF
0258 00176 149              DAT1=A  B       *D1 := #FF
0259 00179
0260 00179 8F00000          GOSBVL  =AVS2DS
0261 00180          *
0262 00180          * Chercher la ligne courante et l'afficher
0263 00180          *
0264 00180 1E0000           D0=(5)  =COUR
0265 00187 146              C=DAT0  A       (A) := courante
0266 0018A 8E0000           GOSUBL  =seek   D0 := ^ longueur LIF
0267 00190 8E0000           GOSUBL  =dsplin
0268 00196 8F0000           GOSBVL  =CRLFND
0269 0019D 6D00             GOTO    nul900
0270 001A1
0271 001A1 330000    nul800 LC(4)   (=id)^(=teEOF)
0272 001A7 7021             GOSUB   =dspmsg
0273 001AB
0274 001AB          *
0275 001AB          * Attend le relachement de toutes les touches.
0276 001AB          *
0277 001AB 8F0000    nul900 GOSBVL  =KYDN?
0278 001B2 58F              GONC    nul900
0279 001B5 01               RTN
0280 001B7
0281 001B7          ********************************************************
0282 001B7          * BOUCLE
0283 001B7          *
0284 001B7          * But: la boucle principale de l'éditeur
0285 001B7          * Entree:
0286 001B7          *   - PCMD = pointeur sur le premier caractère de la ligne
0287 001B7          *   - LCMD = longueur de cette ligne de commande
0288 001B7          * Historique:
0289 001B7          *   88/05/14: PD/JT conception & codage
0290 001B7          *   88/05/15: PD/JT séparation de "xeddec" et "prompt"
0291 001B7          *   88/11/06: PD/JT aiguillage selon le message "teVCMD"
0292 001B7          *   88/11/13: PD/JT ajout de viewln
0293 001B7          ********************************************************
0294 001B7
0295 001B7 8C0000    cmNULL GOLONG  =cmNULL Rallonge
0296 001BD
0297 001BD 84A      =BOUCLE ST=0    10
0298 001C0           BCL000
0299 001C0          *
0300 001C0          * Décomposition de la commande
```

```
Page 008          Editeur, boucle principale <xbcl.as>
AREUH ASS. V2.4

0351 00218          *
0352 00218 80D0             P=C     0       P := numéro de la commande
0353 0021C
0354 0021C 1B00000          D0=(5)  =FUNCR0
0355 00223 146              C=DAT0  A
0356 00226 D7               D=C     A
0357 00228 164              D0=D0+  5
0358 0022B 146              C=DAT0  A
0359 0022E 135              D1=C    A
0360 00231 164              D0=D0+  5
0361 00234 146              C=DAT0  A
0362 00237 108              R0=C    A
0363 0023A 164              D0=D0+  5
0364 0023D 146              C=DAT0  A
0365 00240 109              R1=C    A
0366 00243
0367 00243          *
0368 00243          * Exeéution des commandes
0369 00243          *
0370 00243 8F0000           GOSBVL  =TBLJMP switch (P)
0371 0024A 640              REL(3)  cmCOPY  C
0372 0024D F40              REL(3)  cmDEL   D
0373 00250 000              REL(3)  =cmEXIT E
0374 00253 550              REL(3)  cmHELP  H
0375 00256 000              REL(3)  =cmINS  I
0376 00259 940              REL(3)  cmJOIN  J
0377 0025C 000              REL(3)  =cmLIST L
0378 0025F 730              REL(3)  cmMOVE  M
0379 00262 000              REL(3)  =cmPRNT P
0380 00265 000              REL(3)  =cmEXIT Q
0381 00268 C10              REL(3)  cmREPL  R
0382 0026B 000              REL(3)  =cmSRCH S
0383 0026E 000              REL(3)  =cmTEXT T
0384 00271 910              REL(3)  cmXCHG  X
0385 00274          *
0386 00274          * On ne sait jamais. L'information vient ici d'un message,
0387 00274          * qui peut être traduit. C'est donc potentiellement un
0388 00274          * risque. Il vaut mieux assurer un comportement stable
0389 00274          * plutôt que de risquer un "Memory Lost".
0390 00274          *
0391 00274 600              REL(3)  invcmd
0392 00277 300              REL(3)  invcmd
0393 0027A
0394 0027A 330000    invcmd LC(4)   (=id)^(=teICMD) Invalid Command
0395 00280 6D20             GOTO    =xederr
0396 00284          *
0397 00284          * Quelques rallonges...
0398 00284          *
0399 00284 8C0000    cmREPL GOLONG  =cmREPL
0400 0028A 8C0000    cmXCHG GOLONG  =cmXCHG
```

```
Page 009          Editeur, boucle principale <xbcl.as>
AREUH ASS. V2.4

0401 00290 8C0000    cmCOPY GOLONG  =cmCOPY
0402 00296 8C0000    cmMOVE GOLONG  =cmMOVE
0403 0029C 8C0000    cmDEL  GOLONG  =cmDEL
0404 002A2 8C0000    cmJOIN GOLONG  =cmJOIN
0405 002A8 8C0000    cmHELP GOLONG  =cmHELP
0406 002AE
0407 002AE           ********************************************************************
0408 002AE           * xederr, dspmsg
0409 002AE           *
0410 002AE           * But: afficher une erreur et repartir à la boucle (erreur)
0411 002AE           *   ou revenir à l'appelant (dspmsg)
0412 002AE           * Entrée:
0413 002AE           *   - C(3-0) = numéro d'erreur
0414 002AE           * Sortie:
0415 002AE           *   - par BOUCLE (erreur)
0416 002AE           *   - par RTN (dspmsg)
0417 002AE           * Abime: A-D, D0, D1, R0
0418 002AE           * Appelle: MFWRN
0419 002AE           * Niveaux: 2 (dspmsg) ou 3 (erreur)
0420 002AE           * Note: Un traitement spécial est effectué pour eMEM car
0421 002AE           *   MFWRN ne revient pas si C(3-0) = eMEM. Nous remplacons
0422 002AE           *   donc eMEM par eXMEM qui est notre "Insufficient Mem" à
0423 002AE           *   nous mêmes que d'abord que.
0424 002AE           * Historique:
0425 002AE           *   88/05/14: PD/JT conception & codage
0426 002AE           *   88/07/10: PD/JT séparation de dspmsg
0427 002AE           *   88/11/13: PD/JT traitement de eMEM
0428 002AE           ********************************************************************
0429 002AE
0430 002AE 760A      =xederr GOSUB   mem?
0431 002B2 DA                P=      %10#
0432 002B4           *
0433 002B4           * 1     BEEF
0434 002B4           *  0    store ERRN
0435 002B4           *   1   display message only (no "WRN Lxxx:")
0436 002B4           *    0  DELAY respecté
0437 002B4           *
0438 002B4 8F0####           GOSBVL  =MFWRN
0439 002B8 D2                C=0     A
0440 002BD 1F0####           D1=(5)  =LCMD
0441 002C4 145               DAT1=C  A       LCMD := W
0442 002C7 FFFE              GOTO    =BOUCLE
0443 002CB
0444 002CB 7300      =dspmsg GOSUB   mem?
0445 002CF D2                P=      %A111
0446 002D1           *
0447 002D1           * W     no BEEP
0448 002D1           *  1    do not store ERRN
0449 002D1           *   1   display message only (no "WRN Lxxx:")
0450 002D1           *    1  no DELAY
```

```
Page 011          Editeur, boucle principale <xbcl.as>
AREUH ASS. V2.4     •••• SYMBOL TABLE ••••

=AVMEME       Extrn Ukn         -  0148
=AVS2DS       Extrn Ukn         -  0260
=BOUCLE       001BD Rel  0297   -  0213 0442
=BSERR        Extrn Ukn         -  0182
=BUFADR       Unkwn Ukn  0015   -
=BUFID        Unkwn Ukn  0014   -
 BCL000       001C0 Rel  0298   -  0322
 BCL100       001D2 Rel  0320   -  0310
 BCL200       001DF Rel  0324   -  0303
=CHKTAT       Extrn Ukn         -  0173
=CHKWRT       Extrn Ukn         -  0170
=COUR         Unkwn Ukn  0018   -  0195 0239 0249 0264
=CRLFND       Extrn Ukn         -  0268
=DERN         Unkwn Ukn  0019   -  0237
=EOLXCK       Extrn Ukn         -  0099
=EXPEXC       Extrn Ukn         -  0101
=FCNTMP       Unkwn Ukn  0009   -
=FILADR       Unkwn Ukn  0017   -  0174
=FILE1        Extrn Ukn         -  0139
=FILE2        Extrn Ukn         -  0166
=FLIMSK       00002 Abs  0053   -  0204 0208
=FLAG1        Unkwn Ukn  0022   -  0210
=FLGREG       Extrn Ukn         -  0201
=FUNCR0       Extrn Ukn         -  0331 0354
 FiLeNd       002F6 Rel  0485   -
=IVAERR       Extrn Ukn         -  0079
=KYDN?        Extrn Ukn         -  0277
=LCMD         Unkwn Ukn  0021   -  0158 0440
=LDCSPC       Extrn Ukn         -  0045
=MFWRN        Extrn Ukn         -  0438 0452
=MODE71       Unkwn Ukn  0007   -  0133
=NCMDED       0000D Abs  0054   -
=Num2D1       Extrn Ukn         -  0252
=OBCOLL       Extrn Ukn         -  0246
=PCMD         Unkwn Ukn  0020   -  0153
=POP1S        Extrn Ukn         -  0146
=QUERY        Unkwn Ukn  0008   -
=RDONLY       Unkwn Ukn  0023   -  0171
=RNDAHX       Extrn Ukn         -  0071
=SCRTCH       Extrn Ukn         -  0039 0047
=STSAVE       Extrn Ukn         -  0026
=TBLJMP       Extrn Ukn         -  0370
=TEDITd       Extrn Ukn         -  0118
=TEDITe       00041 Rel  0120   -
=TEDITp       Extrn Ukn         -  0119
=TMP1         Unkwn Ukn  0047   -
=TMP3         Unkwn Ukn  0046   -
=TMP5         Unkwn Ukn  0045   -
=TRFMEF       Extrn Ukn         -  0007 0008 0003 0014 0015 0017 0018 0019 0020
                                 •  0021 0022 0023 0029 0126
```

```
Page 010          Editeur, boucle principale <xbcl.as>
AREUH ASS. V2.4

0451 00CD1           •
0452 00CD1 8D#####           GOLONG  =MFWRN
0453 00CD5
0454 00CD5
0455 00CD5           ********************************************************************
0456 00CD5           * mem?
0457 00CD5           *
0458 00CD5           * But: MFWRN ne revenant pas s'il y a une erreur eMEM, il
0459 00CD5           *   faut tester spécialement ce cas et remplacer eMEM par
0460 00CD5           *   notre message d'erreur.
0461 00CD5           * Entrée:
0462 00CD5           *   - C(4-0) = message d'erreur
0463 00CD5           * Sortie:
0464 00CD8           *   - C(4-0) = message d'erreur modifié si eMEM
0465 00CD8           *   - P = 0
0466 00CD8           * Abime: A(A), C(A), P
0467 00CD8           * Appelle: -
0468 00CD8           * Niveaux: 0
0469 00CD8           * Historique:
0470 00CD8           *   88/11/13: PD/JT conception & codage
0471 00CD8           ********************************************************************
0472 00CD8
0473 00CD8 D0        mem?    A=0     A
0474 00CDA 23                P=      3
0475 00CDC A9A               A=C     WP
0476 00CDF 20                P=      0
0477 00CE1 D2                C=0     A
0478 00CE3 310#              LC(2)   =eMEM
0479 00CE7 DE                ACEX    A
0480 00CE9 8A6               ?A#C    A
0481 00CEC 00                RTNYES
0482 00CEE 350###            LC(4)   (=id)(=eXMEM,  notre erreur à nous
0483 00CF4 01                RTN
0484 00CF5
0485 00CFF                   END
```

```
Page 013          Editeur, boucle principale <xbcl.as>
AREUH ASS. V2.4   **** SYMBOL TABLE ****

=teEOF        Extrn Ukn       -  0271
=teICMD       Extrn Ukn       -  0394
 viewln       0013G Rel  0233 -  0315
=xeddec       Extrn Ukn       -  0302
=xederr       002AE Rel  0430 -  0395



Source  : xbcl.as

Object  : obj/xbcl.ao

Listing : list/xbcl.al

Date    : Sat Aug 12 17:41:01 1989

Errors  : 000


Areuh Assembler/Linker V2.4. (c) P. David & J. Taillandier 1986 Paris, France
```

```
Page 001          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0001 00000                    TITLE   Editeur, chaines génériques <xgen.as>
0002 00000
0003 00000          *
0004 00000          * Ce module n'a pas l'air de consommer de mémoire (sauf pour
0005 00000          * stocker le buffer bien sur).
0006 00000          * En revanche, la consommation de registres est
0007 00000          * impressionnante.
0008 00000          *
0009 00000
0010 00000          *
0011 00000          * Définitions pour les tokens compilés
0012 00000          *
0013 00000           CHAR   EQU     0       a character (valeur cablee)
0014 00000           EOL    EQU     1       end of line
0015 00000           ANY    EQU     2       any character
0016 00000           CCL    EQU     3       character class
0017 00000           NCCL   EQU     4       not in character class
0018 00000           END    EQU     5       end of compiled stream
0019 00000           BOL    EQU     6       begin of line
0020 00000           CLOS   EQU     7       closure
0021 00000          *
0022 00000          * et maintenant leur taille
0023 00000          *
0024 00000           BOLS   EQU     1       BOL
0025 00000           EOLS   EQU     1       EOL
0026 00000           ANYS   EQU     1       ANY
0027 00000           CCLS   EQU     1+256   CCL c0 c1 ... c255
0028 00000           NCCLS  EQU     1+256   NCCL c0 c1 ... c255
0029 00000           CHARS  EQU     1+2     CHAR c
0030 00000           CLOSS  EQU     1       CLOS `token`
0031 00000
0032 00000          *
0033 00000          * Définitions pour les caractères HP-UX
0034 00000          *
0035 00000           xBOL   EQU     '^'     begin of line
0036 00000           xEOL   EQU     '$'     end of line
0037 00000           xANY   EQU     '.'     any character
0038 00000           xBCCL  EQU     '['     begin of character class
0039 00000           xECCL  EQU     ']'     end of character class
0040 00000           xNCCL  EQU     '^'     not in character class
0041 00000           xICCL  EQU     '-'     interval in character class
0042 00000           xCLOS  EQU     '*'     closure
0043 00000           xREP   EQU     '%'     remplacement
0044 00000           xESC   EQU     '\'     escape
0045 00000          *
0046 00000          * Définitions pour les caractères HP-71
0047 00000          *
0048 00000           8BOL   EQU     '^'     begin of line
0049 00000           8EOL   EQU     '$'     end of line
0050 00000           8ANY   EQU     '.'     any character
```

```
Page 002          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0051 00000           8REPT  EQU     '@'     repetition (.*)
0052 00000           8REF   EQU     '%'     remplacement
0053 00000           8TOGL  EQU     '~'     toggle
0054 00000
0055 00000          *
0056 00000          * Notes sur les closures et leur utilisation :
0057 00000          * - Les closures sont codées sous forme d'un tableau de 256
0058 00000          *   quartets (on pourrait le compresser à 256 bits).
0059 00000          * - il y a donc un nombre maximum de 14 closures.
0060 00000          * - pour les utiliser, il faut une pile, contenant à chaque
0061 00000          *   fois 20 quartets.
0062 00000          * - on place cette pile en début de buffer, ce qui laisse
0063 00000          *   toujours 14 closures.
0064 00000          *
0065 00000           STKSIZ EQU     14*20   14 closures * 20 quartets
0066 00000
0067 00000          ********************************************************
0068 00000          * COMPUX
0069 00000          *
0070 00000          * But: compiler une chaine générique
0071 00000          * Entree:
0072 00000          *   - D1 = ` M.S. sur la chaine à compiler
0073 00000          *   - A(A) = longueur en octets
0074 00000          * Sortie:
0075 00000          *   - Cy = 1 : erreur
0076 00000          *       C(4-0) = numéro d'erreur pour BSERR
0077 00000          *   - Cy = 0 : pas d'erreur
0078 00000          *       R0(7-3) = D1 = ` début du buffer (sur les donnees)
0079 00000          *       R0(2-0) = B(X) = buffer ID
0080 00000          * Abime: A-D, D0, D1, R0-R2
0081 00000          * Appelle: gettbuf, addchr, marque, lookah, getchr,
0082 00000          *   FINDA, I/OCON, WIPOUT, gettab, MOVED2
0083 00000          * Niveaux: 4 (gettbuf)
0084 00000          * Detail:
0085 00000          *   créer le buffer
0086 00000          *   tant que non fin de chaine
0087 00000          *     faire
0088 00000          *       mémoriser le début du pattern précédent
0089 00000          *       selon caractère lu
0090 00000          *         '\' : ajouter_normal (caractere suivant)
0091 00000          *         '.' : ajouter (ANY)
0092 00000          *         '^' : si début de ligne
0093 00000          *                 alors: ajouter (BOL)
0094 00000          *                 sinon ajouter_normal ('^')
0095 00000          *               fin si
0096 00000          *         '$' : si fin de ligne
0097 00000          *                 alors: ajouter (EOL)
0098 00000          *                 sinon ajouter_normal ('$')
0099 00000          *               fin si
0100 00000          *         '[' : ajouter (CCL ou NCCL selon caractere suivant
```

