

```

0001 00000 TITLE Editeur, execution des commandes <xcmd.as>
0002 00000
0003 00000
0004 00000
0005 00000 * JPC:003
0006 00000 * 89/08/07: PD Correction des valeurs par défaut de J
0007 00000
0008 00000
0009 00000 .....
0010 00000 * Variables locales
0011 00000 .....
0012 00000
0013 00000 * "tmp" démarre juste après les variables globales de XEDIT.
0014 00000 * Attention de ne pas avoir une zone de 1 quartet comme
0015 00000 * dernier endroit de cette zone, car cela génèrait les
0016 00000 * D0=(2)...
0017 00000
0018 00000 * tmp EQU =XEDTMP début de la zone temporaire
0019 00000
0020 00000
0021 00000 * "stat" démarre en STMIR0. Cette zone est à utiliser avec
0022 00000 * beaucoup de précautions car beaucoup de routines
0023 00000 * l'utilisent en totalité ou en partie.
0024 00000
0025 00000 * stat EQU =STMIR0 début de la zone TRES temporaire
0026 00000
0027 00000
0028 00000 * L / P (disponible : 20 quartets)
0029 00000
0030 00000 * cour EQU 00*tmp 5 q : no de la courante
0031 00000 * remain EQU 05*tmp 5 q : reste encore ... lignes
0032 00000 * optn EQU 10*tmp 1 q : option "N" active
0033 00000
0034 00000
0035 00000 * I / T (disponible : 20 quartets)
0036 00000
0037 00000 * insert EQU 00*tmp 1 q : 0 si "I", 1 si "I"
0038 00000 * stradr EQU 01*tmp 5 q : adresse de la chaîne
0039 00000 * strlen EQU 06*tmp 5 q : longueur de la chaîne
0040 00000 * curlin EQU 11*tmp 5 q : adresse de la ligne
0041 00000 * lastly EQU 16*tmp 1 q : 1 si [V] ou [g][V]
0042 00000
0043 00000
0044 00000 * S (disponible : 20 quartets)
0045 00000
0046 00000
0047 00000 * ldeb EQU 00*tmp 5 q : no ligne courante
0048 00000 * lfin EQU 05*tmp 5 q : no ligne de fin
0049 00000 * debadr EQU 10*tmp 5 q : adresse ligne courante
0050 00000 * offset EQU 15*tmp 5 q : dernière occurrence

```

```

0100 00000 .....
0101 00000 * svRIR2
0102 00000
0103 00000
0104 00000 * But: Sauver R1 et R2 dans ldeb et lfin. UEile dans les
0105 00000 * commandes S, R, D et C/M
0106 00000
0107 00000 * Entrée:
0108 00000 * - R1(A) = numéro de la ligne de début
0109 00000 * - R2(A) = numéro de la ligne de fin
0110 00000 * Sortie:
0111 00000 * - ldeb = numéro de la ligne de début
0112 00000 * - lfin = numéro de la ligne de fin
0113 00000
0114 00000 * Abime: C, D0
0115 00000 * Appelle: -
0116 00000 * Niveaux: 0
0117 00000 * Historique:
0118 00000 * 88/11/13: PD/JT séparation de S, R, D et C/M
0119 00000 .....
0120 00000 svRIR2 D0=(5) ldeb
0121 00000 C=R1
0122 00000 DAT0=C A ldeb := R1
0123 00000 D0=D0+ (lfin)-(ldeb)
0124 00000 C=R2
0125 00000 DAT0=C A lfin := R2
0126 00000 RTN
0127 00000
0128 00000 .....
0129 00000 * getdbf
0130 00000
0131 00000 * But: Dans les commandes D et C/M internes, chercher les
0132 00000 * adresses de début et de fin à partir de celles stockées
0133 00000 * dans ldeb et lfin et les placer dans addeb et adfin, en
0134 00000 * calculant la différence dans findeb.
0135 00000
0136 00000 * Entrée:
0137 00000 * - ldeb = numéro de la ligne de début
0138 00000 * - lfin = numéro de la ligne de fin
0139 00000 * Sortie:
0140 00000 * - addeb = adresse de la ligne de début
0141 00000 * - findeb = adfin - addeb
0142 00000 * - nblig = lfin - ldeb + 1
0143 00000 * Abime: A-D, R0-R3, D0, D1, THP5
0144 00000 * Appelle: seek
0145 00000 * Niveaux: 5
0146 00000 * Historique:
0147 00000 * 88/11/12: PD/JT séparation de D
0148 00000 * 88/11/13: PD/JT ajout du calcul de nblig
0149 00000 .....
0150 00000

```

```

0051 00000
0052 00000 * R (disponible : 10 quartets)
0053 00000
0054 00000 * ldeb EQU ... 5 q : no ligne courante
0055 00000 * lfin EQU ... 5 q : no ligne de fin
0056 00000 * dern EQU 00*stat 5 q : dernière ligne ok
0057 00000 * remp EQU 05*stat 10 q : chaîne de remplac
0058 00000 * savlin EQU 15*stat 10 q : " ligne ds le fichier
0059 00000
0060 00000
0061 00000 * H (disponible : 20 quartets)
0062 00000
0063 00000 * curcmd EQU 00*tmp 1 q : numéro de la commande courante
0064 00000
0065 00000
0066 00000 * J (disponible : 20 quartets)
0067 00000
0068 00000 * cour EQU ... 5 q : adresse ligne courante
0069 00000 * remain EQU ... 5 q : encore ... lignes
0070 00000 * prem EQU 10*tmp 5 q : numéro de la première ligne
0071 00000 * jsave EQU 00*stat 15 q : sauvegarde pendant '?'
0072 00000
0073 00000
0074 00000 * D (disponible : 20 quartets)
0075 00000
0076 00000 * ldeb EQU 00*tmp 5 q : no ligne courante
0077 00000 * lfin EQU 05*tmp 5 q : no ligne de fin
0078 00000 * EXTFIL EQU 10*tmp 5 q : adresse fichier extérieur
0079 00000 * findeb EQU 15*tmp 5 q : adfin - addeb pendant MVFILE
0080 00000 * nblig EQU 00*stat 5 q : lfin - ldeb + 1
0081 00000 * addeb EQU 05*stat 5 q : adresse ldeb
0082 00000 * adfin EQU 10*stat 5 q : adresse lfin
0083 00000
0084 00000
0085 00000 * C / M (disponible : 20 quartets)
0086 00000
0087 00000
0088 00000 * ldeb EQU 00*tmp 5 q : no ligne courante
0089 00000 * lfin EQU 05*tmp 5 q : no ligne de fin
0090 00000 * EXTFIL EQU 10*tmp 5 q : adresse fichier extérieur
0091 00000 * findeb EQU 15*tmp 5 q : adfin - addeb pendant MVFILE
0092 00000 * nblig EQU 00*stat 5 q : lfin - ldeb + 1
0093 00000 * addeb EQU 05*stat 5 q : adresse ldeb
0094 00000 * adfin EQU 10*stat 5 q : adresse lfin
0095 00000 * copie EQU 15*stat 1 q : 1 si copie ou 0 si move
0096 00000 * curadr EQU 16*stat 5 q : adresse de la ligne courante
0097 00000
0098 00000 * Ceave EQU 15*tmp 2 q : sauvegarde temporaire

```

```

0150 00000 getdbf D0=(5) ldeb
0151 00000 C=DAT0 A (A) := numéro de ligne
0152 00000 GOSUBL =seek (A) := " début de la ligne
0153 00000 CD0EX (A) := " début de la ligne
0154 00000 D0=(5) addeb
0155 00000 DAT0=C A addeb := " début de ldeb
0156 00000
0157 00000 D0=(4) lfin
0158 00000 C=DAT0 A (A) := numéro de ligne
0159 00000 C=C+1 A (A) := " début de la ligne
0160 00000 GOSUBL =seek (A) := " début de la ligne
0161 00000 CD0EX (A) := " début de la ligne
0162 00000 D0=(5) adfin
0163 00000 DAT0=C A adfin := " fin de la zone
0164 00000
0165 00000
0166 00000 * On prépare la mise à jour du buffer qui aura lieu tout à
0167 00000 * la fin. MVFILE abimant addeb et adfin, on doit s'en
0168 00000 * détacher le plus vite possible. Seul "adfin - addeb" est
0169 00000 * nécessaire pour "dellin". On calcule donc cette valeur
0170 00000 * et on la met en lieu sûr.
0171 00000
0172 00000 D0=D0- 5 D0=(5) addeb
0173 00000 A=DAT0 A A(A) := addeb
0174 00000 C=C-A A (A) := adfin - addeb
0175 00000 D0=(4) findeb
0176 00000 DAT0=C A findeb := adfin - addeb
0177 00000
0178 00000
0179 00000 * Calculer le nombre de lignes traitées et le placer
0180 00000 * dans nblig.
0181 00000
0182 00000 D0=(4) ldeb
0183 00000 C=DAT0 A (A) := ldeb
0184 00000 D0=D0+ (lfin)-(ldeb)
0185 00000 A=DAT0 A A(A) := lfin
0186 00000 C=C-A A (A) := lfin - ldeb
0187 00000 C=C+1 A (A) := lfin - ldeb + 1
0188 00000 D0=(4) nblig
0189 00000 DAT0=C A
0190 00000
0191 00000 RTN
0192 00000
0193 00000 .....
0194 00000
0195 00000 * ck1prm
0196 00000
0197 00000 * But: vérifier la cohérence pour une syntaxe admettant 0 ou
0198 00000 * 1 paramètre.
0199 00000
0200 00000 * Entrée:

```

```

Page 005      Editeur, exécution des commandes <xcmd.as>
AREUH ASS. V2.4  Utilitaires

0200 00082      • - R0 = nombre de paramètres
0201 00082      • - R1 = ldebut éventuellement
0202 00082      • Sortie:
0203 00082      • - C(A) = R1 = ldebut ou courante, vérifiée
0204 00082      • Abime: A, C, D0
0205 00082      • Niveaux: 0
0206 00082      • Detail:
0207 00082      • selon nombre de paramètres
0208 00082      • 0 : renvoyer courante
0209 00082      • 1 : selon ldebut
0210 00082      • 2 : renvoyer 1
0211 00082      • 3 : dernière + 1 : renvoyer ldebut
0212 00082      • 4 : sinon : dernière + 1
0213 00082      • fin selon
0214 00082      • 2 : "Invalid Parm"
0215 00082      • fin selon
0216 00082      • Historique:
0217 00082      • 88/05/23: PD/JT conception & codage
0218 00082      .....
0219 00082      .....
0220 00082 110      ckipm A=R0      A(A) := nb de paramètres
0221 00082 CC      A=A-1 A
0222 00082 472      GOC ckip50 0 paramètres
0223 00082 CC      A=A-1 A
0224 00082 513      GONC Invrm 2 paramètres : "Invalid Parm"
0225 00082      .....
0226 00082 1800000      • 1 paramètre
0227 00082 142      D0=(5) =DERN
0228 00082 119      A=DAT0 A      A(A) := dernière
0229 00082 8AA      C=R1      C(A) := ldebut
0230 00082 90      J=C0 A
0231 00082 8BE      GOYES ckip10 ldebut = 0
0232 00082 00      J=C=A A
0233 00082 00      RTNYES      (ldebut := dernière
0234 00082 05      C=A A      C(A) := dernière
0235 00082 109      ckip10 C=C+1 A
0236 00082 01      S1=C      (ldebut := 1 ou (dernière + 1)
0237 00082 01      RTN
0238 00082      .....
0239 00082 1800000      • 0 paramètre
0240 00082 145      ckip50 D0=(5) =COUR
0241 00082 109      A=DAT0 A
0242 00082 01      J=C A      ldebut := courante
0243 00082 01      RTN
0244 00082 610A      Invrm 2077 Invrm
0245 00082      .....
0246 00082      .....
0247 00082      .....
0248 00082      .....
0249 00082      .....

```

```

Page 007      Editeur, exécution des commandes <xcmd.as>
AREUH ASS. V2.4  Utilitaires

0300 00085      .....
0301 00085 118      set.db C=R0
0302 00085 8AC      J=C0 A      aucun paramètre fourni ?
0303 00085 11      GOYES sdb010 alors "default to R1"
0304 00085 1800000      D0=(5) =COUR sinon "default to ."
0305 00085 142      A=DAT0 A      ldebut := courante
0306 00085 06      C=A A      lfin := courante
0307 00085 570      GONC sdb020
0308 00085 111      sdb010 A=R1      ldebut := ldebut
0309 00085 06      C=A A      lfin := ldebut
0310 00085 7320      sdb020 C0SUB default
0311 00085 500      RTNNC      ok, pas d'erreur
0312 00085 6E75      GOTO notfnd intervalle inexistant
0313 0010F      .....
0314 0010F      .....
0315 0010F      .....
0316 0010F      .....
0317 0010F      .....
0318 0010F      .....
0319 0010F      .....
0320 0010F      .....
0321 0010F      .....
0322 0010F      .....
0323 0010F      .....
0324 0010F      .....
0325 0010F      .....
0326 0010F      .....
0327 0010F      .....
0328 0010F      .....
0329 0010F      .....
0330 0010F      .....
0331 0010F      .....
0332 0010F      .....
0333 0010F      .....
0334 0010F      .....
0335 0010F      .....
0336 0010F 1800000      set.$ D0=(5) =COUR
0337 00116 142      A=DAT0 A
0338 00119 E4      A=A-1 A      +1
0339 00118 1900      D0=(2) =DERN
0340 0011F 146      C=DAT0 A
0341 00122 E6      C=C+1 A      $+1
0342 00124      .....
0343 00124 88A      7A=C A
0344 00127 40      GOYES default
0345 00129 DA      A=C A
0346 0012B      .....
0347 0012B      .....
0348 0012B      .....
0349 0012B      .....

```

```

Page 006      Editeur, exécution des commandes <xcmd.as>
AREUH ASS. V2.4  Utilitaires

0250 00082      • Entrée:
0251 00082      • - C(A) = nouvelle valeur de ligne courante
0252 00082      • Sorties:
0253 00082      • - COUR remis à jour
0254 00082      • Abime: A, C, D0
0255 00082      • Niveaux: 0
0256 00082      • Detail:
0257 00082      • selon 0/4
0258 00082      • 0 : courante := 1
0259 00082      • 1 : dernière + 1 : courante := C(A)
0260 00082      • 2 : sinon : courante := dernière + 1
0261 00082      • fin selon
0262 00082      • Historique:
0263 00082      • 88/05/23: PD/JT conception & codage
0264 00082      .....
0265 00082      .....
0266 00082 1800000      setcou D0=(5) =DERN
0267 00082 8AE      J=C0 A
0268 00082 70      GOYES sc10
0269 00082 06      C=C+1 A
0270 00082 5E0      GONC sc90      B.E.T., courante := 1
0271 00082 142      sc10 A=DAT0 A
0272 00082 E4      A=A-1 A      A(A) := dernière + 1
0273 00082 8BE      J=C=A A
0274 00082 40      GOYES sc90      ok, courante := C(A)
0275 00082 06      C=A A      C(A) := dernière + 1
0276 00082 1300      sc90 D0=(2) =COUR
0277 00082 144      DAT0=C A      courante := C(A)
0278 00082 01      RTN
0279 00082      .....
0280 00082      .....
0281 00082      .....
0282 00082      .....
0283 00082      .....
0284 00082      .....
0285 00082      .....
0286 00082      .....
0287 00082      .....
0288 00082      .....
0289 00082      .....
0290 00082      .....
0291 00082      .....
0292 00082      .....
0293 00082      .....
0294 00082      .....
0295 00082      .....
0296 00082      .....
0297 00082      .....
0298 00082      .....
0299 00082      .....

```

```

Page 008      Editeur, exécution des commandes <xcmd.as>
AREUH ASS. V2.4  Utilitaires

0350 0012B      .....
0351 0012B      .....
0352 0012B      .....
0353 0012B      .....
0354 0012B      .....
0355 0012B      .....
0356 0012B      .....
0357 0012B      .....
0358 0012B      .....
0359 0012B      .....
0360 0012B      .....
0361 0012B      .....
0362 0012B      .....
0363 0012B      .....
0364 0012B      .....
0365 0012B      .....
0366 0012B      .....
0367 0012B      .....
0368 0012B      .....
0369 0012B      .....
0370 0012B      .....
0371 0012B      .....
0372 0012B      .....
0373 0012B      .....
0374 0012B      .....
0375 0012B      .....
0376 0012B      .....
0377 0012B      .....
0378 0012B      .....
0379 0012B      .....
0380 0012B      .....
0381 0012B      .....
0382 0012B D8      default B=A      A      B(A) := ldebut par default
0383 0012D      .....
0384 0012D 110      A=R0      C(A) := nombre de paramètres
0385 00130 CC      A=A-1 A
0386 00132 4D0      GOC def110 nb paramètres = 0
0387 00135 CC      A=A-1 A
0388 00137 501      GONC def150 nb paramètres = 2
0389 0013A      .....
0390 0013A 10A      R2=C      R2 := lfin par default
0391 0013D 4A0      GOC def150      B.E.T.
0392 00140      .....
0393 00140 10A      R2=C      R2 := lfin par default
0394 00143 D4      A=B A      A(A) := ldebut par default
0395 00145 101      R1=A      R1 := ldebut par default
0396 00145      .....
0397 00145      .....
0398 00145      .....
0399 00145      .....