```
Page 003          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0101 00000          *                 analyser ce qui suit et créer tableau
0102 00000          *         '*' : décaler le pattern précédent
0103 00000          *                 insérer la closure avant
0104 00000          *               sinon : ajouter_normal (caractère lu)
0105 00000          *       fin selon
0106 00000          *   fin tant que
0107 00000          * Historique:
0108 00000          *   88/06/05: PD/JT conception & codage
0109 00000          *   88/10/29: PD/JT suppression du buffer en cas d'erreur
0110 00000          ********************************************************
0111 00000
0112 00000 7B24     =COMPUX GOSUB   getbuf
0113 00004 400              RTNC            si erreur
0114 00007          *
0115 00007          * B(A) = taille du buffer
0116 00007          * D(A) = taille de la chaine
0117 00007          * D1 = ` chaine
0118 00007          * D0 = ` buffer
0119 00007          * R0(2-0) = bufid
0120 00007          * R0(7-3) = ` buffer
0121 00007          *
0122 00007 D2               C=0     A
0123 00009 109              R1=C            marque := 0
0124 0000C          *
0125 0000C          * D(A) = nb de caractères restant dans la chaine
0126 0000C          * B(A) = taille du buffer
0127 0000C          * D1 = ` chaine
0128 0000C          * D0 = ` buffer
0129 0000C          * R1 = ` token précédent
0130 0000C          *
0131 0000C 8E0000   cux-10 GOSUBL  =getchr
0132 00010 404              GOC     cmpend  Fin de chaine
0133 00015 8F00000          GOSBVL  =FINDA
0134 00019 C5               CON(2)  xESC
0135 0001E 750              REL(3)  cuESC
0136 00021 E2               CON(2)  xANY
0137 00023 E60              REL(3)  cuANY
0138 00026 E5               CON(2)  xBOL
0139 00028 680              REL(3)  cuBOL
0140 0002B 42               CON(2)  xEOL
0141 0002D CB0              REL(3)  cuEOL
0142 00030 B5               CON(2)  xBCCL
0143 00032 3F0              REL(3)  cuCCL
0144 00035 A2               CON(2)  xCLOS
0145 00037 A02              REL(3)  cuCLOS
0146 0003A 00               NIBHEX  00
0147 0003B
0148 0003C 7CA4             GOSUB   marque  marque lasttoken
0149 00040 73B4             GOSUB   addchr
0150 00044 400              RTNC            if error
```

```
Page 004          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0151 00047 64CF             GOTO    cux-10
0152 0004B
0153 0004B          ********************************************************
0154 0004B          * Sortie de la compilation
0155 0004B          ********************************************************
0156 0004B 6BD3     System GOTO     system
0157 0004E 64C4     noroom GOTO     noroom
0158 00053
0159 00053          cmpend
0160 00053          *
0161 00053          * Marqueur de fin
0162 00053          *
0163 00053 A3D              R=R-1   X
0164 00056 48F              GOC     noroom
0165 00059 305              LC(1)   END
0166 0005C 1540             DAT0=C  P
0167 00060          *
0168 00060          * Réduire la taille du buffer
0169 00060          *
0170 00060 118              C=R0            C(X) = buffer id
0171 00063 3F00000          GOSBVL  =I/OCON
0172 00067 50E              GONC    System  buffer not found !
0173 0006D          *
0174 0006D          * Maintenant, il faut renvoyer les parametres du buffer
0175 0006D          *
0176 0006D 118              C=R0
0177 00070 AB5              B=C     X       B(A) := buffer id
0178 00073          * Cy = 0 : pas d'erreur
0179 00073          * D1 = ` début du buffer (sur les donnees) (après I/OCON)
0180 00073          * B(X) = buffer ID
0181 00073 03               RTNCC
0182 00075
0183 00075 8E0000   cuESC  GOSUBL  =getchr
0184 00079 411              GOC     InvPat  "\" et fin de ligne
0185 0007E 7A64             GOSUB   marque
0186 00082 7174             GOSUB   addchr
0187 00086 400              RTNC            if error
0188 00089 6Q8F             GOTO    cux-10  et retour à la boucle
0189 0008D
0190 0008D 6112     InvPat GOTO     invpat
0191 00091
0192 00091 7754     cuANY  GOSUB    marque
0193 00095 32100            LC(3)   ANYS
0194 0009A B31              B=B-C   X
0195 0009D 474              GOL     Noroom
0196 000A0 302              LC(1)   ANY
0197 000A3 1540             DAT0=C  P
0198 000A7 1B0              D0=D0+  1
0199 000AA 616F             GOTO    cux-10  et retour à la boucle
0200 000AE
```

```
Page 005          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0201 000AE 111       cuBOL  A=R1
0202 000B1 7734             GOSUB   marque
0203 000B5 8A8              ?A=0    A       Début de ligne ?
0204 000B8 41               GOYES   cuBL10  oui
0205 000BA 31E5             LCASC   '*'
0206 000BE AEA              A=C     B
0207 000C1 7234             GOSUB   addchr   '*' normal
0208 000C5 400              RTNC            if error
0209 000C8 634F             GOTO    cux-10  et retour à la boucle
0210 000CC 32100     cuBL10 LC(3)   BOLS    '*' générique
0211 000D1 B31              B=B-C   X
0212 000D4 401              GOC     Noroom
0213 000D7 30b              LC(1)   BOL
0214 000DA 1540             DAT0=C  P
0215 000DE 160              D0=D0+  1
0216 000E1 6A2F             GOTO    cux-10  et retour à la boucle
0217 000E5
0218 000E5 6E24      Noroom GOTO    noroom
0219 000E9
0220 000E9 7FF3      cuEOL  GOSUB   marque
0221 000ED 8E0000           GOSUBL  =lookah
0222 000F3 441              GOC     cuEL10  fin de ligne
0223 000F6 3142             LCASC   '$'
0224 000FA AEA              A=C     B
0225 000FD 76F3             GOSUB   addchr  '$' normal
0226 00101 400              RTNC            if error
0227 00104 670F             GOTO    cux-10  et retour à la boucle
0228 00108 32100     cuEL10 LC(3)   EOLS    '$' générique
0229 0010D B31              B=B-C   X
0230 00110 44D              GOC     Noroom
0231 00113 301              LC(1)   EOL
0232 00116 1540             DAT0=C  P
0233 0011A 160              D0=D0+  1
0234 0011D 6EEE             GOTO    cux-10  et retour à la boucle
0235 00121
0236 00121 6D71      Invpat GOTO    invpat
0237 00125
0238 00125 7BC3      cuFCL  GOSUB   marque
0239 00129 32101            LC(3)   CCLS
0240 0012E B31              B=B-C   X
0241 00131 438              GOC     Noroom
0242 00134 8E0000           GOSUBL  =lookah
0243 0013A 46E              GOC     Invpat  "[" et fin de ligne
0244 0013D 303              LC(1)   CCL
0245 00140 816              CSRC            C(S) := CCL
0246 00143 31E5             LCASC   ']'
0247 00147 966              ?A#B    B
0248 0014A E0               GOYES   cuCL10
0249 0014C 304              LC(1)   NCCL
0250 0014F 816              CSRC
```

```
Page 007          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0301 001E8 8E0000    cuCL80 GOSUBL  =getchr passer le '-'
0302 001EE 8E0000           GOSUBL  =getchr
0303 001F4 4FE              GOC     invPat  "[...-" et fin de ligne
0304 001F7 31C5             LC(2)   XESC
0305 001FB 966              ?A#C    B
0306 001FE B0               GOYES   cuCL82
0307 00200 8E0000           GOSUBL  =getchr
0308 00206 4D0              GOC     invPat  "[...-\" et fin de ligne
0309 00209 AE6       cuCL82 C=A     B       C(B) := upper
0310 0020C F4               ASR     A
0311 0020E F4               ASR     A       A(B) := lower
0312 00210 B62              C=C-A   B       range
0313 00213 10A              R2=C            R2 := range
0314 00216 4DC              GOC     invPat  lower > upper
0315 00219 7303             GOSUB   gettab  D0 := ' tab [lower]
0316 0021D 12A              CR2EX           R2 := ' tab [0] ; C(B) := range
0317 00220 AEA              A=C     B       compteur dans A(B)
0318 00223 301              LC(1)   1
0319 00226 1540      cuCL84 DAT0=C  P
0320 0022A 160              D0=D0+  1
0321 0022D A8C              A=A-1   B
0322 00230 55F              GONC    cuCL84
0323 00233 11A              C=R2            C(A) := ' tab [0]
0324 00236 134              D0=C
0325 00239 654F             GOTO    cuCL50
0326 0023D
0327 0023D 66D2      noRoom GOTO    noroom
0328 00241
0329 00241 A3D       cuCLOS B=B-1   X
0330 00244 48F              GOC     noRoom
0331 00247 111              A=R1            last token
0332 0024A 8A8              ?A=0    A
0333 0024D 79               GOYES   invPat  "•"
0334 00250 132              AD0EX           A(A) := courant; D0 := ' last
0335 00252 102              R2=A            R2 := courant
0336 00255 1520             A=DAT0  P       A(0) := last token
0337 00259 306              LC(1)   BOL
0338 0025C 90C              ?A=C    P
0339 0025F 04               GOYES   invpat  "^•"
0340 00261 307              LC(1)   CLOS
0341 00264 90C              ?A=C    P
0342 00267 83               GOYES   invpat  "...••"
0343 00269
0344 00269 133              AD1EX           A(A) := D1
0345 0026C 122              AR2EX           A(A) := end of dest; R2 := D1
0346 0026F 131              D1=A
0347 00272 101              R1=A            on remettra last token après
0348 00275 170              D1=D1+  1       D1 := ' end of dest
0349 00278 136              CD0EX
0350 0027B           •
```

```
Page 006          Editeur, chaines génériques: <xgen.as>
AREUH ASS. V2.4

0251 00152 8E0000           GOSUBL  =getchr
0252 00158 1544      cuCL10 DAT0=C  S
0253 0015C 1E0              D0=D0+  1
0254 0015F           •
0255 0015F           • tab [0..255] := 0
0256 0015F           •
0257 0015F 136              CD0EX
0258 00162 134              D0=C
0259 00165 137              CD1EX           D1 := D0
0260 00168 10A              R2=C            R2 := D1
0261 0016B 3A0100           LC(5)   (CCLS)-1
0262 00170 8F0000           GOSBVL  =WIPOUT
0263 00174 11A              C=R2
0264 00178 135              D1=C
0265 0017C
0266 0017C 8E0000    cuCL50 GOSUBL  =getchr
0267 00182 4B8              GOC     Invpat  "]" manquant
0268 00185 31D5             LC(2)   XECCL   ']'
0269 00189 962              ?A=C    B
0270 0018C 24               GOYES   cuCL90
0271 0018E 31C5             LC(2)   XESC    '\'
0272 00192 966              ?A#C    B
0273 00195 B0               GOYES   cuCL80
0274 00198 8E0000           GOSUBL  =getchr
0275 0019E 408              GOC     Invpat  "[...\" et fin de ligne
0276 001A1 F0        cuCL60 ASL     A
0277 001A3 F0               ASL     A       A(0-2) = caractère
0278 001A5 8E0000           GOSUBL  =lookah
0279 001AB 4B0              GOC     cuCL70  fin de ligne (anormal)
0280 001AE 31DC             LC(2)   XICCL   '-'
0281 001B2 962              ?A=C    B
0282 001B5 17               GOYES   cuCL80
0283 001B8           • caractère simple
0284 001B8 F4        cuCL70 ASR     A
0285 001BB F4               ASR     A       A(B) := caractère
0286 001BD 7F53             GOSUB   gettab  D0 := ' caractère dans le tableau
0287 001C1 DA               A=C     A       A(A) := ancienne adresse
0288 001C3 301              LC(1)   1
0289 001C6 1540             DAT0=C  P       tab [caractère] := true
0290 001CA 130              D0=A            D0 := ' tab [0]
0291 001CD 61BF             GOTO    cuCL50  et un tour de boucle de plus !
0292 001D1
0293 001D1 132       cuCL90 AD0EX
0294 001D4 3400100          LC(5)   (CCLS)-1
0295 001DB CA               A=A+C   A
0296 001DD 130              D0=A
0297 001E0 6B2E             GOTO    cux-10  et retour à la boucle
0298 001E4
0299 001E4 6AB0      invPat GOTO    invpat
0300 001E8           • ...--
```

```
Page 008          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0351 0027B           • C(A) = start of source
0352 0027B           • A(A) = end of source
0353 0027B           • D1 = end of destination
0354 0027B           • R1 = end of source
0355 0027B           • R2 = ancien D1
0356 0027B           •
0357 0027B 8F0000           GOSBVL  =MOVEDD
0358 0027F
0359 0027F 112              A=R2            A(A) := ancien D1
0360 00283 131              D1=A            D1 est restauré
0361 00286 307              LC(1)   CLOS
0362 0028B 1540             DAT0=C  P
0363 0028F
0364 0028F 132              AD0EX           A(A) := ' last token
0365 00293 121              AR1EX
0366 00297           • R1 := ' last token ; A(A) := end of source
0367 00297 130              D0=A
0368 0029B 160              D0=D0+  1
0369 0029F
0370 0029F 6A7D             GOTO    cux-10
0371 002A3
0372 002A3 7776      invpat GOSUB   =ENDPOS
0373 002A8 330000           LC(4)   (=id)^(=teIPAT)
0374 002AD 02               RTNSC
0375 002AF
0376 002AF           ********************************************************
0377 002AF           • COMP71
0378 002AF           •
0379 002AF           • But: compiler une chaine générique
0380 002AF           • Entrees:
0381 002AF           •   - D1 = ' M.S. sur la chaine à compiler
0382 002AF           •   - A(A) = longueur en octets
0383 002AF           • Sorties:
0384 002AF           •   - Cy = 1 : erreur
0385 002AF           •        C(4-0) = numero d'erreur pour BSERR
0386 002AF           •   - Cy = 0 : pas d'erreur
0387 002AF           •        R0(7-3) = D1 = ' début du buffer (sur les données)
0388 002AF           •        R0(2-0) = B(X) = buffer ID
0389 002AF           • Abime: A-D, D0, D1, R0-R2
0390 002AF           • Appelle: getbuf, addchr, lookah, getchr, FINDA, I/OCON
0391 002AF           • Niveaux: 4 (getbuf)
0392 002AF           • Detail:
0393 002AF           •   créer le buffer
0394 002AF           •   état := 0
0395 002AF           •   tant que non fin de chaine
0396 002AF           •     faire
0397 002AF           •       lire le caractère
0398 002AF           •       si caractère lu = '\'
0399 002AF           •         alors
0400 002AF           •           selon état
```

```
Page 009          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

002AB               *                  0 : état = 1
002AB               *                  1 : ajouter_normal ('\')
002AB               *                      état = 0
002AB               *                  2 : état = 3
002AB               *                  3 : ajouter_normal ('\')
002AB               *                      état = 2
002AB               *                fin selon
002AB               *              sinon
002AB               *                selon état
002AB               *                  0 : ajouter_normal (caractère lu)
002AB               *                  1 : état = 2
002AB               *                      traiter_spécial (caractère lu)
002AB               *                  2 : ajouter_normal (caractère lu)
002AB               *                  3 : état = 0
002AB               *                      ajouter__normal (caractère lu)
002AB               *          fin si
002AB               *    fin tant que
002AB               *
002AB               * traiter_spécial (caractère)
002AB               *    selon caractère lu
002AB               *       '.' : ajouter (ANY)
002AB               *       '^' : si début de ligne
002AB               *               alors ajouter (BOL)
002AB               *               sinon ajouter_normal ('^')
002AB               *             fin si
002AB               *       '$' : si fin de ligne
002AB               *               alors ajouter (EOL)
002AB               *               sinon ajouter_normal ('$')
002AB               *             fin si
002AB               *       '@' : ajouter (CLOSURE)
002AB               *             ajouter (ANY)
002AB               *    fin selon
002AB               * Historique:
002AB               *    88/06/26: PD/JT conception & codage d'après COMPUX
002AB               *    88/10/29: PD/JT suppression du buffer en cas d'erreur
002AB               **************************************************************
002AB
002AB 7051          =COMP71 GOSUB   getbuf
002AF 400                   RTNC            si erreur
002B2               *
002B2               * B(A) = taille du buffer
002B2               * D(A) = taille de la chaine
002B2               * D1 = ^ chaine
002B2               * D0 = ^ buffer
002B2               * R0(2-0) = bufid
002B2               * R0(7-3) = ^ buffer
002B2               *
002B2 300                   LC(1)   0
002B5 109                   R1=C            état := 0
002B8               *
```

```
Page 010          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

002B8               * D(A) = nb de caractères restant dans la chaine
002B8               * B(A) = taille du buffer
002B8               * D1 = ^ chaine
002B8               * D0 = ^ buffer
002B8               * R1(0) = ^ état de l'automate cablé
002B8               *
002B8 8E0000  c71-10 GOSUBL =getchr
002BE 411            GOC     cmpend  Fin de chaine
002C1 31C5           LC(2)   =TOGL
002C5 966            ?A#C    B       A(B) = caractère lu
002C8 C1             GOYES   c71nor  caractère normal (ou générique)
002CA 119            C=R1
002CD 8F00000        GOSBVL  =TBLJMC
002D4 620            REL(3)  c7b0    back-slash
002D7 D20            REL(3)  c7b1
002DA 130            REL(3)  c7b2
002DD 830            REL(3)  c7b3
002E0 6270    cmpend GOTO    cmpend  rallonge
002E4 119     c71nor C=R1
002E7 8F00000        GOSBVL  =TBLJMC
002EE D20            REL(3)  c7n0    normal
002F1 530            REL(3)  c7n1
002F4 830            REL(3)  c7n2
002F7 D50            REL(3)  c7n3
002FA
002FA 301     c7b0   LC(1)   1
002FD 109            R1=C            état := 1
00300 67BF           GOTO    c71-10
00304 300     c7b1   LC(1)   0
00307 6010           GOTO    c7b30
0030B 303     c7b2   LC(1)   3
0030E 109            R1=C            état := 3
00311 66AF           GOTO    c71-10
00315 302     c7b3   LC(1)   2
00318 109     c7b30  R1=C            état := 0
0031B               *
0031B               * Attention ! le code continue !!!
0031B               *
0031B 7?D1    c7n0   GOSUB   addchr  ajouter_normal (caractère lu)
0031F 400            RTNC
00322 659F           GOTO    c71-10
00326 302     c7n1   LC(1)   2
00329 109            R1=C
0032C 8F00000 c7n2   GOSBVL  =FINDA  routine "traite_spécial"
00333 E1             CON(2)  SANY
00335 030            REL(3)  c7ANY
00338 E5             CON(2)  SBOL
0033A 440            REL(3)  c7BOL
0033D 4C             CON(2)  SEOL
0033F 190            REL(3)  c7EOL
```

```
Page 011          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

00342 04             CON(2)  SREPT
00344 0C0            REL(3)  c7REPT
00347 00             NIBHEX  00
00349 7AA1           GOSUB   addchr
0034D 400            RTNC            if error
00350 676F           GOTO    c71-10
00354
00354 7F91    c7n3   GOSUB   addchr  ajouter_normal (caractère lu)
00358 400            RTNC
0035B 300            LC(1)   0
0035E 109            R1=C            état := 0
00361 665F           GOTO    c71-10
00365
00365 32100   c7ANY  LC(3)   ANYS
0036A B31            B=B-C   X
0036D 4E5            GOC     NoRoom
00370 302            LC(1)   ANY
00373 1540           DAT0=C  P
00377 160            D0=D0+  1
0037A 6D3F           GOTO    c71-10
0037E 110     c7BOL  A=R0
00381 BF4            ASR     W
00384 BF4            ASR     W
00387 BF4            ASR     W       A(A) := ^ début buffer
0038A 3481100        LC(5)   STKSIZ
00391 C2             C=C+A   A       C(A) := ^ début motif (après pile)
00393 132            AD0EX           A(A) := ^ position courante
00396 8A2            ?A=C    A
00399 71             GOYES   c7BL10  en début de buffer
0039B 132            AD0EX
0039E 31E5           LCASC   '^'
003A2 AEA            A=C     B
003A5 7E41           GOSUB   addchr  '^' normal
003A9 400            RTNC            si erreur
003AC 6B0F           GOTO    c71-10
003B0 132     c7BL10 AD0EX
003B3 32100          LC(3)   BOLS    '^' générique
003B8 B31            B=B-C   X
003BB 401            GOC     NoRoom
003BE 306            LC(1)   BOL
003C1 1540           DAT0=C  P
003C5 160            D0=D0+  1
003C8 6FEE           GOTO    c71-10
003CC 6741    NoRoom GOTO    noroom
003D0 8E0000  c7EOL  GOSUBL  =lookah
003D6 441            GOC     c7EL10  fin de ligne
003D9 3142           LCASC   '$'
003DD AEA            A=C     B
003E0 7311           GOSUB   addchr  '$' normal
003E4 400            RTNC            si erreur
```

```
Page 012          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

003E7 60DE           GOTO    c71-10
003EB 32100   c7EL10 LC(3)   EOLS
003F0 B31            B=B-C   X
003F3 48D            GOC     NoRoom
003F6 301            LC(1)   EOL
003F9 1540           DAT0=C  P
003FD 160            D0=D0+  1
00400 67BE           GOTO    c71-10
00404 32200   c7REPT LC(3)   (CLOSS)+(ANYS)
00409 B31            B=B-C   X
0040C 4FE            GOC     NoRoom
0040F 307            LC(1)   CLOS
00412 1540           DAT0=C  P
00416 160            D0=D0+  1
00419 302            LC(1)   ANY
0041C 1540           DAT0=C  P
00420 160            D0=D0+  1
00423 649E           GOTO    c71-10
00427
00427 330000  system LC(4)   =eSYSER
0042D 02             RTNSC
0042F
0042F               **************************************************************
0042F               * getbuf
0042F               *
0042F               * But: allouer un buffer de taille maximum pour la chaîne à
0042F               *    compiler.
0042F               * Entree:
0042F               *    - AVMEMS et AVMEME positionnés
0042F               * Sortie:
0042F               *    - D1 n'est pas modifié
0042F               *    - D(A) = A(A) en entrée
0042F               *    - D0 = ^ buffer (après la pile)
0042F               *    - R0(2-0) = buffid
0042F               *      R0(7-3) = ^ début du buffer (partie donnees)
0042F               *    - B(A) = taille du buffer
0042F               * Abime: A(15-5), B-D, D0, R0
0042F               * Appelle: IOFSCR, I/OALL
0042F               * Niveaux: 3 (I/OALL)
0042F               * Detail: le buffer est alloué au maximum de la place
0042F               *    mémoire disponible. STKSIZ quartets sont réservés pour
0042F               *    stocker la pile. La taille renvoyée et le pointeur
0042F               *    D0 tiennent compte de la pile. En revanche, le pointeur
0042F               *    dans R0 est le vrai début du buffer
0042F               * Historique:
0042F               *    88/06/05: PD/JT conception & codage
0042F               **************************************************************
0042F
0042F BF0     getbuf ASL     W
00432 BF0            ASL     W
```

```
0601 00435 BF0              ASL     W
0602 00438 BF0              ASL     W
0603 0043B BF0              ASL     W
0604 0043E 133              AD1EX
0605 00441 100              R0=A            R0 := sauvegarde de A et D1
0606 00444                  •
0607 00444                  • Trouver un id scratch disponible
0608 00444                  •
0609 00444 8F00000          GOSBVL  =IOFSCR A, C(A), D1, 1 niveau
0610 0044B 4BD              GOC     system  No available buffer
0611 0044E                  •
0612 0044E                  • C(X) = buffer id
0613 0044E                  •
0614 0044E D5               B=C     A       B(A) := buffer id
0615 00450 1F00000          D1=(5)  =AVMEME
0616 00457 143              A=DAT1  A
0617 0045A 1C0              D1=D1-  (=AVMEME)-(=AVMEMS)
0618 0045D 147              C=DAT1  A
0619 00460 EA               A=A-C   A       A(A) := MEM en quartets
0620 00462 D2               C=0     A
0621 00464 3100             LC(2)   (=LEEWAY)*7
0622 00468 EA               A=A-C   A       A(A) := place restante après LEEWAY
0623 0046A 497              GOC     errmem
0624 0046D 34FFF00          LC(5)   #FFF    4095 nibs pour le buffer
0625 00474 8BA              ?A<=C   A
0626 00477 40               GOYES   getb10
0627 00479 DA               A=C     A       C(A) := 4095 nibs
0628 0047B                  • B(A) = id du buffer
0629 0047B                  • A(A) = taille demandée
0630 0047B 3481100  getb10  LC(5)   STKSIZ
0631 00482 882              ?A<C    A
0632 00485 F5               GOYES   errmem
0633 00487 D6               C=A     A       C(A) := taille demandée
0634 00489 DD               BCEX    A       B(A) = taille demandée, C(A) = bufid
0635 0048B 8F00000          GOSBVL  =I/OALL A-D, D1, D0, 2 niveaux
0636 00492 515              GONC    errmem
0637 00495                  •
0638 00495                  • D1 = ^ past buffer header
0639 00495                  • B(A) = bufsize
0640 00495                  • C(6-0) = header (C(1-3) = buffer id)
0641 00495                  •
0642 00495 133              AD1EX
0643 00498 130              D0=A            D0 := ^ buffer
0644 0049B BF0              ASL     W
0645 0049E BF0              ASL     W
0646 004A1 BF0              ASL     W       A(7-3) = ^ buffer, A(2-0) = 000
0647 004A4 F6               C!R     A       C(X) = buffid
0648 004A6 A3A              A=A+C   X       A(7-3) = ^ buffer, A(2-0) = buffid
0649 004A9 120              AR0EX
0650 004AC 131              D1=A            D1 := ancien D1 restauré
```



```
0651 004AF BF4              ASR     W
0652 004B2 BF4              ASR     W
0653 004B5 BF4              ASR     W
0654 004B8 BF4              ASR     W
0655 004BB BF4              ASR     W       A(A) := ancien D1 restauré
0656 004BE D9               C=B     A
0657 004C0 D7               D=C     A       D(A) := taille du buffer
0658 004C2 D2               C=0     A
0659 004C4 307              LC(1)   7       pour le header
0660 004C7 E3               D=D-C   A
0661 004C9                  •
0662 004C9                  • D1 et A(A) ne sont pas modifiés
0663 004C9                  • D0 = ^ buffer
0664 004C9                  • R0(2-0) = buffid
0665 004C9                  • R0(7-3) = ^ buffer
0666 004C9                  • D(A) = taille du buffer
0667 004C9                  •
0668 004C9 DB               C=D     A       C(A) := taille du buffer
0669 004CB DE               ACEX    A       A(A) := taille ; C(A) := LEN chaine
0670 004CD D7               D=C     A       D(A) := longueur de la chaine
0671 004CF D8               B=A     A       B(A) := taille du buffer
0672 004D1
0673 004D1 3481100          LC(5)   STKSIZ
0674 004D8 E1               B=B-C   A       B(A) := taille (- la pile)
0675 004DA 132              AD0EX
0676 004DD CA               A=A+C   A
0677 004DF 130              D0=A            D0 := ^ après la pile
0678 004E2                  •
0679 004E2                  • B(A) = taille restante dans le buffer
0680 004E2                  • D(A) = taille de la chaine
0681 004E2                  • D1 = ^ chaine
0682 004E2                  • D0 = ^ buffer
0683 004E2                  • R0(2-0) = bufid
0684 004E2                  • R0(7-3) = ^ buffer
0685 004E2                  •
0686 004E2 03               RTNCC
0687 004E4
0688 004E4 330000   errmem LC(4)   =nMEM
0689 004EA 02               RTNSC
0690 004EC
0691 004EC                  ********************************************************
0692 004EC                  • marque
0693 004EC                  •
0694 004EC                  • But: marque la position courante dans la chaine compilée
0695 004EC                  •   comme "last token"
0696 004EC                  • Entree:
0697 004EC                  •   - D0 = ^ buffer
0698 004EC                  • Sortie:
0699 004EC                  •   - R1 = ^ last token
0700 004EC                  • Abime: R1
```



```
0701 004EC                  • Niveaux: 0
0702 004EC                  • Historique:
0703 004EC                  •   88/06/05: PD/JT conception & codage
0704 004EC                  ********************************************************
0705 004EC
0706 004EC 136              marque CD0EX
0707 004EF 109              R1=C
0708 004F2 136              CD0EX
0709 004F5 01               RTN
0710 004F7
0711 004F7                  ********************************************************
0712 004F7                  • addchr
0713 004F7                  •
0714 004F7                  • But: ajoute un caractère au buffer, c'est à dire le torer
0715 004F7                  •   CHAR suivi du caractère.
0716 004F7                  • Entree:
0717 004F7                  •   - A(B) = caractère à écrire
0718 004F7                  •   - B(A) = place restante dans le buffer
0719 004F7                  •   - D0 =   buffer
0720 004F7                  • Sortie:
0721 004F7                  •   - Cy = 1 : erreur
0722 004F7                  •        C(3-0) = error number
0723 004F7                  •   - Cy = 0 : ça s'est bien passé
0724 004F7                  •        D0 réactualisé
0725 004F7                  • Abime: C(A)
0726 004F7                  • Niveaux: 0
0727 004F7                  • Historique:
0728 004F7                  •   88/06/05: PD/JT conception & codage
0729 004F7                  ********************************************************
0730 004F7
0731 004F7 32300    addchr LC(3)   CHARS   size of pattern
0732 004FC B31              B=B-C   X
0733 004FF 441              GOC     noroom
0734 00502 300              LC(1)   CHAR
0735 00505 1540             DAT0=C  P
0736 00509 160              D0=D0+  1
0737 0050C 140              DAT0=A  B
0738 0050F 161              D0=D0+  2       Cy = 0
0739 00512 01               RTN
0740 00514
0741 00514 7204     noroom GOSUB   =ENDPOS
0742 00518 330000           LC(4)   (=id)^(=teICMP) "No Room for Pattern"
0743 0051E 02               RTNSC           Cy = 1 : erreur
0744 00520
0745 00520                  ********************************************************
0746 00520                  • gettab
0747 00520                  •
0748 00520                  • But: obtenir l'adresse du caractère A(B) dans le tableau
0749 00520                  •   pointé par D0.
0750 00520                  • Entree:
```



```
0751 00520                  •   - A(B) = numéro du caractère
0752 00520                  •   - D0 = ^ premier caractère dans le tableau
0753 00520                  • Sortie:
0754 00520                  •   - D0 = ^ tab [caractère]
0755 00520                  •   - C(A) = ancienne adresse (^ tab [0])
0756 00520                  • Abime: C(A), A(4-2)
0757 00520                  • Niveaux: 0
0758 00520                  • Historique:
0759 00520                  •   88/06/05: PD/JT conception & codage
0760 00520                  ********************************************************
0761 00520
0762 00520 D2       gettab C=0      A
0763 00522 AE6              C=A     B
0764 00525 DA               A=C     A
0765 00527 136              CD0EX
0766 0052A 134              D0=C
0767 0052D C2               C=C+A   A
0768 0052F 136              CD0EX
0769 00532 01               RTN
0770 00534
0771 00534                  ********************************************************
0772 00534                  • POSADR
0773 00534                  •
0774 00534                  • But: chercher une chaine générique dans une chaine objet.
0775 00534                  • Entree:
0776 00534                  •   - D0 = ^ chaine à chercher
0777 00534                  •   - A(A) = longueur de la chaine à chercher
0778 00534                  •   - R0(7-0) = caractéristiques du buffer (cf. COMPUX/71)
0779 00534                  •   - R1(A) = ^ vrai début de la chaine à chercher
0780 00534                  • Sortie:
0781 00534                  •   Cy = 1 : match found
0782 00534                  •        - D0 = ^ occurrence dans la chaine objet
0783 00534                  •        - C(A) = longueur de l'occurrence trouvée (en octets)
0784 00534                  •   Cy = 0 : match not found ou erreur
0785 00534                  • Abime: A-D, R0(12-8), R1, R2, D0, D1
0786 00534                  • Appelle: smatch
0787 00534                  • Niveaux: 3
0788 00534                  • Notes:
0789 00534                  •   - La chaine objet doit être dans le "bon" sens, c'est
0790 00534                  •     dire celui des fichiers, et pas celui de la M.S. Cela
0791 00534                  •     oblige à inverser la chaine à chercher si elle est sur
0792 00534                  •     la M.S.
0793 00534                  •   - R1(A) contient l'adresse du "vrai" début de la chaine.
0794 00534                  •     D0 contient l'adresse à partir laquelle POSADR
0795 00534                  •     cherche. Ceci est utile en cas de recherche ne
0796 00534                  •     commençant pas au début (ex: substitution globale,
0797 00534                  •     paramètre de GENPOS, etc.)
0798 00534                  • Remarque: R0(7-0) n'est pas abimé
0799 00534                  • Historique:
0800 00534                  •   88/06/05: PD/JT conception & codage (de quatre lignes)
```