```

Page 009 Editeur, execution des commandes <xcmd.as>  
AREUH ASS. V2.4 Utilitaires

```

0400 00148      * si ldebut = lfin alors erreur
0401 00148      * lfin = min (lfin, dernière)
0402 00148      * si ldebut > dernière + 1 alors erreur
0403 00148      * si ldebut <= dernière alors INTERVALLE EXISTANT
0404 00148      * INTERVALLE INEXISTANT (ldebut = dernière + 1)
0405 00148      *
0406 00148 111      deft50 A=R1      ldebut
0407 00148 8A8      7A=0 A
0408 00148 D2      GOYES invprm "Invalid Parm"
0409 00150 11A      C=R2      lfin
0410 00153 8B6      7A=C A
0411 00156 52      GOYES invprm "Invalid Parm"
0412 00158 1B00000  D0=(5) =DERN
0413 0015F 142      A=DAT0 A A(A) := dernière
0414 00162 8BE      2C<=A A si lfin <= dernière
0415 00165 50      GOYES deft50 alors deft50
0416 00167 102      R2=A A sinon lfin := dernière
0417 0016A 119      deft60 C=R1 C(A) := ldebut
0418 0016D E4      A=A+1 A A(A) := dernière + 1
0419 0016F 8B2      7C=A A
0420 00172 90      GOYES invprm "Invalid Parm"
0421 00174 8A2      7C=A A
0422 00177 00      RTNYES Cy = 1. intervalle inexistant
0423 00179 01      RTN Cy = 0. intervalle existant
0424 0017B
0425 00178 3300000 invprm LC(4) =ellPAR "Invalid Parm"
0426 00181 8C00000 GOLONG =xederr
0427 00187
0428 00187
0429 00187      * valide, valid?
0430 00187
0431 00187      * But: valider ce qui a été pris (dans la ligne) par la cmd
0432 00187      * "valide" sort éventuellement en erreur, alors que
0433 00187      * "valid?" retourne à l'appelant, la Carry indiquant
0434 00187      * si une erreur a été trouvée.
0435 00187      * Entrée:
0436 00187      * - D1 = " position dans la ligne
0437 00187      * - D(A) = nb de caractères restant
0438 00187      * Sortie:
0439 00187      * - PCMD, LCHD mis à jour
0440 00187      * valid?
0441 00187      * Cy = 1 : erreur (C(4)=0 = numéro d'erreur)
0442 00187      * cy = 0 : ok
0443 00187      * Abime: A(B), C(A), D(A), D1, S0
0444 00187      * Appelle: skip, lookah, getchr
0445 00187      * Niveaux: 2 (skip)
0446 00187      * Note: cette routine vérifie qu'il n'y a plus rien sur la
0447 00187      * ligne de commande.
0448 00187      * Historique:
0449 00187      * 88/05/15: PD/JT conception & codage

```

Page 011 Editeur, execution des commandes <xcmd.as>  
AREUH ASS. V2.4 Utilitaires

```

0500 001E9
0501 001E9      * But: mettre l'information "mode interactif demandé" dans
0502 001E9      * variable globale QUERY.
0503 001E9      * Entrée:
0504 001E9      * - R3(S) = valeur laissée par la décomposition
0505 001E9      * Sortie:
0506 001E9      * - QUERY = R3(S)
0507 001E9      * Abime: C, D0
0508 001E9      * Appelle: -
0509 001E9      * Niveaux: 0
0510 001E9      * Historique:
0511 001E9      * 28/10/30: PD/JT isolement de "S"
0512 001E9
0513 001E9
0514 001E9 118      quer? C=R3
0515 001EC 1B00000 D0=(5) =QUERY
0516 001F3 1544      DAT0=C S
0517 001F7 01      RTN
0518 001F9
0519 001F9
0520 001F9
0521 001F9
0522 001F9
0523 001F9
0524 001F9
0525 001F9
0526 001F9
0527 001F9
0528 001F9
0529 001F9
0530 001F9
0531 001F9 1B00000 clrprt D0=(5) =KEYPTR
0532 00200 D2      C=0 A
0533 00202 1562      C=DAT0 XS
0534 00206 92E      7C#0 XS
0535 00209 00      RTNYES retour si buffer vide
0536 00208 1A00000 D0=(4) (=KEYBUF)*2+14
0537 00211 14C      DAT0=C B
0538 00214 01      RTN
0539 00216
0540 00216
0541 00216
0542 00216
0543 00216
0544 00216
0545 00216      * But: analyser les arguments des commandes L, P et J
0546 00216      * Entrée:
0547 00216      * - A(A) = dernière ligne par défaut
0548 00216      * - R0 = nombre de lignes introduites
0549 00216      * - R1 = ldebut
0550 00216      * - R2 = lfin
0551 00216      * - D(A) = LEN(psrnm)

```

Page 010 Editeur, execution des commandes <xcmd.as>  
AREUH ASS. V2.4 Utilitaires

```

0450 00187      * 88/10/29: PD/JT ajout de A(B) dans les "abimes"
0451 00187      * 88/11/11: PD/JT ajout de "valid?" pour D et X
0452 00187
0453 00187
0454 00187 850      valid? ST=1 0
0455 0018A 6500      GOTO va1000
0456 0018E 840      valide ST=0 0
0457 00191 8E00000 va1000 GOSUBL =skip
0458 00197 8E00000 GOSUBL =lookah
0459 0019D 411      GOC valok EOL trouvé
0460 001A0 8E00000 GOSUBL =getchr EOL non trouvé. Est-ce ':' ?
0461 001A6 31B3      LCAS0 ':'
0462 001AA 966      7A=C B
0463 001AD A1      GOYES va1100 non : "Invalid Cmd"
0464 001AF
0465 001A7 137      valok CDIEX
0466 001B2 1F00000 D1=(5) =PCMD
0467 001B9 145      DAT1=C A PCMD := D1
0468 001BC 1D00      D1=(2) =LCHD
0469 001C0 D8      C=D A
0470 001C2 145      DAT1=C A LCHD := D(A)
0471 001C5 03      RTNCC
0472 001C7
0473 001C7 3300000 va1100 LC(4) (=id)=(telCMD)
0474 001CD 870      7S=1 0 reporter simplement l'erreur ?
0475 001D0 00      RTNYES oui : Cy = 1
0476 001D2 501      GONC erreur B.E.T. "Invalid Cmd"
0477 001D5
0478 001D5
0479 001D5      * noquer
0480 001D5
0481 001D5      * But: vérifier que "query" = 0
0482 001D5      * Entrée:
0483 001D5      * - R3 = query (0 ou 1)
0484 001D5      * Sortie: -
0485 001D5      * Abime: C(W)
0486 001D5      * Niveaux: 0
0487 001D5      * Historique:
0488 001D5      * 88/05/15: PD/JT conception & codage
0489 001D5
0490 001D5
0491 001D5 118      noquer C=R3
0492 001D8 94A      7C#0 S
0493 001DB 00      RTNYES
0494 001DD
0495 001DD 3300000 invcmd LC(4) (=id)=(telCMD) "Invalid Cmd"
0496 001E3 8C00000 erreur GOLONG =xederr
0497 001E9
0498 001E9
0499 001E9      * quer?

```

Page 012 Editeur, execution des commandes <xcmd.as>  
AREUH ASS. V2.4 Utilitaires

```

0550 00216      * - D1 = " param
0551 00216      * - S0 = 1 si le nombre de lignes fourni par l'utilisateur
0552 00216      * ne comprend pas la ligne de début (utilisé par J)
0553 00216      * Sortie:
0554 00216      * - R1 = ldebut
0555 00216      * - R3 = nombres de lignes valides (déjà testé, évent. 0)
0556 00216      * Abime: A(W), B(A), C(A), D0, R3 (D1 et D(A) réactualisés)
0557 00216      * Appelle: nombre, skip, ASLW5, ASRW5
0558 00216      * Niveaux: 2
0559 00216      * Note: la première ligne par défaut est la courante (".")
0560 00216      * dans les deux cas (J et L/P).
0561 00216      * Historique:
0562 00216      * 88/10/29: PD/JT séparation de L
0563 00216
0564 00216
0565 00216 8F00000 abCn GOSBVL =ASLW5 A(S-5) := lfin si il n'y en a pas
0566 0021D 8E00000 GOSUBL =nombre
0567 00223 4A4      GOC abCn50 nombre non reconnu
0568 00226
0569 00226
0570 00226      * ... <xcmd> <nombre> ...
0571 00226      * le nombre est dans A(A)
0572 00226 860      7ST#0 0 cas standard ?
0573 00229 40      GOYES abCn05 oui
0574 00228 E4      A=A+1 A
0575 0022D 103      abCn05 R3=A R3 := nombre de lignes
0576 00230 8E00000 GOSUBL =skip Passer éventuellement le nombre
0577 00236
0578 00236
0579 00236
0580 00236 110      A=R0 A(A) := nb de paramètres
0581 00239 CC      A=A+1 A
0582 00238 4B0      GOC abCn10 0 paramètre
0583 0023E CC      A=A+1 A
0584 00240 431      GOC abCn20 1 paramètre
0585 00243 699F      GOTO invcmd "Invalid Cmd"
0586 00247
0587 00247
0588 00247 1B00000 * nb de paramètres = 0
0589 0024E 142      abCn10 D0=(5) =COUR
0590 00251 1B1      R1=A
0591 00254 111      abCn20 A=R1 A(A) := ldebut
0592 00257 118      C=R3 C(A) := compte de lignes
0593 0025A CA      A=A+C A
0594 0025C CC      A=A+1 A A(A) := numéro de la dernière ligne
0595 0025E 1900      D0=(2) =DERN
0596 00262 146      C=DAT0 A C(A) dernière
0597 00265 8BA      7A<=C A
0598 00268 00      RTNYES
0599 0026A 601F      GOTO invprm "Invalid Parm"

```

Page 013 Editeur, execution des commandes <xcmd.as>  
AREUH ASS. V2.4 Utilitaires

```

0600 0026E
0601 0026E
0602 0026E 8F00000
0603 00275 D6
0604 00277 1B00000
0605 0027E 142
0606 00281 76AE
0607 00285
0608 00285
0609 00285
0610 00285
0611 00285 111
0612 00288 11A
0613 0028B E2
0614 0028D E6
0615 0028F 10B
0616 00292
0617 00292
0618 00292
0619 00292
0620 00292 01
0621 00294