```
Page 017          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0001 00534          •   88/06/26: FD/JT conception & codage (du début du reste)
0002 00534          •   88/07/02: FD/JT débogage et ajout de recherche multiple
0003 00534          •   88/10/07: FD/JT débogage
0004 00534          •   88/10/07: FD/JT retrait du compte dans R1(A) pour amatch
0005 00534          •   88/10/08: FD/JT débogage du cas '^'
0006 00534          •   88/10/09: FD/JT renvoie maintenant la longueur en octets
0007 00534          •   88/10/16: FD/JT séparation de UPDTR0
0008 00534          •   88/10/16: FD/JT suppression du point d'entrée POSID
0009 00534          ************************************************************
0010 00534
0011 00534 118      =POSADR C=R0
0012 00537 8F00000          GOSBVL  =CSRC3
0013 0053E 105              D1=C            D1 := ^ buffer
0014 00541          •
0015 00541          • D1 = ^ buffer (début des données)
0016 00541          • D0 = ^ chaine
0017 00541          • A(A) = longueur de la chaine en octets
0018 00541          •
0019 00541 133              AD1EX           sauvegarde de A dans D1
0020 00544 DA               A=C     A       A(A) := longueur de la chaine
0021 00546 119              C=R1            C(A) := ^ vrai début
0022 00549 8F00000          GOSBVL  =CSRC5  C(15-11) := ^ vrai début
0023 00550 3481100          LC(5)   STASIZ
0024 00557 C2               C=A+C   A       C(A) :=   début du pattern
0025 00559 133              AD1EX           restauration de A(A)
0026 0055C 8F00000          GOSBVL  =CSLC5
0027 00563 109              R1=C            R1(9-5):=pattern: R1(A):=^vrai début
0028 00566
0029 00566 106              CD0EX           C(A) := ^ chaine à examiner
0030 00569 135              D1=C            D1 := ^ chaine
0031 0056C D6               C=A     A
0032 0056E D7               D=C     A       D(A) := nb de caractères
0033 00570
0034 00570          •
0035 00570          • Si le premier caractère est normal, on rentre dans une
0036 00570          • boucle spéciale pour éviter la grosse machinerie de
0037 00570          • "amatch", et en particulier sa récursivité.
0038 00570          •
0039 00570 119              C=R1
0040 00573 8F00000          GOSBVL  =CSRC5  C(A) := pattern
0041 00576 104              D0=C
0042 00578 1520             A=DAT0  P       A(0) := type du premier pattern
0043 0057A 308              ?A=0    P
0044 0057C EA               GOYES   Poschr  boucle spéciale
0045 00580          •
0046 00580          • Si le premier caractère est "^", on ne teste qu'une
0047 00580          • fois sur le début de ligne.
0048 00580          •
0049 00580
0050 00582 306              LC(1)   B0L
```

```
Page 018          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0051 00585 90F              ?A#C    P
0052 00587 A0               GOYES   pos10
0053 0058A 63F0             GOTO    posbol  boucle spéciale pour '^'
0054 0058E
0055 0058E FF70     Poschr GOTO     poschr
0056 00592          •
0057 00592          • Tant pis, il faut y aller. On a signé, c'est..
0058 00592          •
0059 00592
0060 00592          •
0061 00592          • Invariant de boucle :
0062 00592          •   R1(9-5) = pattern; R1(A) =   vrai début
0063 00592          •   D(A) = nb de caractères dans la chaine à chercher
0064 00592          •   D1 = ^ premier caractère à chercher
0065 00592          •
0066 00592 114      pos10  C=R1
0067 00595 816              C#C             C(A) := ^ début pattern
0068 00598 816              C#C
0069 0059B 816              C#C
0070 0059E 816              C#C
0071 005A1 816              C#C
0072 005A4 104              D0=C            D0 := ^ début pattern
0073 005A8
0074 005AB 107              CD1EX
0075 005AE 105              D1=C
0076 005B1 512              CSLC            CSLC5
0077 005B4 812              CSLC
0078 005B7 812              CSLC
0079 005BA 812              CSLC
0080 005BD 812              CSLC
0081 005C0 DB               C=D     A
0082 005C2 10A              R2=C            sauvegarde de D et D1
0083 005C5 7FF0             GOSNB   amatch
0084 005C9 430              GOC     posc20  match found
0085 005CC
0086 005CC 11A              C=R2            restauration de D et D1
0087 005CF D7               D=C     A
0088 005D1 EF6              CSR     W       CSR 5
0089 005D4 BF6              CSR     W
0090 005D7 BF6              CSR     W
0091 005DA BF6              CSR     W
0092 005DD BF6              CSR     W
0093 005E0 135              D1=C
0094 005E3
0095 005E3 171              D1=D1+  2
0096 005E6 CF               D=D-1   A
0097 005E8 5DA              GONC    pos10
0098 005EB 03       RtnCC  RTNCC            no match
0099 005ED
0100 005ED          posc20
```

```
Page 019          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0001 005ED          •
0002 005ED          • D1 = ^ fin de l'occurrence dans la chaine objet
0003 005ED          • R2(9-5) = ^ début de l'occurence
0004 005ED          •
0005 005ED 11A              C=R2
0006 005F0 8F00000          GOSBVL  =CSRC5
0007 005F7          •
0008 005F7          • D1 = ^ fin de l'occurrence dans la chaine objet
0009 005F7          • C(A) = ^ début de l'occurrence
0010 005F7          •
0011 005F7 134      pos30  D0=C             D0 := ^ début de l'occurrence
0012 005FA AD0              A=0     M       pour la division par 2
0013 005FD 133              AD1EX           A(A) := ^ fin occurrence
0014 00600 EA               A=A-C   A       C(A) := 2•LEN(occurrence)
0015 00602 81C              ASRB
0016 00605 D6               C=A     A       C(A) := LEN(occurrence)
0017 00607          •
0018 00607          • D0 = ^ occurrence dans la chaine objet
0019 00607          • C(A) = longueur de l'occurrence trouvée (en octets)
0020 00607          •
0021 00607 02               RTNSC           match found
0022 00609
0023 00609          •
0024 00609          • Boucle spéciale sur le caractère
0025 00609          •
0026 00609 119      poschr C=R1             C(9-5) := ^ début pattern
0027 0060C 816              CSRC            CSRC 5
0028 0060F 816              CSRC
0029 00612 816              CSRC
0030 00615 816              CSRC
0031 00618 816              CSRC
0032 0061B 104              D0=C            D0 := ^ début pattern
0033 0061E
0034 0061E 160              D0=D0+  1
0035 00621 14A              A=DAT0  B       A(B) := premier caractère du motif
0036 00624 161              D0=D0+  2
0037 00627          •
0038 00627          • Boucle très rapide
0039 00627          •   D1 = ^ premier caractère à chercher
0040 00627          •   D(A) = nb de caractères dans la chaine à chercher
0041 00627          •   D0 = ^ patern "après" le premier caractère
0042 00627          •
0043 00627 8AB      posc10 ?D=0     A
0044 0062A 1C               GOYES   RtnCC   no match
0045 0062C 14F              C=DAT1  B
0046 0062F CF               D=D-1   A
0047 00631 171              D1=D1+  2
0048 00634 966              ?A#C    B
0049 00637 0F               GOYES   posc10
0050 00639
```

```
Page 020          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

0051 00639 107              CD1EX
0052 0063C 135              D1=C
0053 0063F BF2              CSL     W       CSL 5
0054 00642 BF2              CSL     W
0055 00645 BF2              CSL     W
0056 00648 BF2              CSL     W
0057 0064B BF2              CSL     W
0058 0064E DB               C=D     A
0059 00650 10A              R2=C            R2 := D1 et D(A)
0060 00653 7860             GOSUB   amatch
0061 00657 4D1              GOC     posc20  match found
0062 0065A
0063 0065A 11A              C=R2            restauration de D1 et D(A)
0064 0065D D7               D=C     A
0065 0065F BF6              CSR     W       CSR 5
0066 00662 BF6              CSR     W
0067 00665 BF6              CSR     W
0068 00668 BF6              CSR     W
0069 0066B BF6              CSR     W
0070 0066E 135              D1=C
0071 00671 679F             GOTO    poschr
0072 00675
0073 00675 11A      posc20 C=R2             C(9-5) := ^ deuxième caractère
0074 00678 8F00000          GOSBVL  =CSRC5
0075 0067F CE               C=C-1   A
0076 00681 CE               C=C-1   A       C(A) := ^ début de l'occurrence
0077 00683 637F             GOTO    pos30
0078 00687
0079 00687          •
0080 00687          • Portion de code spéciale pour ne tester qu'une seule
0081 00687          • fois le "^" de début de ligne
0082 00687          •
0083 00687 119      posbol C=R1             C(A) :=   vrai début
0084 0068A 133              AD1EX
0085 0068D 131              D1=A
0086 00690 8A6              ?A#C    A
0087 00693 62               GOYES   Rtncc
0088 00695
0089 00695 160              D0=D0+  1       passer le BOL
0090 00698
0091 00698 107              CD1EX
0092 0069B 135              D1=C
0093 0069E BF2              CSL     W       CSL 5
0094 006A1 BF2              CSL     W
0095 006A4 BF2              CSL     W
0096 006A7 BF2              CSL     W
0097 006AA BF2              CSL     W
0098 006AD DB               C=D     A
0099 006AF 10A              R2=C
1000 006B2 7900             GOSUB   amatch
```

```
1001 006B6 440            GOC     Pos20
1002 006B9 03      Rtncc  RTNCC           no match
1003 006BB
1004 006BB 613F    Pos20  GOTO    pos20
1005 006BF
1006 006BF          ********************************************************
1007 006BF          * amatch
1008 006BF          *
1009 006BF          * But: reconnaitre une chaine de caractères
1010 006BF          * Entree:
1011 006BF          *   - D0 = ' motif courant
1012 006BF          *   - D1 = ' caractère courant, D(A) = nb de car. restant
1013 006BF          * Sortie:
1014 006BF          *   - Cy = 0 : match not found
1015 006BF          *   - Cy = 1 : match found
1016 006BF          *   - D0 inchangé
1017 006BF          *   - D1 et D(A) actualisés si besoin
1018 006BF          * Abime: A-D, R0(12-8), D0, D1
1019 006BF          * Appelle: omatch, nxtpat
1020 006BF          * Niveaux: 2
1021 006BF          * Algorithme:
1022 006BF          *   tant que pas arrivé à la fin du pattern
1023 006BF          *     si closure
1024 006BF          *       alors
1025 006BF          *         chercher la dernière possibilité (avec omatch)
1026 006BF          *         revenir en arrière jusqu'à ce que
1027 006BF          *               "amatch (reste) = TRUE"
1028 006BF          *         si trouvé alors
1029 006BF          *           alors return TRUE
1030 006BF          *           sinon return FALSE
1031 006BF          *         fin si
1032 006BF          *       sinon si omatch = FALSE alors return FALSE
1033 006BF          *     fin si
1034 006BF          *   fin tant que
1035 006BF          * Historique:
1036 006BF          *   88/06/26: PD/JT conception & codage
1037 006BF          *   88/07/02: PD/JT ajout des closures
1038 006BF          *   88/10/07: PD/JT retrait du compte dans R1(A) pour '*'
1039 006BF          *   88/10/14: PD/JT bug GENLEN("AB","A.*B") corrigée
1040 006BF          ********************************************************
1041 006BF
1042 006BF 118     amatch C=R0             Initialisation de "' courant"
1043 006C2 816             CSRC            CSRC3
1044 006C5 816             CSRC
1045 006C8 816             CSRC
1046 006CB DA              A=C     A       A(A) := ' début de la pile
1047 006CD 816             CSRC            CSRC5
1048 006D0 816             CSRC
1049 006D3 816             CSRC
1050 006D6 816             CSRC
```



```
1051 006D9 816             CSRC
1052 006DC D6              C=A     A
1053 006DE 812             CSLC            CSRC3
1054 006E1 812             CSLC
1055 006E4 812             CSLC
1056 006E7 812             CSLC
1057 006EA 812             CSLC
1058 006ED 812             CSLC
1059 006F0 812             CSLC
1060 006F3 812             CSLC
1061 006F6 108             R0=C            R0(12-8) := ' courant
1062 006F9
1063 006F9 1520    amarec A=DAT0  P
1064 006FD 305             LC(1)   END
1065 00700 902             ?A=C    P
1066 00703 B4              GOYES   return  match found
1067 00705 307             LC(1)   CLOS
1068 00708 902             ?A#C    P
1069 0070B 11              GOYES   amaclo  "closure"
1070 0070D 79F0            GOSUB   omatch
1071 00711 5C3             GONC    return  no match possible
1072 00714 7E71            GOSUB   nxtpat  D0 := ' motif suivant
1073 00718 60EF            GOTO    amarec
1074 0071C
1075 0071C          *
1076 0071C          * Closure trouvée. Les ennuis commencent...
1077 0071C          *
1078 0071C 160     amaclo D0=D0+  CLOSS   passer la closure
1079 0071F DB              C=D     A
1080 00721 D5              B=C     A       B(A) := pos. initiale pour closure
1081 00723 8E0000  acl010 GOSUBL  =lookah
1082 00729 490             GOC     acl100  on a fini la closure
1083 0072C 7AD0            GOSUB   omatch
1084 00730 42F             GOC     acl010  tant qu'on trouve
1085 00733          *
1086 00733          * On est arrivé au maximum. Faut revenir en arrière sur D1
1087 00733          * while D=R
1088 00733          *   do
1089 00733          *     if amatch (reste) return TRUE
1090 00733          *
1091 00733 7F51    acl100 GOSUB   nxtpat  D0 est bien positionné
1092 00737
1093 00737 6D80    acl150 GOTO    call    appel recursif...
1094 0073B 421     aclret GOC     return  retour si amatch = TRUE
1095 0073E 1C1             D1=D1-  2
1096 00741 E7              D=D+1   A
1097 00743 DB              C=D     A
1098 00745 8B9             ?C<=B   A
1099 00748 FE              GOYES   acl150
1100 0074A          *
```



```
1101 0074A          * on est arrivé au bout sans trouver, alors :
1102 0074A          *
1103 0074A 6300            GOTO    return  et on a Cy = 0
1104 0074E
1105 0074E          *
1106 0074E          * Appels et retours récursifs...
1107 0074E          *
1108 0074E AC0     return A=0     S       sauvegarde de la carry
1109 00751 550             GONC    ret010
1110 00754 B44             A=A+1   S
1111 00757
1112 00757 118     ret010 C=R0
1113 0075A 816             CSRC            CSRC3
1114 0075D 816             CSRC
1115 00760 816             CSRC
1116 00763 DA              A=C     A       A(A) := ' début
1117 00765 816             CSRC            CSRC5
1118 00768 816             CSRC
1119 0076B 816             CSRC
1120 0076E 816             CSRC
1121 00771 816             CSRC            C(A) := ' courant
1122 00774 8A2             ?A=C    A
1123 00777 74              GOYES   ret090
1124 00779 137             CD1EX
1125 0077C          * D1 := ' courant ; C(A) := ancienne valeur de D1
1126 0077C
1127 0077C          *
1128 0077C          * Correction du 88/10/14 : lorsqu'on a trouvé un
1129 0077C          * match, il ne faut surtout pas restaurer D1, D etc.
1130 0077C          * mais au contraire les préserver car ce sont eux qui
1131 0077C          * donneront la longueur du match.
1132 0077C          *
1133 0077C 948             ?A=0    S       retour sans avoir trouvé
1134 0077F B0              GOYES   ret020
1135 00781 1C9             D1=D1-  10      dépilage sans restauration
1136 00784 1C9             D1=D1-  10
1137 00787          * C(A) = ancienne valeur de D1 (celle qu'il faut renvoyer)
1138 00787 512             GONC    ret050  B.E.T.
1139 0078A          *
1140 0078A          * Fin de la correction du 88/10/14
1141 0078A          *
1142 0078A 1C4     ret020 D1=D1-  5       D0
1143 0078D 147             C=DAT1  A
1144 00790 134             D0=C
1145 00793
1146 00793 1C4             D1=D1-  5       B
1147 00796 147             C=DAT1  A
1148 00799 D5              B=C     A
1149 0079B
1150 0079B 1C4             D1=D1-  5       D
```



```
1151 0079E 147             C=DAT1  A
1152 007A1 D7              D=C     A
1153 007A3
1154 007A3 1C4             D1=D1-  5       D1
1155 007A6 147             C=DAT1  A
1156 007A9          *
1157 007A9          * Ce label est un ajout du 88/10/14, faisant pendant
1158 007A9          * à ce qui se trouve au dessus de ret0C0.
1159 007A9          *
1160 007A9          ret050
1161 007A9          *
1162 007A9          * C(A) = valeur de D1 à renvoyer
1163 007A9          * D1 = nouveau pointeur de pile réajusté
1164 007A9          *
1165 007A9 137             CD1EX           C(A) := ' courant réactualisé
1166 007AC
1167 007AC 8F0000          GOSBVL  =CSLC8  C(12-8) := pointeur de pile
1168 007B3 108             R0=C            restauration du pointeur de pile
1169 007B6 7400            GOSUB   ret090  positionne la Cy
1170 007BA 608F            GOTO    aclret  retour de l'appel "récursif"
1171 007BE
1172 007BE 940     ret090 ?A#0    S
1173 007C1 00              RTNYES          Cy = 1
1174 007C3 01              RTN             Cy = 0
1175 007C5
1176 007C5 118     call   C=R0            12-S = ' courant ; 7-2 = ' début
1177 007C8 8F0000          GOSBVL  =CSRC8  C(A) := ' courant
1178 007CF 137             CD1EX           D1 := ' pile ; C(A) := ancien D1
1179 007D2 1504            DAT1=C  5
1180 007D6 174             D1=D1+  5
1181 007D9
1182 007D9 DF              CDEX    A       D
1183 007DB 145             DAT1=C  A
1184 007DE 174             D1=D1+  5
1185 007E1 DF              CDEX    A
1186 007E3
1187 007E3 DD              CBEX    A       B
1188 007E5 145             DAT1=C  A
1189 007E8 174             D1=D1+  5
1190 007EB DD              CBEX    A
1191 007ED
1192 007ED 136             CD0EX   A       D0
1193 007F0 145             DAT1=C  A
1194 007F3 174             D1=D1+  5
1195 007F6 136             CD0EX   A
1196 007F9
1197 007F9 137             CD1EX           C(A) := ' courant
1198 007FC 8F0000          GOSBVL  =CSLC8
1199 00803 108             R0=C
1200 00806 62FE            GOTO    amarec
```

```
Page 025          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

1201 0080A
1202 0080A          ********************************************************
1203 0080A          • omatch
1204 0080A          •
1205 0080A          • But: chercher le motif courant dans la chaine
1206 0080A          • Entree:
1207 0080A          •   - D0 = ^ motif courant
1208 0080A          •   - D1 = ^ caractère courant, D(A) = nb de car. restant
1209 0080A          • Sortie:
1210 0080A          •   - Cy = 0 : match not found
1211 0080A          •   - Cy = 1 : match found
1212 0080A          •   - D0 inchangé
1213 0080A          •   - D1 et D(A) actualisés si besoin
1214 0080A          • Abime: A(A), C(W)
1215 0080A          • Appelle: lookah, rhcteg, TBLJMC, gettab
1216 0080A          • Niveaux: 1
1217 0080A          • Detail: cf. "Software tools in Pascal"
1218 0080A          • Historique:
1219 0080A          •   88/06/26: PD/JT conception & codage
1220 0080A          •   88/10/07: PD/JT retrait du compte dans R1(A) pour amatch
1221 0080A          ********************************************************
1222 0080A
1223 0080A 1564     omatch C=DAT0  S       C(S) := motif
1224 0080E A4E             C=C-1   S
1225 00811 5F1             GONC    oma10
1226 00814          •
1227 00814          • Déstructuration complète du code pour gagner le plus de
1228 00814          • cycles possibles
1229 00814          •
1230 00814          omCHAR
1231 00814          • GOSUBL =lookah
1232 00814 8AB             ?D=0    A
1233 00817 83              GOYES   rtncc
1234 00819 14B             A=DAT1  B
1235 0081C          • fin de GOSUBL =lookah
1236 0081C 160             D0=D0+  1
1237 0081F 14E             C=DAT0  B       C(B) := caractère trouve
1238 00822 180             D0=D0-  1
1239 00825 966             ?A#C    B       A(B) = caractère lu par lookah
1240 00828 72              GOYES   rtncc   match not found
1241 0082A          • GOSUB =rhcteg on avance
1242 0082A 171             D1=D1+  2
1243 0082D CF              D=D-1   A
1244 0082F          • fin de GOSUB =rhcteg
1245 0082F 02              RTNSC           match found
1246 00831
1247 00831 A4E      oma10  C=C-1   S
1248 00834 4E1             GOC     omEOL
1249 00837 A4E             C=C-1   S
1250 0083A 4E1             GOC     omANY
```

```
Page 026          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

1251 0083D A4E             C=C-1   S
1252 00840 41C             GOC     omCCL
1253 00843 A4E             C=C-1   S
1254 00846 4B1             GOC     omNCCL
1255 00849 A4E             C=C-1   S
1256 0084C 440             GOC     omEND
1257 0084F 03       rtncc  RTNCC           CLOS ou BOL : match not found
1258 00851
1259 00851 02       omEND  RTNSC           tout ma[rt]ch !
1260 00853
1261 00853          •
1262 00853          • omBOL : enlève le 88/10/07 car le cas '^' est traité à
1263 00853          • part dans POSADR.
1264 00853          •
1265 00853
1266 00853 8C0000   omEOL  GOLONG  =lookah Cy = match
1267 00859
1268 00859 7180     omANY  GOSUB   =rhcteg
1269 0085D 41F             GOC     rtncc
1270 00860 02              RTNSC           match found
1271 00862          omCCL
1272 00862 8E0000   omNCCL GOSUBL  =lookah
1273 00868 46E             GOC     rtncc
1274 0086B 160             D0=D0+  1
1275 0086E 7EAC            GOSUB   gettab  D0 := ^ élément
1276 00872 1564            C=DAT0  S       C(S) := tab [caractère]
1277 00876 134             D0=C
1278 00879 180             D0=D0-  1       D0 := ^ CCL ou NCCL
1279 0087C 1520            A=DAT0  P       A(0) := CCL ou NCCL
1280 00880 303             LC(1)   CCL
1281 00883 902             ?A=C    P
1282 00886 50              GOYES   omCC10
1283 00888 A4E             C=C-1   S       si NCCL : C(S) := not tab [carlu]
1284 0088B 94A      omCC10 ?C=0    S
1285 0088E 1C              GOYES   rtncc   no match
1286 00890 7A40            GOSUB   =rhcteg
1287 00894 02              RTNSC           match found
1288 00896
1289 00896          ********************************************************
1290 00896          • nxtpat
1291 00896          •
1292 00896          • But: positionner D0 sur le motif suivant
1293 00896          • Entree:
1294 00896          •   - D0 := ^ motif courant
1295 00896          • Sortie:
1296 00896          •   - D0 := ^ motif suivant
1297 00896          • Abime: A(A), C(A), D0
1298 00896          • Appelle: -
1299 00896          • Niveaux: 1
1300 00896          • Detail: cf. "Software tools in Pascal", c'est la nouvelle
```

```
Page 027          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

1301 00896          •   fonction "patsize"
1302 00896          • Historique:
1303 00896          •   88/06/26: PD/JT conception & codage
1304 00896          ********************************************************
1305 00896
1306 00896 1524     nxtpat A=DAT0  S
1307 00898 94C             ?A#0    S
1308 0089D 70              GOYES   ps05
1309 0089F 162             D0=D0+  CHARS
1310 008A2 01              RTN
1311 008A4
1312 008A4 7810     ps05   GOSUB   ps10
1313 008A8 300             CON(3)  CHARS   inutile
1314 008AB 100             CON(3)  EOLS
1315 008AE 100             CON(3)  ANYS
1316 008B1 101             CON(3)  CCLS
1317 008B4 101             CON(3)  NCCLS
1318 008B7 000             CON(3)  0       END
1319 008BA 000             CON(3)  0       BOL
1320 008BD 100             CON(3)  CLOSS
1321 008C0 D0       ps10   A=0     A
1322 008C2 1520            A=DAT0  P       A(A) := motif courant
1323 008C6 D6              C=A     A
1324 008C8 CA              A=A+C   A
1325 008CA CA              A=A+C   A       A(A) := 3*motif
1326 008CC 07              C=RSTK          C(A) := ^ début de la table
1327 008CE C2              C=C+A   A
1328 008D0 136             CD0EX
1329 008D3 15A2            A=DAT0  3
1330 008D7 C2              C=A+C   A
1331 008D9 134             D0=C
1332 008DC 01              RTN
1333 008DE
1334 008DE          ********************************************************
1335 008DE          • rhcteg
1336 008DE          •
1337 008DE          • But: lire un caractère, et passer au suivant (getchr)
1338 008DE          • Entree:
1339 008DE          •   - D1 = ^ chaîne
1340 008DE          •   - D(A) = nb de caractères
1341 008DE          • Sortie:
1342 008DE          •   - Cy = 1 : fin de la chaîne
1343 008DE          •   - Cy = 0, A(B) = caractère lu
1344 008DE          • Abime: A(B)
1345 008DE          • Appelle: -
1346 008DE          • Niveaux: 0
1347 008DE          • Historique:
1348 008DE          •   88/06/26: PD/JT conception & pompage (getchr et lookah)
1349 008DE          ********************************************************
1350 008DE
```

```
Page 028          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

1351 008DE 8AB      =rhcteg ?D=0    A
1352 008E1 00               RTNYES
1353 008E3 CF               D=D-1   A
1354 008E5 14B              A=DAT1  B
1355 008E8 171              D1=D1+  2
1356 008EB 01               RTN             Cy = 0
1357 008ED
1358 008ED          ********************************************************
1359 008ED          • UPDTR0
1360 008ED          •
1361 008ED          • But: mettre à jour l'adresse du buffer contenue dans
1362 008ED          •   R0(7-3) à partir de son ID contenu dans R0(2-0).
1363 008ED          • Entree:
1364 008ED          •   - R0(2-0) = ID du buffer
1365 008ED          • Sortie:
1366 008ED          •   - Cy = 1 : erreur
1367 008ED          •       - C(3-0) = numéro d'erreur (System Error)
1368 008ED          •   - Cy = 0 : pas d'erreur
1369 008ED          •       - R0(7-0) = caractéristique complète du buffer
1370 008ED          •       - D1 = ^ past buffer header
1371 008ED          • Abime: A, C, D1
1372 008ED          • Appelle: I/OFND0
1373 008ED          • Niveaux: 1
1374 008ED          • Historique:
1375 008ED          •   88/10/16: PD/JT séparation de POSADR
1376 008ED          ********************************************************
1377 008ED
1378 008ED 693B      sysTem GOTO    system
1379 008F1
1380 008F1 118      =UPDTR0 C=R0            C(X) := buffer id
1381 008F4 8F00000          GOSBVL  =IOFND0
1382 008FB 51F              GONC    sysTem  "System Error" + Cy = 1
1383 008FE          •
1384 008FE          • C(X) = ID
1385 008FE          • D1 = ^ past buffer header
1386 008FE          •
1387 008FE 118              C=R0            actualisation de R0(7-2)
1388 00901 8F00000          GOSBVL  =CSRC3
1389 00905 137              CD1EX
1390 00908 135              D1=C
1391 0090E 8F00000          GOSBVL  =CSLC3
1392 00915 108              R0=C
1393 00918 03               RTNCC           pas d'erreur
1394 0091A
1395 0091A          ********************************************************
1396 0091A          • ENDPOS
1397 0091A          •
1398 0091A          • But: terminer une recherche générique, c'est à dire
1399 0091A          •   effacer le buffer.
1400 0091A          • Entree:
```

```
Page 029          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4

1401 0091A          •   - R0(7-2) = caractéristique du buffer
1402 0091A          • Sortie: -
1403 0091A          • Abime: A-C, D1, D0
1404 0091A          • Appelle: I/ODAL
1405 0091A          • Niveaux: 2
1406 0091A          • Historique:
1407 0091A          •   88/07/02: PD/JT conception & codage
1408 0091A          ********************************************************
1409 0091A
1410 0091A 118      =ENDPOS C=R0
1411 0091D 8D00000          GOVLNG  =I/ODAL
1412 00924
```

```
Page 030          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1414 00924
1415 00924          ********************************************************
1416 00924          • CALCab
1417 00924          •
1418 00924          • But: calculer les coefficients a et b permettant de
1419 00924          •   connaitre la longueur de la chaine remplacée. La routine
1420 00924          •   REPLIN les utilise.
1421 00924          • Entree:
1422 00924          •   - R3 = motif de remplacement (dans le sens Math Stack)
1423 00924          •   - MODE71 = 1 si mode HP-71, 0 si mode HP-UX
1424 00924          • Sortie:
1425 00924          •   - les zones "coeffa" et "coeffb" sont remplies.
1426 00924          • Abime: A(S), A(A), B(A), C(S), C(A), D(A), D1
1427 00924          • Niveaux: 1 (utilisation de RSTK)
1428 00924          • Note : Le registre R3 se décompose en
1429 00924          •   - R3(8-5) = adresse du prochain caractère à lire
1430 00924          •   - R3(4-0) = nombre de caractères dans la chaine
1431 00924          • Detail : les coefficients a et b sont respectivement le
1432 00924          •   nombre de caractères constants (normaux) et le nombre de
1433 00924          •   caractères '&' dans la chaine.
1434 00924          •   Dans ces conditions, si "n" est la longueur de
1435 00924          •   l'occurrence trouvée, f(n) = a + bn est la longueur de
1436 00924          •   la chaine après remplacement.
1437 00924          • Historique:
1438 00924          •   88/10/08: PD/JT conception
1439 00924          ********************************************************
1440 00924
1441 00924 11B      =CALCab C=R3
1442 00927 D7               D=C     A       D(A) := longueur du motif
1443 00929 8F00000          GOSBVL  =CSRC5  C(A) := ^ premier caractère du motif
1444 0092D 1F00000          D1=(5)  =MODE71
1445 00931 1574             C=DAT1  S       C(S) := 0 si mode HP-UX
1446 00933 135              D1=C            D1 := ^ premier caractère du motif
1447 00936
1448 00936 AC0              A=0     S       A(S) := pas de "\"
1449 00941 D1               B=0     A       a := 0 (caractère normal)
1450 00943 D2               C=0     A       b := 0 (caractère '&')
1451 00945 06               RSTK=C
1452 00947
1453 00947 94E              ?C#0    S       mode HP-71 ?
1454 0094A E3               GOYES   CALC50  oui
1455 0094C          •
1456 0094C          • Mode HP-UX
1457 0094C          •
1458 0094C 8E0000   CALC10 GOSUBL  =getchr
1459 00952 413              GOC     calc99  on est arrivé au bout
1460 00955
1461 00955 94C              ?A#0    S       selon état
1462 00958 32               GOYES   CALC25  "\" avant ce caractère
1463 0095A          • pas de "\"
```

```
Page 031          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1464 0095A 31C5             LC(2)   %ESC    "\"
1465 0095E 962              ?A=C    B
1466 00961 41               GOYES   CALC20  état := 1 ; next one
1467 00963 3162             LC(2)   %REP    "&"
1468 00967 966              ?A#C    B
1469 0096A 41               GOYES   CALC30  caractère normal
1470 0096C          • caractère '&' générique
1471 0096C 07               C=RSTK
1472 0096E E6               C=C+1   A       b++
1473 00970 06               RSTK=C
1474 00972 59D              GONC    CALC10  next one (B.E.T.)
1475 00975 B44      CALC20 A=A+1    S       état := 1
1476 00978 53D              GONC    CALC10  B.E.T.
1477 0097B          • Caractère normal
1478 0097B AC0      CALC25 A=0      S       état := 0
1479 0097E E5       CALC30 B=B+1    A       a++
1480 00980 6BCF             GOTO    CALC10
1481 00984
1482 00984 6180     calc99 GOTO     CALC99
1483 00988
1484 00988          •
1485 00988          • Mode HP-71
1486 00988          •
1487 00988 8E0000   CALC50 GOSUBL  =getchr
1488 0098E 477              GOC     CALC99  on a fini
1489 00991          •
1490 00991          • Automate d'états fini :
1491 00991          • - état = 0 : normal
1492 00991          • - état = 1 : précédent caractère etait un "\"
1493 00991          • - état = 2 : caractères génériques OFF
1494 00991          • - état = 3 : état 2 + précédent caractère était un "\"
1495 00991          •
1496 00991 31C5             LC(2)   %TOGL   pour économiser des quartets
1497 00995
1498 00995 AC6              C=A     S       C(S) := état de l'automate
1499 00998 A4E              C=C-1   S
1500 0099B 4F1              GOC     CALC60
1501 0099E A4E              C=C-1   S
1502 009A1 442              GOC     CALC61
1503 009A4 A4E              C=C-1   S
1504 009A7 414              GOC     CALC62
1505 009AA 962      CALC63 ?A=C     B
1506 009AD 80               GOYES   CAL630
1507 009AF AC0              A=0     S       état := 0
1508 009B2 4D4              GOC     CALC70  B.E.T.
1509 009B5 A4C      CAL630 A=A-1    S       état := 2
1510 009B8 574              GONC    CALC70  B.E.T.
1511 009BB
1512 009BB 966      CALC60 ?A#C     B
1513 009BE 24               GOYES   CALC70  a++ (B.E.T.)
```

```
Page 032          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1514 009C0 B44              A=A+1   S       état := 1
1515 009C3 54C              GONC    CALC50  B.E.T.
1516 009C6
1517 009C6 962      CALC61 ?A=C     B
1518 009C9 11               GOYES   CAL610  "\"
1519 009CB 3162             LC(2)   %REP    "&"
1520 009CF 962              ?A=C    B
1521 009D2 E0               GOYES   CAL615  "&"
1522 009D4 B44              A=A+1   S       état := 2
1523 009D7 58C              GONC    CALC70  a++ (B.E.T.)
1524 009DA AL0      CAL610 A=0      S
1525 009DD 4D2              GOC     CALC70  a++ (B.E.T.)
1526 009E0 07       CAL615 C=RSTK
1527 009E2 E6               C=C+1   A
1528 009E4 06               RSTK=C
1529 009E6 61A              GONC    CALC50  B.E.T.
1530 009E9
1531 009E9 962      CALC62 ?A=C     B
1532 009EC E0               GOYES   CAL620  "\"
1533 009EE 3162             LC(2)   %REP    "&"
1534 009F2 966              ?A#C    B
1535 009F5 B0               GOYES   CALC70  a++
1536 009F7 58E              GONC    CAL615  spaghetti... mais c'est un B.E.T.
1537 009FA B44      CAL620 A=A+1    S       état := 3
1538 009FD 5A8              GONC    CALC50  B.E.T.
1539 00A00
1540 00A00 E5       CALC70 B=B+1    A       a++
1541 00A02 658F             GOTO    CALC50
1542 00A06
1543 00A06          •
1544 00A06          • Etat final
1545 00A06          • Stockage des coefficients
1546 00A06          •
1547 00A06 1F00000  CALC99 D1=(5)   =coeffa
1548 00A0D D3               C=B     A
1549 00A0F 145              DAT1=C  A       coeffa := B(A)
1550 00A12 174              D1=D1+  5       coeffb
1551 00A15 07               C=RSTK
1552 00A17 145              DAT1=C  A       coeffb := RSTK
1553 00A1A 01               RTN
1554 00A1C
1555 00A1C          ********************************************************
1556 00A1C          • REPLIN
1557 00A1C          •
1558 00A1C          • But: remplacer un motif par un autre dans une chaine ou
1559 00A1C          •   il y a déjà une occurrence du motif.
1560 00A1C          • Entree:
1561 00A1C          •   - R0(7-0) = caractéristique du buffer (chaine compilee)
1562 00A1C          •   - (OUTBS..AVMEMS) = chaine à traiter
1563 00A1C          •   - AVMEME indique la fin de la mémoire disponible
```