```

- pas de nombre de lignes fourni après 'L', 'P' ou 'J'
- abCn50 GOSBVL =ASRW5 A(A) := lfin
- C=A A C(A) := lfin
- D0=(5) =COUR
- A=DATA A A(A) := ldebut ('.')
- GOSUB default
- Cas Cy = 1 : intervalle inexistant. Typiquement, après les paramètres par défaut, on a : (\$\*1,\$)
- A=R1
- C=R2
- C=C-A A
- C=C+1 A
- R3=C R3 := nb de lignes (évent. 0)
- R1 = ligne de début
- R3 = nombre de lignes à afficher (déjà testé)
- RTN

Page 015 Editeur, execution des commandes <xcmd.as>  
AREUH ASS. V2.4 Commandes Exit / Quit

```

0650 002AA
0651 002AA
0652 002AA
0653 002AA
0654 002AA
0655 002AA
0656 002AA
0657 002AA
0658 002AA
0659 002AA
0660 002AA
0661 002AA
0662 002AA
0663 002AA
0664 002AA
0665 002AA
0666 002AA
0667 002AA
0668 002AA
0669 002AA
0670 002AA
0671 002AA
0672 002AA 623F
0673 002AE
0674 002AE 732F
0675 002B2 78DE
0676 002B6 118
0677 002B9 8AE
0678 002BC EE
0679 002BE
0680 002BE
0681 002BE
0682 002BE 1B00000
0683 002C5 D2
0684 002C7 1563
0685 002CB 8AA
0686 002CE 90
0687 002D0 8F00000
0688 002D7
0689 002D7
0690 002D7
0691 002D7
0692 002D7 1B00000
0693 002DE 1560
0694 002E2 8E0000
0695 002E8
0696 002E8
0697 002E8
0698 002E8 7D0F
0699 002EC 8F00000

```

- cmEXIT
- But: sortir de l'éditeur
- Entrée:
  - D(A) = LEN(param)
  - D1 = " param
  - R0 = nb de lignes introduites
  - R1 = ldebut
  - R2 = lfin
  - R3(S) = 1 si '?' avant la commande
- Sortie: par NXTSTH
- Abime: tout
- Appelle: noquer, valide, I/ODAL, clrprt, NOSCRL, setf11
- Niveaux: N/A
- Historique:
  - 88/05/15: PD/JT conception & codage
  - 88/10/23: PD/JT ajout du traitement du flag 1
  - 88/11/12: PD/JT traitement du flag 1 par setf11
- invCmd GOTO invcmd
- cmEXIT GOSUB noquer Interdit le '?'
- GOSUB valide Interdit des paramètres après nblignes
- C=R0
- ?C#0 A Paramètres avant ?
- GOYES invCmd oui : "Invalid Cmd"
- Collapser le buffer
- EXIT D0=(5) =BUFID
- C=0 A
- C=DATA X C(X) := buffer id
- ?C=0 A
- GOYES E050 Y-avait pas d'buffer
- GOSBVL =I/ODAL
- E050
- Mettre le flag 1 à son ancienne valeur
- D0=(5) =FLAG1
- C=DATA P C(0) := ancienne valeur
- GOSUBL =setf11
- Effacer la répétition, autoriser le prompt, Tout baigne...
- GOSUB clrprt
- GOSBVL =NOSCRL

Page 014 Editeur, execution des commandes <xcmd.as>  
AREUH ASS. V2.4 Commande nulle

```

0623 00294
0624 00294
0625 00294
0626 00294
0627 00294
0628 00294
0629 00294
0630 00294
0631 00294
0632 00294
0633 00294
0634 00294
0635 00294
0636 00294
0637 00294
0638 00294
0639 00294
0640 00294
0641 00294
0642 00294
0643 00294 7D3F
0644 00298 72FE
0645 0029C 72ED
0646 002A0 7E1E
0647 002A4 8C0000
0648 002AA

```

- cmNULL
- But: ne rien faire (ahhhhhhhhhhh)
- Entrée:
  - D(A) = LEN(param)
  - D1 = " param
  - R0 = nb de lignes introduites
  - R1 = ldebut
  - R2 = lfin
  - R3(S) = 1 si '?' avant la commande
- Sortie: par BOUCLE
- Appelle: noquer, valide, ck1prm, setcou
- Niveaux: N/A
- Historique:
  - 88/05/14: PD/JT conception & codage
  - 88/05/23: PD/JT simplification et utilisation de ck1prm
- cmNULL GOSUB noquer Interdit les "?"
- GOSUB valide Interdit tout après ce qui est lu
- GOSUB ck1prm force ldebut à COUR si besoin est
- GOSUB setcou
- GOLONG =BOUCLE

Page 016 Editeur, execution des commandes <xcmd.as>  
AREUH ASS. V2.4 Commandes Exit / Quit

```

0700 002F3 84E
0701 002F6 8D00000
0702 002FD

```

- SI=0 14
- GOVLNG =NXTSTH

```

0704 002FD .....
0705 002FD .....
0706 002FD .....
0707 002FD .....
0708 002FD .....
0709 002FD .....
0710 002FD .....
0711 002FD .....
0712 002FD .....
0713 002FD .....
0714 002FD .....
0715 002FD .....
0716 002FD .....
0717 002FD .....
0718 002FD .....
0719 002FD .....
0720 002FD .....
0721 002FD .....
0722 002FD .....
0723 002FD .....
0724 002FD .....
0725 002FD 3100 =cmPRNT LC(2) (=PRINT)*16*#F
0726 00301 6700 GOTO L000
0727 00305 .....
0728 00305 3100 =cmLIST LC(2) (=DISPT)*16*#F
0729 00309 1B00000 L000 D0=(5) =MLFFLG
0730 00310 14C DATA=C B MLFFLG := type (DISP ou PRINT)
0731 00313 .....
0732 00313 .....
0733 00313 .....
0734 00313 7E8E .....
0735 00317 .....
0736 00317 .....
0737 00317 .....
0738 00317 .....
0739 00317 1B00000 D0=(5) =DERN
0740 0031E 14C A=DATA A A(A) := ligne de fin par défaut
0741 00321 840 ST=0 0 cas standard
0742 00324 7EEE GOSUB abCn
0743 00328 .....
0744 00328 .....
0745 00328 .....
0746 00328 .....
0747 00328 113 A=R3 A(A) := nb de lignes
0748 00328 1B00000 D0=(5) remain
0749 00332 140 DATA=A A remain := compteur
0750 00335 111 A=R1
0751 00338 1900 D0=(2) cour
0752 0033C 140 DATA=A A cour := ldebut
0753 0033F .....

```

```

0804 00300 135 D1=C
0805 00303 07 D=C A D(A) := debut du buffer
0806 00305 8E0000 GOSUBL =Num201 Extrait de KA
0807 0030B 31A3 LCASC :=
0808 0030F 14D DAT1=C B
0809 0030C 02 C=0 A
0810 0030C 812 CSLC W C(A) := nombre de chiffres à envoyer
0811 00307 E6 C=C+1 A pour rattraper le "-1"
0812 00309 E6 C=C+1 A pour inclure ":"
0813 0030B DA A=C A A(A) := longueur
0814 0030D 08 C=D A C(A) := debut du buffer
0815 0030F 135 D1=C D1 := debut du buffer
0816 0030D 850 ST=1 =InEOL inhibit EOL pour le cas où
0817 00305 8F00000 GOSBVL =SENDWD
0818 0030C .....
0819 0030C .....
0820 0030C .....
0821 0030C 8E0000 L150 GOSUBL =LIFlen
0822 0030E 05 B=C A B(A) := longueur en octets
0823 0030E 163 D0=D0+ 4 D0 := partie données
0824 00307 136 CD0EX
0825 0030A 134 D0=C D0 gardé pour plus tard !!!
0826 0030D 850 ST=1 =InEOL inhibit EOL pour le cas où
0827 0030F 8F00000 GOSBVL =SENDWD+
0828 0030F 8F00000 GOSBVL =SENDEL EOL
0829 0030E .....
0830 0030E .....
0831 0030E .....
0832 0030E 183 D0=D0- 4 D0 était sauvé pendant SENDxxx
0833 00401 8E0000 GOSUBL =LIFlen
0834 00407 136 CD0EX
0835 0040A C2 C=A+C A C(A) := nouvelle ligne
0836 0040C 134 D0=C
0837 0040F .....
0838 0040F 1F00000 D1=(5) cour
0839 00416 147 C=DAT1 A
0840 00419 E6 C=C+1 A
0841 0041B 145 DAT1=C A ldebut++
0842 0041E 1000 D1=(2) remain
0843 00422 143 A=DAT1 A
0844 00425 CC A=A-1 A
0845 00427 141 DAT1=A A
0846 0042A .....
0847 0042A .....
0848 0042A .....
0849 0042A .....
0850 0042A 8F00000 GOSBVL =CK"DN" N'abime ni C(A) ni A(A)
0851 00431 580 GONC L190 [ATTN] hit
0852 00434 .....
0853 00434 .....

```

```

0754 0033F .....
0755 0033F .....
0756 0033F 1900 D0=(2) optn
0757 00343 AC2 C=0 A
0758 00346 1544 DATA=C A Par défaut : non
0759 0034A 8E0000 GOSUBL =lookah
0760 00350 4E1 GOC L070 pas de 'N'
0761 00353 31E4 LCASC 'N'
0762 00357 962 ?A=C B
0763 0035A B0 GOYES L060 il y a un 'N'
0764 0035C 31E6 LCASC 'n'
0765 00360 966 ?A#C B
0766 00363 C0 GOYES L070 il n'y a pas de 'N'
0767 00365 8E0000 L060 GOSUBL =getchr
0768 00368 1540 DATA=C P optn := #E
0769 0036F .....
0770 0036F .....
0771 0036F .....
0772 0036F 7B1E L070 GOSUB valide
0773 00373 .....
0774 00373 .....
0775 00373 .....
0776 00373 .....
0777 00373 118 C=R3 S C(A) := nombre de lignes à lister
0778 00376 8AE ?C#0 A
0779 00379 60 GOYES L080
0780 0037B 60D0 GOTO L999 nul : sortie...
0781 0037F 119 C=R1 C(A) := ldebut
0782 00382 8E0000 GOSUBL =seek D0 := " ligne
0783 00388 8F00000 GOSBVL =CK"INFO prepare HPIL
0784 0038F .....
0785 0038F .....
0786 0038F .....
0787 0038F .....
0788 0038F .....
0789 0038F .....
0790 0038F .....
0791 0038F .....
0792 0038F .....
0793 0038F 1F00000 L100 D1=(5) optn
0794 00396 1534 A=DAT1 S A(S) := 1 si 'N' active
0795 0039A 948 ?A=0 S
0796 0039D F3 GOYES L150
0797 0039F .....
0798 0039F .....
0799 0039F .....
0800 0039F 1000 D1=(2) cour
0801 003A3 143 A=DAT1 A A(A) := nombre à afficher
0802 003A6 1F00000 D1=(5) =AVHMS
0803 003AD 147 C=DAT1 A

```

```

0854 00434 .....
0855 00434 8A8 .....
0856 00437 11 GOYES L200
0857 00439 655F GOTO L100 et on repart pour un tour...
0858 0043D .....
0859 0043D .....
0860 0043D .....
0861 0043D DA L190 A=C A
0862 0043F 8F00000 GOSBVL =POPBUF Enlever la touche [ATTN]
0863 00445 D6 C=A A
0864 00448 .....
0865 00448 .....
0866 00448 .....
0867 00448 .....
0868 00448 767C L200 GOSUB setcou COUR := C(A) (qui valait ldebut++)
0869 0044C .....
0870 0044C 8C0000 L999 GOLONG =BOUCLE
0871 00452 .....

```

```

0873 00452 .....
0874 00452 .....
0875 00452 .....
0876 00452 .....
0877 00452 .....
0878 00452 .....
0879 00452 .....
0880 00452 .....
0881 00452 .....
0882 00452 .....
0883 00452 .....
0884 00452 .....
0885 00452 .....
0886 00452 .....
0887 00452 .....
0888 00452 .....
0889 00452 .....
0890 00452 .....
0891 00452 .....
0892 00452 .....
0893 00452 .....
0894 00452 .....
0895 00452 .....
0896 00452 .....
0897 00452 .....
0898 00452 .....
0899 00452 .....
0900 00452 .....
0901 00452 .....
0902 00452 .....
0903 00452 .....
0904 00452 .....
0905 00452 .....
0906 00452 .....
0907 00452 .....
0908 00452 .....
0909 00452 .....
0910 00452 .....
0911 00452 .....
0912 00452 .....
0913 00452 .....
0914 00452 .....
0915 00452 .....
0917 00455 301 .....
0918 00455 8E0000 .....
0919 00455 .....
0920 00455 .....
0921 00455 6500 .....
0922 00455 .....

```

- cmINS
- But: insérer une ou plusieurs lignes dans le texte
- Entrées:
  - D(A) = LEN(param)
  - D1 = param
  - R0 = nb de lignes introduites
  - R1 = ldebut
  - R2 = lfin
  - R3(S) = 1 si '?' avant la commande
- Sorties: par BOUCLE
- Abime: tout
- Appelles: setfll, noquer, skiprm, valide, seek, editlin, TBLJMC, OBCOLL, MEMCKL, SWPBYT, LIFlen, rpllin, addlin, RFLLIN, syncbf
- Niveaux: N/A
- Algorithmes:
  - si 1
  - alors mettre le flag 1 à 1
  - fin si
  - vérifier les paramètres
  - mettre à 1 notre bit dans la Reserved-Ram
  - tant que l'utilisateur n'a pas frappé [ATTN]
  - faire
  - éditer la ligne courante
  - selon la réponse de l'utilisateur
  - [ATTN] = rien
  - [ENDLINE] = insérer ou ajouter la ligne
  - touche de curseur = déplacer la ligne courante
  - fin tant que
  - Mettre le flag 1 à 0
- Historique:
  - 88/05/23: PD/JT conception & codage
  - 88/10/23: PD/JT ajout du flag dans la reserved-ram
  - 89/06/17: PD/JT test du flag "read-only"

```

=cmINS
•
• Positionner le flag 1 (1 comme "insert" (franch-joke))
•
LC(1) 1
GOSUBL =setfll
•
• C(0) n'est pas modifié. Il vaut donc toujours 1
•
GOTO T000 On y va !

```

```

0973 004F1 04 .....
0974 004F3 07 .....
0975 004F5 .....
0976 004F5 .....
0977 004F5 .....
0978 004F5 8F0000 .....
0979 004FC 081 .....
0980 004FF 810 .....
0981 00502 E8F .....
0982 00505 .....
0983 00505 .....
0984 00505 .....
0985 00505 .....
0986 00505 .....
0987 00505 BF2 .....
0988 00508 8E0000 .....
0989 0050E BF6 .....
0990 00511 8C0000 .....
0991 00517 .....
0992 00517 .....
0993 00517 .....
0994 00517 .....
0995 00517 .....
0996 00517 .....
0997 00517 .....
0998 00517 137 .....
0999 0051A 1F0000 .....
1000 00521 145 .....
1001 00524 1000 .....
1002 00528 141 .....
1003 00528 .....
1004 0052B .....
1005 0052B .....
1006 0052B .....
1007 00528 1F0000 .....
1008 00532 1574 .....
1009 00536 8E0000 .....
1010 0053C 58C .....
1011 0053F .....
1012 0053F .....
1013 0053F .....
1014 0053F 8F0000 .....
1015 00546 1F0000 .....
1016 00540 143 .....
1017 00550 E4 .....
1018 00552 E4 .....
1019 00554 C4 .....
1020 00556 D6 .....
1021 00558 8FA000 .....
1022 0055F 45A .....

```

- A=B A A(A) := LEN si [ENDLINE]
- C=RSTK A C(0) := code retour
- Selon le code retour de editlin
- GOSBVL =TBLJMC n'abime que C(A)
- REL(3) T900 [ATTN] ou [F][OFF]
- REL(3) T400 [ENDLINE]
- REL(3) T100 curseur
- Remettre le flag 1 à 0 et brancher en erreur.
- Terr CSL W C(5-1) := no d'erreur, C(0) := 0
- GOSUBL =setfll Mettre le flag 1 à 0
- CSR W C(A) := no d'erreur
- GOLONG =xederr
- [ENDLINE]
- D1 = premier caractère
- D(A) = A(A) = longueur de la chaîne
- T400 CDIEX
- D1(5) stradr
- DAT1=C A stradr := premier caractère
- D1(2) strlen
- DAT1=A A strlen := longueur en octets
- Ajout du 12/11/88 : test de l'accessibilité du fichier.
- Changement du 17/06/89 : test du flag "read-only"
- D1(5) =RDONLY
- C=DAT1 S
- GOSUBL =WRITE?
- GONC Terr C(A) = eFACCS, eFPROT ou eFOPEN
- Fin de l'ajout du 12/11/88
- GOSBVL =OBCOLL
- D1(5) strlen
- A=DAT1 A
- A=A+1 A
- A=A+1 A
- A=A+1 A
- C=A A A(A) := mémoire nécessaire
- C=A A C(A) := mémoire demandée
- GOSBVL =MEMCKL
- GOC Terr C(A) = eMEM

```

0923 0045F DC .....
0924 00461 1B0000 .....
0925 00463 1540 .....
0926 0046C .....
0927 0046C 7550 .....
0928 00470 7E0C .....
0929 00474 7610 .....
0930 00475 113 .....
0931 00478 1B0000 .....
0932 00482 144 .....
0933 00485 .....
0934 00485 1900 .....
0935 00489 501 .....
0936 00490 1540 .....
0937 00490 .....
0938 00490 .....
0939 00490 .....
0940 00490 .....
0941 00490 .....
0942 00490 1B0000 .....
0943 00497 146 .....
0944 0049A 8E0000 .....
0945 004A0 1F0000 .....
0946 004A7 136 .....
0947 004AA 145 .....
0948 004AD 134 .....
0949 004B0 1F0000 .....
0950 004B7 1574 .....
0951 004B8 .....
0952 004B8 .....
0953 004B8 .....
0954 004B8 .....
0955 004B8 1F0000 .....
0956 004C2 1530 .....
0957 004C6 300 .....
0958 004C9 0E0E .....
0959 004CD 1510 .....
0960 004D1 .....
0961 004D1 8E0000 .....
0962 004D7 .....
0963 004D7 .....
0964 004D7 .....
0965 004D7 06 .....
0966 004D9 08 .....
0967 004DB 1B0000 .....
0968 004E2 1520 .....
0969 004E6 300 .....
0970 004E9 0E0E .....
0971 004ED 1500 .....
0972 004F1 .....