```
Page 033          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1564 00A1C          •   - R3 = motif de remplacement (dans le sens Math Stack)
1565 00A1C          •   - D0 = ° occurrence trouvée
1566 00A1C          •   - C(A) = longueur de l'occurrence
1567 00A1C          •   - coeffa et coeffb calculés par CALCab
1568 00A1C          •   - QUERY = 1 si recherche interactive ("?"), 0 sinon
1569 00A1C          •   - A(A) = numéro de la ligne en cours (si QUERY = 1)
1570 00A1C          •   - MODE71 = 1 si chaines génériques HP-71, 0 si HP-UX
1571 00A1C          • Sortie:
1572 00A1C          •   - Cy = 0 : ok
1573 00A1C          •     suivant C(0) :
1574 00A1C          •       0 : quit sans modification
1575 00A1C          •       1 : quit avec quelque chose déjà modifié
1576 00A1C          •       2 : ok sans modification
1577 00A1C          •       3 : ok avec quelque chose de modifié
1578 00A1C          •     (OUTBS..AVMEMS) = zone où est stockée la ligne finale
1579 00A1C          •   - Cy = 1 : erreur
1580 00A1C          •     C(3-0) = numéro d'erreur
1581 00A1C          • Abime: A-D, R0(12-8), R1-R2, R3(10), R4(15-11), R4(9-5),
1582 00A1C          •   D0, D1
1583 00A1C          • Appelle: query, POSADR
1584 00A1C          • Niveaux: 4 (POSADR) ou 5 (query)
1585 00A1C          • Note : Le registre R3 se décompose en
1586 00A1C          •   - R3(9-5) = adresse du prochain caractère à lire
1587 00A1C          •   - R3(4-0) = nombre de caractères dans la chaine
1588 00A1C          • Algorithme:
1589 00A1C          •   répéter
1590 00A1C          •     procéder au remplacement
1591 00A1C          •     si mode = "?"
1592 00A1C          •       alors
1593 00A1C          •         demander confirmation
1594 00A1C          •         selon réponse
1595 00A1C          •           Q : sortir avec le code "QUIT"
1596 00A1C          •           N : annuler le remplacement
1597 00A1C          •           Y :
1598 00A1C          •         fin selon
1599 00A1C          •     fin si
1600 00A1C          •     rechercher l'occurrence suivante
1601 00A1L          •   jusqu'à ce qu'il n'y ait plus d'occurrence
1602 00A1C          • Historique:
1603 00A1C          •   88/10/08: PD/JT conception
1604 00A1C          •   88/10/14: PD/JT correction du cas "match nul"
1605 00A1C          •   88/10/15: PD/JT correction du cas "R/°//"
1606 00A1C          ********************************************************
1607 00A1C
1608 00A1C 330000     Errmem LC(4)   =eMEM   Insufficient Memory
1609 00A20 02         errrtn RTNSC           Cy = 1 => erreur
1610 00A24
1611 00A24          =REPLIN
1612 00A24          •
1613 00A24          • Sauvegarder la valeur de A(A) dans R4(9-5). On utilise R4,
```

```
Page 034          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1614 00A24          • les temps sont vraiment durs...
1615 00A24          •
1616 00A24 8F00000          GOSBVL  =ASLW5
1617 00A28 104              R4=A            R4(9-5) = no de la ligne en cours
1618 00A2E          •
1619 00A2E          • Mettre à 0 l'indicateur de modifications
1620 00A2E          •
1621 00A2E 113              A=R3
1622 00A31 2A               P=      10
1623 00A33 A80              A=0     P
1624 00A36 103              R3=A
1625 00A39 20               P=      0
1626 00A3B          •
1627 00A3B          • Boucle tant qu'il y a des occurrences. En fait, c'est
1628 00A3B          • une boucle "repeat..until". Voir à ce propos (différences
1629 00A3B          • entre les boucles WHILE et REPEAT) l'excellent article de
1630 00A3B          • notre excellent camarade Janick Taillandier dans
1631 00A3B          • l'excellent JPC 31.
1632 00A3B          •
1633 00A3B           RPL100
1634 00A3B          •
1635 00A3B          • Assertions :
1636 00A3B          • - D0 = ° occurrence trouvée dans la chaine à traiter
1637 00A3B          • - C(A) = longueur de l'occurrence (en caractères)
1638 00A3B          • - R0(7-0) = caractéristique du buffer
1639 00A3B          • - R3 = motif de remplacement (dans le sens Math Stack)
1640 00A3B          • - R4(9-5) = numéro de la ligne courante (si mode = "?")
1641 00A3B          •
1642 00A3B          • Procéder au remplacement :
1643 00A3B          •
1644 00A3B          • n = longueur de l'occurrence trouvée (dans RSTK)
1645 00A3B          • Calcul de f(n) = a + b•n
1646 00A3B          •
1647 00A3B 06               RSTK=C          RSTK := n (longueur de l'occurrence)
1648 00A3D AF0              A=0     W
1649 00A40 DA               A=C     A       A(W) := n
1650 00A42 1F00000          D1=(5)  =coeffb
1651 00A49 AF2              C=0     W
1652 00A4C 147              C=DAT1  A       C(W) := b
1653 00A4F 8F00000          GOSBVL  =MPY    résultat dans A, B et C
1654 00A55 AF0              A=0     W       vérification du résultat
1655 00A53 DA               A=C     A
1656 00A5B 8A6              ?A#C    A       <> 1048575 ?
1657 00A5E EB               GOYES   Errmem  non : Not Enough Memory
1658 00A60          • A(A) = B(A) = C(A) := b•n
1659 00A60 1C4              D1=D1-  5       D1=(5) coeffa
1660 00A63 143              A=DAT1  A
1661 00A66 CA               A=A+C   A       A(A) := f(n)    (a + b•n)
1662 00A68 43B              GOC     Errmem  Overflow
1663 00A6B          •
```

```
Page 035          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1664 00A6B          • Assertions :
1665 00A6B          • - RSTK = n
1666 00A6B          • - A(A) = f(n)
1667 00A6B          •
1668 00A6B 07               C=RSTK          C(A) := n
1669 00A6D D8               B=A     A
1670 00A6F E1               B=B-C   A       B(A) := f(n) - n
1671 00A71 8F00000          GOSBVL  =ASLW5  A(9-5) := f(n)
1672 00A78 DA               A=C     A       A(4-0) := n
1673 00A7A 101              R1=A            R1(9-5) := f(n); R1(A) := n
1674 00A7D
1675 00A7D 1F00000          D1=(5)  =AVMEMS
1676 00A84 143              A=DAT1  A       A(A) := AVMEMS
1677 00A87 D6               C=A     A
1678 00A89 C9               C=C+B   A       C(A) := AVMEMS + f(n) - n
1679 00A8B C9               C=C+B   A       C(A) := AVMEMS + 2•(f(n) - n)
1680 00A8D 8B2              ?C>A    A
1681 00A90 40               GOYES   RPL120
1682 00A92 D6               C=A     A       C(A) := MAX (...)
1683 00A94 8F00000   RPL120 GOSBVL  =CSLC5  C(9-5) := MAX (...)
1684 00A9B 136              CD0EX           C(A) := D0
1685 00A9E 10A              R2=C
1686 00AA1
1687 00AA1          •
1688 00AA1          • Assertions :
1689 00AA1          •   R1(9-5) = f(n)      (longueur du remplacement)
1690 00AA1          •   R1(4-0) = n         (longueur de l'occurrence)
1691 00AA1          •   R2(9-5) = MAX (AVMEMS, AVMEMS + 2•(f(n) - n))
1692 00AA1          •   R2(4-0) = D0        (début de l'occurrence)
1693 00AA1          •
1694 00AA1
1695 00AA1          •
1696 00AA1          • Le décalage se produit en plusieurs étapes :
1697 00AA1          • - copier l'occurrence à l'adresse
1698 00AA1          •   MAX (AVMEMS, AVMEMS + f(n) - n)
1699 00AA1          • - copier ce qu'il y a entre la fin de l'occurrence et
1700 00AA1          •   AVMEMS à l'adresse (début de l'occurrence + f(n))
1701 00AA1          • - remplacer l'occurrence par le motif
1702 00AA1          • - mettre à jour AVMEMS
1703 00AA1          •
1704 00AA1
1705 00AA1          •
1706 00AA1          • Il y a suffisamment de mémoire : on demande 2•f(n) ?
1707 00AA1          •
1708 00AA1 119              C=R1
1709 00AA4 8F00000          GOSBVL  =CSRC5  C(A) := f(n)
1710 00AAB C6               C=C+C   A       C(A) := 2•f(n)
1711 00AAD 21               P=      1       P#0 pour ne pas avoir de LEEWAY
1712 00AAF 8F00000          GOSBVL  =MEMCKL Memory Check without LEEWAY
1713 00AB6          • P=0 en sortie de MEMCKL
```

```
Page 036          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1714 00AB6 400              RTNC            Bip ! "Insufficient Memory"
1715 00AB9
1716 00AB9          •
1717 00AB9          • Préparation des paramètres pour MOVE•M :
1718 00AB9          • - A(A) = source address = D0
1719 00AB9          • - C(A) = destination address = MAX (AVMEMS, AVMEMS+f(n)-n)
1720 00AB9          • - B(A) = longueur en nibs = 2•n
1721 00AB9          •
1722 00AB9 119              C=R1
1723 00ABC D5               B=C     A
1724 00ABE C5               B=B+B   A       B(A) := 2•n
1725 00AC0 11A              C=R2
1726 00AC3 DA               A=C     A       A(A) := D0
1727 00AC5 8F00000          GOSBVL  =CSRC5  C(A) := MAX (...)
1728 00ACC 8F00000          GOSBVL  =MOVE•M
1729 00AD3
1730 00AD3          •
1731 00AD3          • Préparation des paramètres pour MOVE•M :
1732 00AD3          • - A(A) = source address = D0 + 2•n
1733 00AD3          • - C(A) = destination address = D0 + 2•f(n)
1734 00AD3          • - B(A) = longueur en nibs = AVMEMS - (D0 + 2•n)
1735 00AD3          •
1736 00AD3 119              C=R1            C(A) := n
1737 00AD6 112              A=R2            A(A) := D0
1738 00AD9 C6               C=C+C   A
1739 00ADB CA               A=C+A   A       A(A) := D0 + 2•n
1740 00ADD
1741 00ADD 1B00000          D0=(5)  =AVMEMS
1742 00AE4 146              C=DAT0  A
1743 00AE7 E2               C=C-A   A
1744 00AE9 D5               B=C     A       B(A) := AVMEMS - (D0 + 2•n)
1745 00AEB
1746 00AEB 8F00000          GOSBVL  =CSRC5  C(A) := f(n)
1747 00AF2 D7               D=C     A
1748 00AF4 11A              C=R2
1749 00AF7 CB               C=C+D   A
1750 00AF9 CB               C=C+D   A       C(A) := D0 + 2•f(n)
1751 00AFB          •
1752 00AFB          • On pourrait peut-être gagner quelques cycles si le
1753 00AFB          • déplacement avait lieu à la même adresse, mais les
1754 00AFB          • routines appelées par MOVE•M détectent presque
1755 00AFB          • immédiatement ce cas. Il n'y a donc pas lieu de
1756 00AFB          • s'en inquiéter. Nous n'évitons donc pas ce MOVE•M.
1757 00AFB          • Tant pis pour jt (qui se serait bien passé d'un MOVE).
1758 00AFB          •
1759 00AFB 8F00000          GOSBVL  =MOVE•M
1760 00B02          •
1761 00B02          • remplacement de la chaine à l'adresse D0 par la
1762 00B02          • chaine à l'adresse R3, le motif original étant à
1763 00B02          • l'adresse MAX (AVMEMS, AVMEMS + f(n) - n)
```

```
Page 037          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1764 00B02          •
1765 00B02 11A              C=R2
1766 00B05 134              D0=C            D0 := ^ espace à remplir
1767 00B08
1768 00B08 1F0000           D1=(5)  =MODE71 1 si mode HP-71 ou 0 si mode HP-Ux
1769 00B0F 1534             A=DAT1  S       A(S) := 1 si mode HP-71
1770 00B13
1771 00B13 11B              C=R3
1772 00B16 D7               D=C     A       D(A) := nombre de caractères
1773 00B18 8F0000           GOSBVL  =CSRC5
1774 00B1F 135              D1=C            D1 := ^ premier caractère dans M.S.
1775 00B22
1776 00B22 94C              ?A#0    S       mode HP-71 ?
1777 00B25 64               GOYES   RPL300
1778 00B27          •
1779 00B27          • Mode HP-Ux
1780 00B27          •
1781 00B27           RPL200
1782 00B27          •      A=0     S       A était déjà égal à 0
1783 00B27
1784 00B27           RPL290
1785 00B27          •
1786 00B27          • Le label RPL290 est là pour éviter des GOTO qui branchent
1787 00B27          • à des GOTO.
1788 00B27          •
1789 00B27 8E0000    RPL210 GOSUBL  =getchr
1790 00B2D 493              GOC     rpl500
1791 00B30
1792 00B30 94I              ?A#0    S       précédent caractère = "\"
1793 00B33 D1               GOYES   RPL240
1794 00B35 3105             LC(2)   xESC
1795 00B39 962              ?A=C    B
1796 00B3C E0               GOYES   RPL220
1797 00B3E 3106             LC(2)   XREP    "&"
1798 00B42 962              ?A=C    B
1799 00B45 11               GOYES   RPL270  rplc
1800 00B47 561              GONC    RPL280  caractère normal (B.E.T.)
1801 00B4A B44       RPL220 A=A+1    S       état := 1
1802 00B4D 590              GONC    RPL290  next one
1803 00B50
1804 00B50 A4C       RPL240 A=A-1    S       état := 0
1805 00B53 5A0              GONC    RPL280  B.E.T.
1806 00B56
1807 00B56 706C      RPL270 GOSUB    rplc
1808 00B5A 6CCF             GOTO    RPL290
1809 00B5E 148       RPL280 DAT0=A   B       ajoute un caractère
1810 00B61 161              D0=D0+  2
1811 00B64 52C              GONC    RPL210  B.E.T.
1812 00B67
1813 00B67 6C80      rpl500 GOTO    RPL500
```

```
Page 038          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1814 00B6B
1815 00B6B          •
1816 00B6B          • Mode HP-71
1817 00B6B          •
1818 00B6B A40       RPL300 A=0     S
1819 00B6E
1820 00B6E           RPL390
1821 00B6E          •
1822 00B6E          • L'étiquette RPL330 est là pour éviter les GOTO qui
1823 00B6E          • branchent à des GOTO.
1824 00B6E          •
1825 00B6E 8E4000    RPL310 GOSUBL  =getchr
1826 00B74 4F7              GOC     RPL500
1827 00B77
1828 00B77 3105             LC(2)   xTOGL
1829 00B7B A0E              C=A     S       C(S) := état
1830 00B7E A4E              C=C-1   S
1831 00B81 4F1              GOC     RPL320
1832 00B84 A4E              C=C-1   S
1833 00B87 442              GOC     RPL321
1834 00B8A A4E              C=C-1   S
1835 00B8D 4E3              GOC     RPL322
1836 00B90
1837 00B90 96C       RPL323 ?A=C    B
1838 00B93 20               GOYES   RP3230
1839 00B95 A40              A=0     S       état := 0
1840 00B98 525              GONC    RPL380  ajouter normal (B.E.T.)
1841 00B9B A4C       RP3220 A=A-1    S       état := 2
1842 00B9E 504              GONC    RPL380  ajouter normal (B.E.T.)
1843 00BA1
1844 00BA1 966       RPL320 ?A#C    B
1845 00BA4 74               GOYES   RPL380  ajouter normal
1846 00BA6 B44              A=A+1   S
1847 00BA9 54C              GONC    RPL390  B.E.T.
1848 00BAC
1849 00BAC 962       RPL321 ?A=C    B
1850 00BAF 11               GOYES   RP3210  "\"
1851 00BB1 3162             LC(2)   SREP    "&"
1852 00BB5 962              ?A=C    B
1853 00BB8 E0               GOYES   RP3215  "&"
1854 00BBA B44              A=A+1   S       état := 2
1855 00BBD 502              GONC    RPL380  ajoute normal
1856 00BC0 A4C       RP3210 A=A-1    S       état := 0
1857 00BC3 57C              GONC    RPL380  ajoute normal (B.E.T.)
1858 00BC6 B44       RP3215 A=A+1    S       état := 2
1859 00BC9 591              GONC    RPL370  rplc (B.E.T.)
1860 00BCC
1861 00BCC 962       RPL322 ?A=C    B
1862 00BCF E0               GOYES   RP3220  "\"
1863 00BD1 3162             LC(2)   SREP    "&"
```