```

```

=cmTEXT C=A A
T000 D0(5) insert
DAT0=C P Insert := 0 ou 1
GOSUB noquer Interdit le '?'
GOSUB skiprm N'autorise qu'un seul paramètre
GOSUB valide
C=R1 C(A) := ldebut
D0(5) =COUR C(A) := paramètre
DAT0=C A COUR := paramètre
D0(2) =lastly
LC(1) 1
DAT0=C P lastly = [v] ou [g][v]
T100
•
• afficher et éditer la ligne courante
•
D0(5) =COUR
C=DAT0 A C(A) := courante
GOSUB =seek D0 := longueur LIF de la ligne
D1(5) curlin
CDIEX
DAT1=C A curlin := ligne courante
D0=C
D1(5) insert
C=DAT1 S C(S) := mode ('T' ou 'I')
•
• Mettre le bit de la reserved-ram pour signaler que nous
• sommes en input.
•
D1(5) =RESJPC ram-reservée JPC
A=DAT1 F
LC(1) =INEDIT
A=ATC F
DAT1=A F
GOSUBL =editlin
•
• Enlever le bit de la reserved-ram
•
RSTN=C préserver C(0)
B=A A B(A) := LEN si [ENDLINE]
D0(5) =RESJPC
A=DAT0 P
LC(1) =INEDIT
A=ASC F
DAT0=A F

```

```

1023 00562 .....
1024 00562 1B0000 .....
1025 00569 146 .....
1026 0056C 135 .....
1027 0056F 1900 .....
1028 00573 146 .....
1029 00576 05 .....
1030 00578 .....
1031 00578 E6 .....
1032 0057A 81E .....
1033 0057D C6 .....
1034 0057F E6 .....
1035 00581 E6 .....
1036 00583 C6 .....
1037 00585 .....
1038 00585 1B0000 .....
1039 0058C 142 .....
1040 0059F C2 .....
1041 00591 144 .....
1042 00594 130 .....
1043 00597 .....
1044 00597 04 .....
1045 00599 8F0000 .....
1046 005A0 1583 .....
1047 005A4 163 .....
1048 005A7 5E0 .....
1049 005AA .....
1050 005AA 14F .....
1051 005AD 14C .....
1052 005B0 1C1 .....
1053 005B3 161 .....
1054 005B6 0D .....
1055 005B8 51F .....
1056 005B8 .....
1057 005B8 .....
1058 005B8 .....
1059 005B8 1F0000 .....
1060 005C2 147 .....
1061 005C5 1000 .....
1062 005C9 143 .....
1063 005CC E2 .....
1064 005CE 07 .....
1065 005D0 .....
1066 005D0 1F0000 .....
1067 005D7 147 .....
1068 005DA 15 .....
1069 005DC .....
1070 005DC 1B0000 .....
1071 005E3 1564 .....
1072 005E7 94E .....

```

- D0(5) stradr
- C=DAT0 A
- D1=C
- D0(2) strlen
- C=DAT0 A
- B=C A B(A) := longueur en octets
- C=C+1 A
- CSR B
- C=C A C(A) := longueur avec padding LIF
- C=C+1 A
- C=C+1 A C(A) := longueur + LEN(longueur LIF)
- C=C+C A
- D0(5) =AVMEMS
- A=DAT0 A
- C=A+C A C(A) := fin de la chaîne
- DAT0=C A AVMEMS réactualisé
- D0=A D0 := output buffer
- A=B A A(A) := longueur en octets
- GOSBVL =SWPBYT
- DAT0=A 4 longueur LIF
- D0=D0+4
- GONC T420 B.E.T.
- T410 C=DAT1 B
- DAT0=C B
- D1=D1-2
- D0=D0+2
- T420 B=B-1 A
- GONC T410
- 
- OUTBS contient une ligne prête
- D1(5) =AVMEMS
- C=DAT1 A
- D1(2) =OUTBS
- A=DAT1 A
- C=C-A A C(A) := longueur nouvelle ligne
- D=C A D(A) := longueur nouvelle ligne
- 1065 005D0
- D1(5) =COUR
- C=DAT1 A
- B=C A B(A) := numéro de la ligne remplacée
- 1069 005DC
- D0(5) insert
- C=DAT0 S
- ?C#0 S

```

1073 005EA 34      GOYES T460 Insertion
1074 005EC      *
1075 005EC      * remplacement
1076 005EC      *
1077 005EC 1000    T450 D1=(2) =DERN
1078 005F0 143    A=DAT1 A
1079 005F3 2B2    ?>A A
1080 005F6 73      GOYES T460 Insertion pour [Eof]
1081 005F8
1082 005F8 1B0000 D0=(5) curlin
1083 005FF 146    C=DAT0 A
1084 00602 134    D0=C
1085 00605 8E0000 GOSUBL =LIFlen epargne B(A) et D(A)
1086 00608 DB     C=D A C(A) := longueur nouvelle
1087 0060D E2     C=C-A A C(A) := nouvelle - ancienne
1088 0060F DA     A=C A A(A) := différence de longueur
1089 00611
1090 00611 8E0000 GOSUBL =rpllin attention : ce n'est pas RPLLIN !!!
1091 00617
1092 00617 8E0000 GOSUBL =LIFlen
1093 0061D 103     R3=A R3 := longueur de la vieille ligne
1094 00620 136     CD0EX
1095 00623 CA     A=A+C A A(A) := " juste après
1096 00625 6420   GOTO T490
1097 00629
1098 00629 6BDE   TerR GOTO Terr Rallonge
1099 0062D
1100 0062D      *
1101 0062D      * insertion
1102 0062D
1103 0062D DB     T460 C=D A
1104 0062F DA     A=C A A(A) := longueur de ce qu'on rajoute
1105 00631
1106 00631 D2     C=D A
1107 00633 E6     C=C+1 A C(A) := 1 ligne ajoutée
1108 00635 8E0000 GOSUBL =addlin
1109 0063B
1110 0063B      *
1111 0063B      * Ajout d'une nouvelle ligne dans le fichier
1112 0063B 1B0000 D0=(5) curlin
1113 00642 D2     C=D A
1114 00644 19B    R3=C R3 := 0 pour insertion
1115 00647 142    A=DAT0 A A(A) := " juste devant l'insertion
1116 0064A
1117 0064A 1B0000 T490 D0=(5) =FILADR
1118 00651 146    C=DAT0 A C(A) := " file header
1119 00654
1120 00654      *
1121 00654      * A(A) = " dernier nibble + 1 de la vieille ligne
1122 00654      * C(A) = " file header
1123 00654      * R3(A) = longueur de la vieille ligne en nibs

```

```

1149 0065A
1150 0065A      *
1151 0065A      * cmSRCH
1152 0065A      *
1153 0065A      * But: chercher une chaîne dans le fichier
1154 0065A      * Entrée:
1155 0065A      * - D(A) = LEN(param)
1156 0065A      * - D1 = param
1157 0065A      * - R0 = nb de lignes introduites
1158 0065A      * - R1 : ldebut
1159 0065A      * - R2 : lfin
1160 0065A      * - R3(S) = 1 si "?" avant la commande
1161 0065A      * Sortie: par BOUCLE
1162 0065A      * Abime: tout
1163 0065A      * Appelle: quer?, set, $, GETSRC, getchr, valide, seek,
1164 0065A      * SRCADR, query, TBLJMC, ENDPOS, setcou
1165 0065A      * Niveaux: N/A
1166 0065A      * Algorithme:
1167 0065A      * compiler la chaîne à chercher
1168 0065A      * tant que non trouvé
1169 0065A      * faire
1170 0065A      * parcourir le fichier pour trouver une occurrence
1171 0065A      * si non trouvée
1172 0065A      * alors sortir en erreur
1173 0065A      * fin si
1174 0065A      * si mode # "?" (query)
1175 0065A      * alors
1176 0065A      * positionner la ligne courante
1177 0065A      * retour à la boucle de l'éditeur
1178 0065A      * fin si
1179 0065A      * demander confirmation
1180 0065A      * si confirmation = "ok"
1181 0065A      * alors
1182 0065A      * positionner la ligne courante
1183 0065A      * retour à la boucle de l'éditeur
1184 0065A      * fin si
1185 0065A      * fin tant que
1186 0065A      * Historique:
1187 0065A      * 88/07/02: PD/JT conception & codage
1188 0065A      * 88/11/13: PD/JT utilisation de svRIR2
1189 0065A
1190 0065A
1191 0065A 330000 notfnd LC(4) (=id)"(=NFND) "Not Found"
1192 00650 8C0000 erreuR GOLONG =vederr
1193 00656
1194 00656 7F4B   =cmSRCH GOSUB quer? QUERY := R3(S)
1195 0065A
1196 0065A 717A   GOSUB set,$
1197 0065E 4BE     GOC notfnd
1198 006A1

```

```

1123 00654
1124 00654 8F0000 GOSBVL =HGOSUB
1125 00658 000000 CON(5) =RPLLIN
1126 00660 48C    GOC TerR C(A) = eMEM
1127 00663
1128 00663 8E0000 GOSUBL =synchf synchronise l'adresse du buffer
1129 00669
1130 00669 1B0000 D0=(5) =COUR
1131 00670 146    C=DAT0 A
1132 00673 E6     C=C+1 A
1133 00675 144    DAT0=C A
1134 00678 671E   GOTO T100
1135 0067C
1136 0067C      * [ATTN] ou [f][OFF]
1137 0067C      * T900
1138 0067C
1139 0067C      * Remettre le flag 1 à 0. Si on était en mode "T".
1140 0067C      * c'est une no-op. Mais ça enlève un test et ça diminue
1141 0067C      * la taille du code (il n'y a pas de petit profit...).
1142 0067C
1143 0067C D2     C=D A
1144 0067E 8E0000 GOSUBL =setf11
1145 00684
1146 00684 8C0000 GOLONG =BOUCLE
1147 0068A

```

```

1199 006A1
1200 006A1      *
1201 006A1      * Sauver R1 et R2 dans ldeb et lfin
1202 006A1 7B59   GOSUB svRIR2
1203 006A5
1204 006A5 8E0000 GOSUBL =GETSRC analyse, compile chaîne de recherche
1205 006AB 44E    GOC erreuR Erreur
1206 006AE
1207 006AE 8E0000 GOSUBL =getchr passer le delimitateur éventuel
1208 006B4
1209 006B4
1210 006B4      *
1211 006B4      * L'état de la Carry nous importe peu, car on est soit
1212 006B4      * sur le "/" (et alors Cy = 0 en sortie), soit sur le EOL
1213 006B4      * et à ce moment, il ne restera plus de caractère à lire.
1214 006B4      * Et la commande sera donc finie de toutes manières.
1215 006B4
1216 006B4 76DA   GOSUB valide
1217 006B8
1218 006B8 1F0000 D1=(5) =FUNCR0 FUNCR0 := R0
1219 006BF 118     C=R0
1220 006C2 1557   DAT1=C W
1221 006C5
1222 006C5 1B0000 D0=(5) ldeb
1223 006CD 146    C=DAT0 A
1224 006D0 8E0000 GOSUBL =seek D0 := " première ligne
1225 006D6
1226 006D6 1F0000 D1=(5) =FUNCR0
1227 006E0 1577   C=DAT1 W
1228 006E1 108     R0=C
1229 006E4
1230 006E4 136    CD0EX debadr := " ligne début
1231 006E7 1B0000 D0=(5) debadr
1232 006EE 144    DAT0=C A
1233 006F1 1900   D0=(2) offset offset := 0
1234 006F5 D2     C=D A
1235 006F7 144    DAT0=C A
1236 006FA
1237 006FA      *
1238 006FA      * while non trouvée
1239 006FA      * faire
1240 006FA      * ...
1241 006FA S100 D1=(5) offset B(A) := offset
1242 00701 147    C=DAT1 A
1243 00704 D5     B=C A
1244 00706 1000   D1=(2) debadr D0 := " début ligne LIF
1245 0070A 147    C=DAT1 A
1246 00710 134    D0=C
1247 00718 1000   D1=(2) ldeb A(A) := ligne début
1248 0071A 143    A=DAT1 A
1249 00717 1000   D1=(2) lfin C(A) := ligne fin
1250 0071B 147    C=DAT1 A

```

```

1249 0071E *
1250 0071E * A(A) = ligne de début
1251 0071E * C(A) = ligne de fin
1252 0071E * D0 = " ligne LIF
1253 0071E * B(A) = offset
1254 0071E * R0(7-0) = caractéristique du buffer
1255 0071E *
1256 0071E GOSUBL =SRCADR search line in file
1257 00724 *
1258 00724 * A(A) = no ligne (-) ldeb)
1259 00724 * D1 = " longueur LIF (-) curlin)
1260 00724 * C(A) = longueur occurrence (-) poubelle...)
1261 00724 * D0 = " occurrence
1262 00724 *
1263 00724 *
1264 00724 *
1265 00724 * Libération des registres A et D1.
1266 00724 * Attention ! Le registre D0 est toujours utilisé
1267 00724 *
1268 00724 D0 B=A A B(A) := sauvegarde de A(A)
1269 00726 133 ADIEX A(A) := " longueur LIF
1270 00729 1F0000 D1=(5) debadr debadr := " longueur LIF
1271 00730 141 DAT1=A A
1272 00733 1000 D1=(2) ldeb ldeb := no ligne
1273 00737 04 A=B A
1274 00739 141 DAT1=A A on a libere A(A) et D1
1275 0073C *
1276 0073C * si non trouvée
1277 0073C * alors sortir en erreur
1278 0073C *
1279 0073C 5E4 GONC Serr match not found
1280 0073F *
1281 0073F * fin si
1282 0073F * si mode = "?" (query)
1283 0073F * alors
1284 0073F * positionner la ligne courante
1285 0073F * retour à la boucle
1286 0073F *
1287 0073F 1E0000 D1=(1) =QUERY
1288 00745 1534 A=DAT1 S
1289 00749 948 JAVE S
1290 0074C 94 GOVES S900 match found
1291 0074E *
1292 0074E *
1293 0074E * fin si
1294 0074E * demander confirmation
1295 0074E D1=(4) debadr C(8) := " longueur LIF
1296 00754 AF2 C=0 W
1297 00757 147 C=DAT1 A
1298 0075A 132 ADREX A 1/4 := occurrence

```

```

1335 007AF *
1336 007AF *
1337 007AF * cmREPL
1338 007AF *
1339 007AF * But: remplacer un motif par un autre dans le fichier
1340 007AF * Entrée:
1341 007AF * - D(A) = LEN(param)
1342 007AF * - D1 = " param
1343 007AF * - R0 = nb de lignes introduites
1344 007AF * - R1 : ldebut
1345 007AF * - R2 : lfin
1346 007AF * - R3(S) = 1 si '?' avant la commande
1347 007AF * Sortie: par BOUCLE
1348 007AF * Abime: tout
1349 007AF * Appelle: quer?, default, getchr, GETSRC, valide, CALCab,
1350 007AF * SRCLIN, LIFlen, MEMCKL, MOVE=M, REPLIN, FMTLIF, RPLLIN,
1351 007AF * synchr, rpllin, UPDTR0, ENDPOS, setcou
1352 007AF * Niveau: N/A
1353 007AF * Algorithmes:
1354 007AF * compiler la chaîne à chercher
1355 007AF * tant que ligne < lfin
1356 007AF * faire
1357 007AF * parcourir le fichier pour trouver une occurrence
1358 007AF * copier la ligne dans OUTBS
1359 007AF * [
1360 007AF * répéter
1361 007AF * si mode = "?" alors poser la question
1362 007AF * si réponse = "Y" ou pas de "?"
1363 007AF * alors
1364 007AF * remplacer dans OUTBS
1365 007AF * fin si
1366 007AF * chercher une autre occurrence sur la même ligne
1367 007AF * jusqu'à ce qu'il n'y ait plus d'occurrence
1368 007AF * ]
1369 007AF * si ligne modifiée alors remplacer dans le fichier
1370 007AF * fin tant que
1371 007AF * Historique:
1372 007AF * 88/10/07: PD/JT conception & codage
1373 007AF * 88/11/12: PD/JT ajout du test d'accessibilité
1374 007AF * 88/11/12: PD/JT séparation de set.db
1375 007AF * 88/11/13: PD/JT utilisation de svRIR2
1376 007AF * 89/06/17: PD/JT test du flag "read-only"
1377 007AF *
1378 007AF *
1379 007AF 330000 Notfnd LC(4) (=id)=(=NFND) "Not Found"
1380 007B5 8C0000 Erreur GOLONG =xvterr
1381 007B8 *
1382 007B8 7A2A =cmREPL GOSUB quer? QUERY := R3(S) ('?' ou pas)
1383 007BF *
1384 007BF * set., ou "setRIR1"

```

```

1309 0075D 134 D0=C A D0 := " longueur LIF (debadr)
1310 0075E EE C(A) := A C(A) := offset en quarts
1311 0075E 51E /SRB C(A) := offset en octets
1312 0075E 1E C=C-1 A C(A) := offset - longueur LIF + 1
1313 00757 1D00 D1=(2) offset
1314 00758 145 DAT1=C A
1315 0075E *
1316 0075E D1=(2) ldeb
1317 00772 143 A=DAT1 A
1318 00775 *
1319 00775 * A(A) = no ligne (-) ldeb)
1320 00775 * C(A) = no colonne (-) offset)
1321 00775 * D0 = " début de ligne (-) debadr)
1322 00775 *
1323 00775 8E0000 GOSUBL =query
1324 00778 8F0000 GOSBVL =TBLJMC
1325 00782 120 REL(3) S95R [Q] exit (ou [ATTN])
1326 00785 010 REL(3) S300 [Y] match found
1327 00788 27F REL(3) S100 [EN] continue
1328 00788 *
1329 00788 * Match not found
1330 00788 *
1331 00788 * Serr GOSUBL =ENDPOS
1332 00788 8E0000 GOTO notfnd
1333 00795 *
1334 00795 * Match found !
1335 00795 * ldeb = numéro de la ligne trouvée
1336 00795 *
1337 00795 S900 D1=(5) ldeb
1338 0079C 147 C=DAT1 A C(A) := numéro ligne trouvée
1339 0079F 7F19 GOSUB setcou
1340 007A3 8E0000 S950 GOSUBL =ENDPOS on a fini avec le buffer
1341 007A9 8C0000 GOLONG =BOUCLE
1342 007AF *