```
Page 039          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1864 00BD5 960              ?A=C    B
1865 00BD8 B0               GOYES   RPL370  rplc
1866 00BDA 501              GONC    RPL380  ajoute normal (B.E.T.)
1867 00BDD B44       RP3220 A=A+1    S       état := 3
1868 00BE0 5D8              GONC    RPL390  B.E.T.
1869 00BE3
1870 00BE3 73D1      RPL370 GOSUB    rplc
1871 00BE7 668F             rpl390 GOTO     RPL390
1872 00BEB 148       RPL380 DAT0=A   B       ajoute un caractère
1873 00BEE 161              D0=D0+  2
1874 00BF1 55F              GONC    rpl390  B.E.T.
1875 00BF4
1876 00BF4           RPL500
1877 00BF4          •
1878 00BF4          • Après tout ça, il faut actualiser AVMEMS
1879 00BF4          •
1880 00BF4 119              C=R1
1881 00BF7 DA               A=C     A       A(A) := n
1882 00BF9 8F0000           GOSBVL  =CSRC5  C(A) := f(n)
1883 00C00 E2               C=C-A   A
1884 00C02 C6               C=C+C   A       C(A) := 2*(f(n) - n)
1885 00C04 1F0000           D1=(5)  =AVMEMS
1886 00C0B 143              A=DAT1  A
1887 00C0E CA               A=A+C   A
1888 00C10 141              DAT1=A  A       AVMEMS += 2*(f(n) - n)
1889 00C13          •
1890 00C13          • Faut-il poser une question ?
1891 00C13          •
1892 00C13 1F0000           D1=(5)  =QUERY  mode interactif ("?")
1893 00C1A 1574             C=DAT1  S
1894 00C1E 94E              ?C#0    S       Mode interactif ?
1895 00C21 60               GOYES   RPL600  oui : on pose la question
1896 00C23 6ED0             GOTO    RPL800  non : reponse [Y] par défaut
1897 00C27          •
1898 00C27          • Mode interactif :
1899 00C27          •
1900 00C27          • Pour poser la question, il faut qu'il y ait une longueur
1901 00C27          • LIF en début de la chaine. Il faut donc mettre
1902 00C27          • (AVMEMS-OUTBS) en longueur LIF. AVMEMS a été actualisé,
1903 00C27          • entre RPL120 et RPL200.
1904 00C27          •
1905 00C27 A08       RPL600 A=0     M
1906 00C2A 1B0000           D0=(5)  =AVMEMS
1907 00C31 142              A=DAT0  A
1908 00C34 180              D0=D0-  (=AVMEMS)-(=OUTBS)
1909 00C37 145              C=DAT0  A
1910 00C3A 135              D1=C            D1 := ^ début ligne LIF (longueur)
1911 00C3D EA               A=A-C   A       A(A) := longueur en quartets + 4
1912 00C3F 810              ASRB            A(A) := longueur en octets + 2
1913 00C42 CC               A=A-1   A
```

```
Page 040          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1914 00C44 CC               A=A-1   A
1915 00C46 8F0000           GOSBVL  =SWPBYT
1916 00C4D 1F93             DAT1=A  4       longueur LIF
1917 00C51
1918 00C51          •
1919 00C51          • Poser la question, mais sauver d'abord R1 car "query"
1920 00C51          • l'abime.
1921 00C51          • La sauvegarde de R1 est faite en poussant R2 et R3 vers la
1922 00C51          • gauche.
1923 00C51          •
1924 00C51 111              A=R1
1925 00C54 11A              C=R2
1926 00C57 D7               D=C     A       D(A) := D0 (cf plus bas)
1927 00C59 8F0000           GOSBVL  =CSLC5
1928 00C60 D6               C=A     A
1929 00C62 10A              R2=C
1930 00C65
1931 00C65 8F0000           GOSBVL  =ASRW5
1932 00C6C 11B              C=R3
1933 00C6F 8F0000           GOSBVL  =CSLC5
1934 00C76 D6               C=A     A
1935 00C78 10B              R3=C
1936 00C7B          •
1937 00C7B          • Mise en place des paramètres de query
1938 00C7B          •
1939 00C7B AF2              C=0     W
1940 00C7E DB               C=D     A       C(W) := ^ début occurrence
1941 00C80 142              A=DAT0  A       A(A) := ^ OUTBS
1942 00C83 130              D0=A            D0 := ^ début ligne
1943 00C86 E2               C=C-A   A       C(A) := delta en nibs + long LIF
1944 00C88 81E              CSRB
1945 00C8B CE               C=C-1   A       C(A) := no colonne (1..max)
1946 00C8D 114              A=R4
1947 00C90 8F0000           GOSBVL  =ASRW5  A(A) := no ligne
1948 00C97          •
1949 00C97          • A(A) := no ligne
1950 00C97          • C(A) := no colonne
1951 00C97          • D0 := ^ début ligne (longueur LIF)
1952 00C97          •
1953 00C97 8E0000           GOSUBL  =query
1954 00C9D A87              D=C     P       D(0) := code de retour de query
1955 00CA0          •
1956 00CA0          • Remettre les registres dans l'ordre (restaurer R1)
1957 00CA0          •
1958 00CA0 11B              C=R3
1959 00CA3 DA               A=C     A       A(A) := MAx...
1960 00CA5 8F0000           GOSBVL  =CSRC5
1961 00CAC 10B              R3=C
1962 00CAF 8F0000           GOSBVL  =ASLW5
1963 00CB6
```

```
Page 041          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

1964 00CB6 11A              C=R2
1965 00CB9 DA               A=C     A       A(A) := D0
1966 00CBB 8F00000          GOSBVL  =CSRC5
1967 00CC2 10A              R2=C
1968 00CC5 101              R1=A
1969 00CC8          •
1970 00CC8          • Voyons maintenant ce que l'utilisateur a demandé.
1971 00CC8          •
1972 00CC8 A8B              C=D     P       C(0) := code de retour de query
1973 00CCB 8F00000          GOSBVL  =TBLJMC
1974 00CD2 900              REL(3)  RPL700  exit
1975 00CD5 D20              REL(3)  RPL800  yes
1976 00CD8 F10              REL(3)  RPL710  no
1977 00CDB
1978 00CDB          RPL700
1979 00CDB          •
1980 00CDB          • Code de retour = 0 ou 1. C'est exactement la valeur
1981 00CDB          • contenue dans R3(10)
1982 00CDB          •
1983 00CDB 11B              C=R3
1984 00CDE 80FA             CPEX    10      P := code de retour, C(10) := 0
1985 00CE2 80F0             CPEX    0       P := ???, C(0) := code de retour
1986 00CE6 20               P=      0       P := 0
1987 00CE8 90A              ?C=0    P
1988 00CEB A0               GOYES   RPL705  on peut sortir tout de suite
1989 00CED 06               RSTK=C          RSTK := code de retour
1990 00CEF 7DF0             GOSUB   undo    on remet les choses en place
1991 00CF3 07               C=RSTK          RSTK := code de retour
1992 00CF5 03       RPL705 RTNCC
1993 00CF7
1994 00CF7          •
1995 00CF7          • mode interactif et réponse = [N]
1996 00CF7          • Il faut donc remettre les choses en place.
1997 00CF7          •
1998 00CF7 75F0     RPL710 GOSUB   undo    voilà. Les choses sont remises...
1999 00CFB          •
2000 00CFB          • Valeur à ajouter à D0 : +1
2001 00CFB          •
2002 00CFB D2               C=0     A
2003 00CFD E6               C=C+1   A
2004 00CFF 591              GONC    RPL900  B.E.T.
2005 00D02
2006 00D02          •
2007 00D02          • Deux cas :
2008 00D02          •   - mode interactif et réponse = [Y]
2009 00D02          •   - mode non interactif (réponse implicite = [Y])
2010 00D02          • Il y a donc eu modification. R3(10) := 1
2011 00D02          •
2012 00D02 11B      RPL800 C=R3
2013 00D05 2A               P=      10
```

```
Page 042          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

2014 00D07 301              LC(1)   1
2015 00D0A 20               P=      0
2016 00D0C 10B              R3=C            R3(10) := 1
2017 00D0F          •
2018 00D0F          • Mettre dans C(A) le nombre de caractères à ajouter
2019 00D0F          • à D0 pour continuer d'explorer la ligne.
2020 00D0F          •
2021 00D0F 119              C=R1            nb de caractères à ajouter à D0
2022 00D12 8F00000          GOSBVL  =CSRC5  C(A) := f(n) (en caractères)
2023 00D19          •
2024 00D19          • C(A) := taille du bloc que l'on vient de passer c'est à
2025 00D19          • dire la valeur de l'incrément de D0 pour la recherche
2026 00D19          • suivante.
2027 00D19          •
2028 00D19
2029 00D19          RPL900
2030 00D19          •
2031 00D19          • C(A) = taille du bloc que l'on vient de passer c'est à
2032 00D19          • dire la valeur de l'incrément de D0 pour la recherche
2033 00D19          • suivante. Attention : la taille du bloc est en
2034 00D19          • caractères !
2035 00D19          •
2036 00D19          • Rechercher maintenant une nouvelle occurrence
2037 00D19          •
2038 00D19 112              A=R2            A(A) := D0
2039 00D1C CA               A=A+C   A
2040 00D1E CA               A=A+C   A       A(A) := nouveau D0
2041 00D20 130              D0=A            D0 := ^ chaine à chercher
2042 00D23
2043 00D23          •
2044 00D23          • Vérifier si la nouvelle position de D0 est plausible
2045 00D23          • c'est à dire pas derrière AVMEMS)
2046 00D23          • Assertions :
2047 00D23          • A(A) = D0 = nouvelle position de début de recherche
2048 00D23          • Stocker dans R4(15-11) la valeur actuelle de D0 pour
2049 00D23          • comparaison après POSADR, et ceci afin de détecter
2050 00D23          • deux "null match" de suite.
2051 00D23          •
2052 00D23 11C              C=R4
2053 00D26 8F00000          GOSBVL  =CSLC5
2054 00D2D D6               C=A     A
2055 00D2F 8F00000          GOSBVL  =CSRC5
2056 00D36 10C              R4=C
2057 00D39          RPL920
2058 00D39          •
2059 00D39          • Est-on arrivé au bout de la chaine ?
2060 00D39          •
2061 00D39 1F00000          D1=(5)  =AVMEMS
2062 00D40 147              C=DAT1  A       C(A) := AVMEMS
2063 00D43 8B6              ?A>C    A
```

```
Page 043          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

2064 00D46 F5               GOYES   RPL999  oui : Fini !
2065 00D48          •
2066 00D48          • Cas standard : il reste assez de place pour une
2067 00D48          • nouvelle recherche.
2068 00D48          •
2069 00D48 1C4              D1=D1-  5       D1=(5)  =OUTBS
2070 00D4B 147              C=DAT1  A
2071 00D4E 137              CD1EX           C(A) := ^ OUTBS, D1 := OUTBS
2072 00D51 173              D1=D1+  4
2073 00D54 137              CD1EX           A(A) := OUTBS + 4 ; D1=(5) OUTBS
2074 00D57          •
2075 00D57          • 88/10/15 : correction du bug :
2076 00D57          •   GENRPLC$(A$,"^","")
2077 00D57          •   Le vrai début doit être trafiqué, car on ne doit
2078 00D57          •   plus trouver de match en début de chaine. Avant,
2079 00D57          •   on trouvait un match à chaque tour sur l'exemple
2080 00D57          •   ci-dessus.
2081 00D57          •
2082 00D57 CE               C=C-1   A       C'est de l'escroquerie !
2083 00D59 109              R1=C            R1 := ^ vrai début de la chaine
2084 00D5C
2085 00D5C AF2              C=0     W
2086 00D5F 174              D1=D1+  5       D1=(5) =AVMEMS
2087 00D62 147              C=DAT1  A
2088 00D65 E2               C=C-A   A
2089 00D67 81E              CSRB            C(A) := longueur de ce qu'il reste
2090 00D6A DA               A=C     A       A(A) := longueur en caractères
2091 00D6C
2092 00D6C 8E2C7F           GOSUBL  =POSADR
2093 00D72          •
2094 00D72          • Si pas d'occurrence, sortir. La Cy n'a pas été modifiée.
2095 00D72          •
2096 00D72 523              GONC    RPL999  Pas d'occurrence : fini
2097 00D75          •
2098 00D75          • C(A) = LEN (occurrence trouvée)
2099 00D75          • D0 = position de l'occurrence trouvée
2100 00D75          •
2101 00D75 8AE              ?C#0    A
2102 00D78 F1               GOYES   rpl100  ok, on peut remplacer
2103 00D7A 136              CD0EX           D0 := LEN(occurrence), C(A) := D0
2104 00D7D          •
2105 00D7D          • Restauration de l'ancien D0 qui était dans R4(15-11).
2106 00D7D          •
2107 00D7D 114              A=R4
2108 00D80 810              ASLC
2109 00D83 810              ASLC
2110 00D86 810              ASLC
2111 00D89 810              ASLC
2112 00D8C 810              ASLC            A(A) := ancien D0
2113 00D8F 8A2              ?A=C    A
```

```
Page 044          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

2114 00D92 90               GOYES   RPL950  stop ! match nul. 1 partout...
2115 00D94 136              CD0EX           on remet les choses en place
2116 00D97 63AC     rpl100 GOTO    RPL100  et on y va pour le remplacement
2117 00D9B
2118 00D9B          •
2119 00D9B          • Cas spécial : on tombe sur une occurrence de longueur
2120 00D9B          • nulle derrière une occurrence déjà trouvée. On
2121 00D9B          • l'ignore donc.
2122 00D9B          •
2123 00D9B E4       RPL950 A=A+1    A
2124 00D9D E4               A=A+1   A
2125 00D9F 130              D0=A            D0 := A(A) := nouveau D0
2126 00DA2 569              GONC    RPL920  B.E.T.
2127 00DA5
2128 00DA5          RPL999
2129 00DA5          •
2130 00DA5          • La valeur de retour est 2 ou 3. C'est R3(10) + 2
2131 00DA5          •
2132 00DA5 11B              C=R3
2133 00DA8 80FA             CPEX    10      P := code de retour, C(10) := 0
2134 00DAC 80F0             CPEX    0       P := ???, C(0) := code de retour
2135 00DB0 20               P=      0       P := 0
2136 00DB2 B06              C=C+1   P
2137 00DB5 B06              C=C+1   P
2138 00DB8 03               RTNCC           Cy := 0 ; C(0) := 2 ou 3
2139 00DBA
2140 00DBA          ********************************************************
2141 00DBA          • rplc
2142 00DBA          •
2143 00DBA          • But: copie le texte qui a été sauvegardé derrière la
2144 00DBA          •   chaîne normale à l'endroit pointé par D0.
2145 00DBA          • Entree:
2146 00DBA          •   - D0 = destination address
2147 00DBA          •   - R1(4-0) = longueur du motif original (en octets)
2148 00DBA          •   - R2(9-5) = source address
2149 00DBA          • Sortie:
2150 00DBA          •   - D0 est réactualisé
2151 00DBA          • Abime: A, B(A), C(A), D(A), D0
2152 00DBA          • Appelle: MOVE•M, ASRW5, CSRC5, CSLC5
2153 00DBA          • Niveaux: 2 (MOVE•M)
2154 00DBA          • Note: D1 est sauvegarde lors de l'appel
2155 00DBA          • Historique:
2156 00DBA          •   88/10/08: PD/JT conception & codage
2157 00DBA          •   88/10/09: PD/JT débogage
2158 00DBA          ********************************************************
2159 00DBA
2160 00DBA 137      rplc   CD1EX
2161 00DBD 8F00000          GOSBVL  =CSLC5  C(9-5) := D1
2162 00DC4          •
2163 00DC4          • Pour MOVE•M
```

```
Page 045          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

2164 00DC4          • A(A) = source address
2165 00DC4          • B(A) = length in nibs
2166 00DC4          • C(A) = destination address
2167 00DC4          •
2168 00DC4 111              A=R1
2169 00DC7 D8               B=A     A
2170 00DC9 C5               B=B+B   A       B(A) := length in nibs
2171 00DCB 112              A=R2
2172 00DCE 8F00000          GOSBVL  =ASRW5  A(A) := source address
2173 00DD5 136              CD=EX           C(A) := destination address
2174 00DD8 8F00000          GOSBVL  =MOVE•M
2175 00DDF C9               C=C+B   A       C(A) := destination + length
2176 00DE1 134              D0=C            D0 est réactualisé
2177 00DE4 8F00000          GOSBVL  =GSRC5  restauration de D1
2178 00DEB 135              D1=C
2179 00DEE 01               RTN
2180 00DF0
2181 00DF0          ****************************************************************
2182 00DF0          • undo
2183 00DF0          •
2184 00DF0          • But: remet le texte comme il était avant la dernière
2185 00DF0          •   modification.
2186 00DF0          • Entrees:
2187 00DF0          •   - R1(4-0) = longueur du motif original (en octets)
2188 00DF0          •   - R1(9-5) = longueur du motif remplacé (en octets)
2189 00DF0          •   - R2(4-0) = destination
2190 00DF0          •   - R2(9-5) = source
2191 00DF0          • Sortie:
2192 00DF0          •   - AVMEMS est réactualisé
2193 00DF0          • Abime: A, B(A), C, D(A), D0, D1
2194 00DF0          • Appelle: MOVE•M, GSRC5, ASRW5
2195 00DF0          • Niveaux: 2 (MOVE•M)
2196 00DF0          • Historique:
2197 00DF0          •   88/10/16: PD/JT conception & codage
2198 00DF0          ****************************************************************
2199 00DF0
2200 00DF0 119      undo    C=R1
2201 00DF3 D7               D=C     A       D(A) := n
2202 00DF5 C7               D=D+D   A       D(A) := 2•n
2203 00DF7 8F00000          GOSBVL  =GSRC5  C(A) := f(n)
2204 00DFE C6               C=C+C   A       C(A) := 2•f(n)
2205 00E00 112              A=R2            A(A) := D0
2206 00E03 CA               A=A+C   A       A(A) := D0 + 2•f(n) (source addr)
2207 00E06
2208 00E06 1B0000           D0=(5)  =AVMEMS
2209 00E0C 146              C=DAT0  A       C(A) := AVMEMS
2210 00E0F E2               C=C-A   A       C(A) := AVMEMS - (D0 + 2•f(n))
2211 00E11 D5               B=C     A       B(A) := length of block
2212 00E13
2213 00E13 11A              C=R2            C(A) := D0
```

```
Page 046          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   Remplacement des lignes

2214 00E16 CB               C=C+D   A       C(A) := D0 + 2•n
2215 00E18          •
2216 00E18          • A(A) = source address = D0 + 2•f(n)
2217 00E18          • B(A) = length of bloc in nibs = AVMEMS - (D0 + 2•f(n))
2218 00E18          • C(A) = dest address = D0 + 2•n
2219 00E18          •
2220 00E18 8F00000          GOSBVL  =MOVE•M
2221 00E1F
2222 00E1F 119              C=R1
2223 00E22 D5               B=C     A       B(A) := n
2224 00E24 C5               B=B+B   A       B(A) := 2•n
2225 00E26
2226 00E26 112              A=R2
2227 00E29 DE               C=A     A       C(A) := D0
2228 00E2B
2229 00E2B 8F0000           GOSBVL  =ASRW5  A(A) := MAX...
2230 00E32          •
2231 00E32          • A(A) = source address = MAX...
2232 00E32          • B(A) = length of bloc in nibs = 2•n
2233 00E32          • C(A) = dest address = D0
2234 00E32          •
2235 00E32 8F0000           GOSBVL  =MOVE•M
2236 00E39
2237 00E39          •
2238 00E39          • AVMEMS += 2•(n - f(n))
2239 00E39          •
2240 00E39 1B0000           D0=(5)  =AVMEMS
2241 00E40 119              C=R1
2242 00E43 DA               A=C     A       A(A) := n
2243 00E45 8F0000           GOSBVL  =GSRC5  C(A) := f(n)
2244 00E4C EA               A=A-C   A       A(A) := n - f(n)
2245 00E4E 146              C=DAT0  A       C(A) := AVMEMS
2246 00E51 C2               C=C+A   A
2247 00E53 C2               C=C+A   A
2248 00E55 144              DAT0=C  A       AVMEMS += 2•(n - f(n))
2249 00E58 01               RTN
2250 00E5A
2251 00E5A                  END
```

```
Page 047          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   •••• SYMBOL TABLE ••••

=ASLW5      Extrn Ukn       -  1616 1671 1962
=ASRW5      Extrn Ukn       -  1931 1947 2172 2229
=AVMEME     Extrn Ukn       -  0615 0617
=AVMEMS     Extrn Ukn       -  0617 1675 1741 1885 1906 1908 2061 2208 2240
 ANY        00002 Abs  0015 -  0196 0517 0565
 ANYS       00001 Abs  0026 -  0193 0514 0559 1315
 BOL        00006 Abs  0019 -  0213 0337 0540 0850
 BOLS       00001 Abs  0024 -  0210 0537
=CALCab     00924 Rel  1441 -
=COMP71     002AB Rel  0438 -
=COMPUX     00000 Rel  0112 -
=GSLC3      Extrn Ukn       -  1391
=GSLC5      Extrn Ukn       -  0826 1603 1927 1933 2053 2161
=GSLC8      Extrn Ukn       -  1167 1198
=GSRC3      Extrn Ukn       -  0812 1388
=GSRC5      Extrn Ukn       -  0822 0840 0906 0974 1443 1709 1727 1746 1773
                            +  1882 1960 1966 2022 2055 2177 2203 2243
=GSRC8      Extrn Ukn       -  1177
 CAL610     009DA Rel  1524 -  1518
 CAL615     009E0 Rel  1526 -  1521 1536
 CAL620     009FA Rel  1537 -  1532
 CAL630     009B5 Rel  1509 -  1506
 CALC10     0094C Rel  1458 -  1474 1476 1480
 CALC20     00975 Rel  1475 -  1466
 CALC25     0097B Rel  1478 -  1462
 CALC30     0097E Rel  1479 -  1469
 CALC50     00988 Rel  1487 -  1454 1515 1529 1538 1541
 CALC60     009BB Rel  1512 -  1500
 CALC61     009C6 Rel  1517 -  1502
 CALC62     009E9 Rel  1531 -  1504
 CALC63     009AA Rel  1505 -
 CALC70     00A00 Rel  1540 -  1508 1510 1513 1523 1525 1535
 CALC99     00A06 Rel  1547 -  1482 1488
 CCL        00003 Abs  0015 -  0244 1200
 CCLS       00101 Abs  0027 -  0239 0261 0294 1316
 CHAR       00000 Abs  0013 -  0734
 CHARS      00003 Abs  0029 -  0731 1309 1312
 CLOS       00007 Abs  0020 -  0340 0361 0562 1067
 CLOSS      00001 Abs  0030 -  0559 1078 1320
 Cmpend     002E0 Rel  0468 -  0458
=ENDFOS     0091A Rel  1410 -  0372 0741
 END        00005 Abs  0018 -  0165 1064
 EOL        00001 Abs  0014 -  0231 0555
 EOLS       00001 Abs  0025 -  0228 0552 1314
 Errmem     00A1C Rel  1608 -  1657 1662
=FINDA      Extrn Ukn       -  0133 0494
 FiLeNd     00E5A Rel  2251 -
=I/OALL     Extrn Ukn       -  0635
=I/OCON     Extrn Ukn       -  0171
=I/ODAL     Extrn Ukn       -  1411
```

```
Page 048          Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4   •••• SYMBOL TABLE ••••

=IOFND0     Extrn Ukn       -  1381
=IOFSCR     Extrn Ukn       -  0609
 InvPat     0008D Rel  0190 -  0184
 Invpat     00121 Rel  0236 -  0243 0267 0275
=LEEWAY     Extrn Ukn       -  0621
=MEMCKL     Extrn Ukn       -  1712
=MODE71     Extrn Ukn       -  1444 1768
=MOVE•M     Extrn Ukn       -  1728 1759 2174 2220 2235
=MOVED2     Extrn Ukn       -  0357
=MPY        Extrn Ukn       -  1653
 NCCL       00004 Abs  0017 -  0249
 NCCLS      00101 Abs  0028 -  1317
 NoRoom     003CC Rel  0544 -  0516 0539 0554 0561
 Nxroom     00AE5 Rel  0218 -  0195 0212 0230 0241
=OUTBS      Extrn Ukn       -  1308
=POSAOR     00534 Rel  0811 -  2032
 Pos20      00BBB Rel  1004 -  1001
 Poschr     00592 Rel  0855 -  0844
=QUERY      Extrn Ukn       -  1892
=REPLIN     00A24 Rel  1611 -
 RP3210     00BC0 Rel  1856 -  1850
 RP3215     00BC6 Rel  1858 -  1853
 RP3220     00BDD Rel  1867 -  1862
 RP3230     00B9B Rel  1841 -  1838
 RPL100     00A3B Rel  1633 -  0116
 RPL120     00A94 Rel  1683 -  1681
 RPL200     00BD7 Rel  1781 -
 RPL210     00BD7 Rel  1789 -  1811
 RPL220     00B4A Rel  1801 -  1796
 RPL240     00B50 Rel  1804 -  1793
 RPL270     00B56 Rel  1807 -  1799
 RPL280     00B5E Rel  1809 -  1800 1805
 RPL290     00BC7 Rel  1784 -  1802 1808
 RPL300     00BEB Rel  1818 -  1777
 RPL310     00BEE Rel  1825 -
 RPL320     00BA1 Rel  1844 -  1831
 RPL321     00BAC Rel  1849 -  1833
 RPL322     00BCC Rel  1861 -  1835
 RPL323     00B90 Rel  1837 -
 RPL370     00BE3 Rel  1870 -  1859 1865
 RPL380     00BEB Rel  1872 -  1840 184C 1845 1855 1857 1866
 RPL390     00BEE Rel  1829 -  1847 1868 1871
 RPL500     00BF4 Rel  1876 -  1813 1826
 RPL600     00C27 Rel  1905 -  1895
 RPL700     00CDB Rel  1978 -  1974
 RPL705     00CF5 Rel  1992 -  1988
 RPL710     00CF7 Rel  1998 -  1976
 RPL800     00D02 Rel  2012 -  1896 1975
 RPL900     00D19 Rel  2029 -  2004
 RPL920     00D39 Rel  2057 -  2126
```

```
Page 049            Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4      **** SYMBOL TABLE ****

 RPL950         00D9B Rel  2123 -  2114
 RPL999         00DA5 Rel  2128 -  2064 2096
 RtnCC          005EB Rel  0898 -  0944
 Rtnoc          006B9 Rel  1002 -  0987
=SWPBYT         Extrn Ukn       -  1915
 SANY           0002E Abs  0050 -  0495
 SBOL           0005E Abs  0048 -  0497
 SEOL           00024 Abs  0049 -  0499
 SREP           00026 Abs  0052 -  1519 1533 1851 1863
 SREPT          00040 Abs  0051 -  0501
 STKSIZ         00118 Abs  0065 -  0525 0630 0673 0823
 STOGL          0005C Abs  0053 -  0459 1496 1828
 System         0004B Rel  0156 -  0172
=TBLJMC         Extrn Ukn       -  0463 0470 1973
=UPDTR0         008F1 Rel  1380 -
=WIPOUT         Extrn Ukn       -  0262
 XANY           0002E Abs  0037 -  0136
 XBCCL          0005B Abs  0038 -  0142
 XBOL           0005E Abs  0035 -  0138
 XCLOS          0002A Abs  0042 -  0144
 XECCL          0005D Abs  0039 -  0268
 XEOL           00024 Abs  0036 -  0140
 XESC           0005C Abs  0044 -  0134 0271 0304 1464 1794
 XICCL          0002D Abs  0041 -  0280
 XNCCL          0005E Abs  0040 -
 XREF           00026 Abs  0043 -  1467 1797
 acl010         00723 Rel  1081 -  1084
 acl100         00733 Rel  1091 -  1082
 acl150         00737 Rel  1093 -  1099
 aclret         0073B Rel  1094 -  1170
 addchr         004F7 Rel  0731 -  0149 0186 0207 0225 0489 0504 0508 0533 0549
 amaclo         0071C Rel  1078 -  1069
 amarec         006F9 Rel  1063 -  1073 1200
 amatch         006BF Rel  1042 -  0883 0960 1000
=coeffa         Extrn Ukn       -  1547
=coeffb         Extrn Ukn       -  1650
 c71-10         002B8 Rel  0457 -  0478 0483 0491 0506 0512 0520 0535 0543 0551
                                +  0558 0568
 c71nor         002E4 Rel  0469 -  0461
 c7ANY          00365 Rel  0514 -  0436
 c7BL10         003E0 Rel  0536 -  0529
 c7BOL          0037E Rel  0521 -  0430
 c7EL10         003EB Rel  0552 -  0546
 c7EOL          003D0 Rel  0545 -  0500
 c7REPT         00404 Rel  0559 -  0502
 c7b0           002FA Rel  0476 -  0464
 c7b1           00304 Rel  0479 -  0465
 c7b2           0030B Rel  0481 -  0466
 c7b3           00315 Rel  0484 -  0467
 c7b30          00313 Rel  0485 -  0430
```

```
Page 050            Editeur, chaines génériques <xgen.as>
AREUH ASS. V2.4      **** SYMBOL TABLE ****

 c7n0           0031B Rel  0489 -  0471
 c7n1           00326 Rel  0492 -  0472
 c7n2           0032C Rel  0494 -  0473
 c7n3           00354 Rel  0508 -  0474
 calc99         00984 Rel  1482 -  1459
 call           00705 Rel  1176 -  1093
 cmpend         00053 Rel  0159 -  0132 0468
 cuANY          00091 Rel  0192 -  0137
 cuBL10         000CC Rel  0210 -  0204
 cuBOL          000AE Rel  0201 -  0139
 cuCCL          00125 Rel  0233 -  0143
 cuCL10         00153 Rel  0252 -  0243
 cuCL50         0017F Rel  0266 -  0231 0325
 cuCL60         001A3 Rel  0276 -  0273
 cuCL70         001B9 Rel  0284 -  0279
 cuCL80         001E8 Rel  0301 -  0282
 cuCL82         00209 Rel  0309 -  0306
 cuCL84         00226 Rel  0319 -  0322
 cuCL90         001D1 Rel  0293 -  0270
 cuCLOS         00241 Rel  0329 -  0145
 cuEL10         00102 Rel  0228 -  0222
 cuEOL          000E9 Rel  0220 -  0141
 cuESC          00075 Rel  0183 -  0135
 cux-10         0000C Rel  0131 -  0151 0188 0199 0209 0216 0227 0234 0297 0370
=eMEM           Extrn Ukn       -  0688 1608
=eSYCEP         Extrn Ukn       -  0570
 errmem         004E4 Rel  0688 -  0623 0632 0636
 errrtn         00A22 Rel  1609 -
=getchr         Extrn Ukn       -  0131 0183 0251 0266 0274 0301 0302 0307 0457
                                +  1458 1487 1789 1825
 getb10         0047B Rel  0630 -  0626
 getbuf         0042F Rel  0599 -  0112 0433
 gettab         00520 Rel  0762 -  0286 0315 1275
=id             Extrn Ukn       -  0373 0742
 irvPat         001E4 Rel  0299 -  0303 0308 0314 0333
 invpat         0029F Rel  0372 -  0130 0236 0299 0339 0342
=lookat         Extrn Ukn       -  0221 0243 0278 0545 1081 1266 1272
 marque         004EC Rel  0706 -  0148 0185 0192 0202 0220 0238
 noRoom         0023D Rel  0327 -  0330
 noroοM         0004F Rel  0157 -  0164
 noroom         00514 Rel  0741 -  0157 0218 0327 0544 0733
 nxtpat         00896 Rel  1306 -  1072 1091
 omANY          00859 Rel  1268 -  1250
 omCC10         0088B Rel  1284 -  1282
 omCCL          00862 Rel  1271 -  1252
 omCHAR         00814 Rel  1230 -
 omEND          00851 Rel  1259 -  1256
 omEOL          00853 Rel  1266 -  1248
 omNCCL         00862 Rel  1272 -  1254
 oma10          00831 Rel  1247 -  1225
```

```
Page 051            Editeur, chaines génériques <xgem.as>
AREUH ASS. V2.4      **** SYMBOL TABLE ****

 omatch         0080A Rel  1223 -  1070 1083
 pos10          00596 Rel  0866 -  0852 0897
 pos20          005ED Rel  0900 -  0884 1004
 pos30          005F7 Rel  0911 -  0977
 posbol         00637 Rel  0983 -  0853
 posc10         00627 Rel  0943 -  0949
 posc20         00675 Rel  0973 -  0961
 poschr         00609 Rel  0926 -  0855 0971
 ps05           008A4 Rel  1312 -  1308
 ps10           008C0 Rel  1321 -  1312
=query          Extrn Ukn       -  1953
=rhcteg         008DE Rel  1351 -  1268 1286
 ret010         00757 Rel  1112 -  1109
 ret020         0078A Rel  1142 -  1134
 ret050         007A9 Rel  1160 -  1138
 ret090         007BE Rel  1172 -  1123 1169
 return         0074E Rel  1108 -  1066 1071 1094 1103
 rpl100         00D97 Rel  2116 -  2102
 rpl390         00BE7 Rel  1871 -  1874
 rpl500         00B67 Rel  1813 -  1790
 rplc           00DBA Rel  2160 -  1807 1870
 rtncc          0084F Rel  1257 -  1233 1240 1269 1273 1285
 sysTem         008ED Rel  1378 -  1382
 system         00427 Rel  0570 -  0156 0610 1378
=teICMP         Extrn Ukn       -  0742
=teIPAT         Extrn Ukn       -  0373
 undo           00DF0 Rel  2200 -  1990 1998


Source  : xgem.as

Object  : obj/xgem.ao

Listing : list/xgem.al

Date    : Sat Aug 12 17:41:36 1989

Errors  : 000


Areuh Assembler/Linker V2.4. (c) P. David & J. Taillandier 1986 Paris, France
```

Suite au prochain numéro...

Le Journal JPC est le bulletin de liaison entre les membres de l'Association "PPC Paris", régie par la loi de 1901. Le Club est éditeur de JPC, et son siège social est au 56, rue Jean-Jacques Rousseau, 75001 Paris.

La maquette de ce numéro a été préparée et réalisée par Jacques Belin et Laurent Chouraki grâce à un système comprenant un HP71B, un lecteur de disquettes HP9114A, un HP9807A, deux HP9154 et une imprimante LaserJet Séries II.

Les dessins sont de Jean-Jacques Dhénin et Paul Courbis.

Directeur de la publication : Jean Reibel
Numéro ISSN : 0762 - 381X

Veuillez adresser toute correspondance à :
PPC Paris, BP 604, 75028 Paris Cedex 01.

Imprimé par Copy-Express, 42 86 91 94.