```

```

1335 007BF *
1336 007BF 8E329F * GOSUBL set.db (... ldebut)
1337 007C5 *
1338 007C5 * R1 et R2 contiennent maintenant les bonnes bornes.
1339 007C5 *
1340 007C5 8E538F GOSUBL svRIR2 alors les sauve dans ldeb et lfin
1341 007CB *
1342 007CB GOSUBL =GETSRC analyse et compile le motif
1343 007D1 43E GOC Erreur Erreur
1344 007D4 8E0000 GOSUBL =DELIM délimite la chaîne de remplacement
1345 007DA 521 GONC R030 pas d'erreur
1346 007D0 *
1347 007D0 * Génère une erreur, mais libère auparavant le buffer
1348 007D0 * utilisé pour stocker la chaîne générique.
1349 007D0 *
1350 007D0 Rerr D=C A D(A) := numéro d'erreur
1351 007D7 8E0000 GOSUBL =ENDPOS n'abime pas D(A)
1352 007E5 08 C=D A restaure dans C(A) le No d'erreur
1353 007E7 8C0000 GOLONG =xvterr
1354 007ED *
1355 007ED R030 D0=(5) remp
1356 007F4 144 DAT0-C A remp+0 := nb de caractères
1357 007F7 164 D0=D0+5
1358 007FA 140 DAT0-A A remp+5 := " chaîne de remplacement
1359 007FD 8E0000 GOSUBL =getchr passer le delimitaire
1360 00803 *
1361 00803 * Lorsque "valide" détecte une erreur, il faut libérer le
1362 00803 * buffer utilisé pour stocker la chaîne générique.
1363 00803 *
1364 00803 GOSUB valid?
1365 00807 45D GOC Rerr Libérer le buffer
1366 0080A *
1367 0080A * Tester l'accessibilité du fichier. Teste s'il est en Ram.
1368 0080A * si tel est le cas, que le fichier n'est pas sécurisé.
1369 0080A * Changement du 17/06/89 : test du flag "read-only"
1370 0080A *
1371 0080A D1=(5) =RDONLY
1372 00811 1574 C=DAT1 S
1373 00815 8E0000 GOSUBL =WRITE?
1374 0081B 51C GONC Rerr C(A) = eFACCS, eFPROT ou eFOPEN
1375 0081E *
1376 0081E D0=(5) dern dernière ligne modifiée
1377 00825 02 C=0 A
1378 00827 144 DAT0-C A dernière ligne modifiée := 0
1379 0082A *
1380 0082A * Calculer les coefficients a et b pour le remplacement
1381 0082A * Ces coefficients sont calculés une fois pour toutes et
1382 0082A * permettent de connaître la longueur de la chaîne
1383 0082A * remplacée en fonction de la longueur de la chaîne
1384 0082A

```



Page 033 Editeur, execution des commandes (xcmd.as)  
AREUH ASS. V2.4 Commande Replace

```

1435 0082A      * originale. Ils sont placés en haut de TRFMBF.
1436 0082A      *
1437 0082A 1A0000      D0=(4) remp
1438 00830 15E9        C=DAT0 10
1439 00834 10B        R3=C
1440 00837 8E0000      GOSUBL =CALCab Voila, c'est fait !
1441 0083D
1442 0083D
1443 0083D      *
1444 0083D      * Boucle de remplacement.
1445 0083D      * Assertions :
1446 0083D      * ldeb = no dernière ligne où on a trouvé quelque chose
1447 0083D      * lfin = no ligne limite
1448 0083D      * dern = no dernière ligne modifiée
1449 0083D      * remp = 10 q. pour conserver R3 entre les appels à REPLIN
1450 0083D 1B00000      R100 D0=(5) ldeb
1451 00844 142          A=DAT0 A
1452 00847 160          D0=D0+ (lfin)-(ldeb)
1453 0084A 146          C=DAT0 A
1454 0084D 8B6          ?A>C A
1455 00850 80          GOYES r900 On est arrivé au bout
1456 00852
1457 00852      * Recherche
1458 00852      * R0(7-0) = caractéristiques du buffer
1459 00852      * A(A) = ligne de début
1460 00852      * C(A) = ligne de fin
1461 00852
1462 00852 8E0000      GOSUBL =SRLLIN
1463 00858 460          GOC R110 trouvé !
1464 0085B 6B71        r900 GOTO R900 terminé. On ferme...
1465 0085F
1466 0085F      *
1467 0085F      * A(A) = numéro de la ligne trouvée
1468 0085F      * C(A) = longueur de l'occurrence trouvée (en octets)
1469 0085F      * D0 = occurrence trouvée
1470 0085F      * D1 = début ligne (longueur LIF) contenant l'occurrence
1471 0085F 06          R110 RSTK=C RSTK := LEN(occurrence)
1472 00861 136          CD0EX C(A) := occurrence
1473 00864
1474 00864 1B00000      D0=(5) savlin savlin := (no ligne, " ligne)
1475 0086B 140          DAT0=A A savlin + 0 := no ligne trouvée
1476 0086E 164          D0=D0+ 5
1477 00871 133          ADIEX A(A) := " début ligne LIF
1478 00874 140          DAT0=A A savlin + 5 := " début ligne LIF
1479 00877
1480 00877 E2          C=C-A A C(A) := offset de l'occurrence
1481 00879 06          RSTK=C RSTK := offset occurrence
1482 0087E
1483 0087E      *
1484 0087E      * RSTK(1) = LEN (occurrence)
1484 0087E      * RSTK(0) = offset occurrence dans la ligne

```

Page 035 Editeur, execution des commandes (xcmd.as)  
AREUH ASS. V2.4 Commande Replace

```

1535 008D6
1536 008D6 07          C=RSTK C(A) := LEN (occurrence)
1537 008D8
1538 008D8 1E0000      D1=(4) savlin
1539 008DE 143          A=DAT1 A A(A) := no de la ligne trouvée
1540 008E1
1541 008E1      *
1542 008E1      * (OUTBS..AVMEMS) = chaîne à traiter
1543 008E1      * R3 = motif de remplacement
1544 008E1      * D0 = occurrence trouvée
1545 008E1      * C(A) = longueur de l'occurrence
1546 008E1      * coeffa et coeff calculés par CALCab
1547 008E1      * A(A) = no de la ligne en cours
1548 008E1
1549 008E1 8E0000      GOSUBL =REPLIN
1550 008E7 5A0          GONC R150 pas e0 d'erreur
1551 008EA 62FE          rerr GOTO Rerr on libère le buffer
1552 008EE 69B0          r200 GOTO R200 rallonge
1553 008F2
1554 008F2      *
1555 008F2      * C(0) = code de retour
1556 008F2      * savlin + 0 = no de la ligne trouvée
1557 008F2      * savlin + 5 = " début de la ligne dans le fichier
1558 008F2      * (OUTBS..AVMEMS) = la ligne remplacée
1559 008F2
1559 008F2 1F00000      R150 D1=(5) =TMP3
1560 008F9 1550          DAT1=C P TMP3(0) := code de retour
1561 008FD 822          SB=0
1562 00900 81E          CSRB
1563 00903 832          ?SB=0
1564 00906 8E          GOYES r200 réécriture nécessaire ?
1565 00908
1566 00908      *
1567 00908      * Réécriture nécessaire
1568 00908      * TMP3(0) = code de retour
1569 00908      * savlin + 0 = no de la ligne trouvée
1570 00908      * savlin + 5 = " début de la ligne dans le fichier
1571 00908      * (OUTBS..AVMEMS) = la ligne remplacée
1572 00908
1573 00908
1574 00908      *
1575 00908      * RPLIN abimant RR, il faut le sauvegarder momentanément
1576 00908 1B00000      D0=(5) =FUNCR0
1577 0090F 118          C=R0
1578 00912 15C7        DAT0=C 8 sauvegarder RR(7-0)
1579 00916
1580 00916
1581 00916      *
1582 00916      * La ligne dans OUTBS a été modifiée. Il faut remettre une
1583 00916      * longueur LIF et un padding éventuel :
1584 00916
1584 00919 8E0000      ST=1 0 ajuster AVMEMS si besoin est
1584 00919 8E0000      GOSUBL =FMTLIF formate la ligne en (OUTBS..AVMEMS)

```

Page 034 Editeur, execution des commandes (xcmd.as)  
AREUH ASS. V2.4 Commande Replace

```

1485 00878      * savlin + 0 = no ligne trouvée
1486 00878      * savlin + 5 = " ligne trouvée
1487 00878      * A(A) = " ligne trouvée
1488 00878
1489 00878      *
1490 00878      * Maintenan. amener la ligne dans OUTBS
1491 0087B 130          D0=A D0 := " début ligne LIF (longueur)
1492 0087E 8E0000      GOSUBL =LIFlen C(A) := line length in bytes
1493 00884 E6          C=C+1 A
1494 00886 E6          C=C+1 A C(A) := LEN + 2
1495 00888 C6          C=C+A C(A) := longueur en quarts
1496 0088A
1497 0088A      *
1498 0088A      * C(A) := amount to check
1499 0088A
1500 0088A 8F00000      GOSBVL =MEMCAL
1501 00891 560          GONC R120 pas d'erreur
1502 00894 684F          GOTO Rerr "Insufficient Memory"
1503 00898
1504 00898      *
1505 00898      * B(A) = length in nibs
1506 00898      * D0 = " début ligne LIF
1507 00898 132          R120 AD0EX A(A) := " début ligne LIF
1508 008A2 146          D0=(5) =OUTBS
1509 008A5          C=DAT0 A C(A) := " début OUTBS
1510 008A5
1511 008A5      *
1512 008A5      * A(A) = source address (" ligne LIF)
1513 008A5      * C(A) = dest address (OUTBS)
1514 008A5      * B(A) = length in nibs (longueur LIF)
1515 008A5
1516 008A5 8F00000      GOSBVL =MOVE+M
1517 008AC C9          C=C-B A
1518 008AE 1B00000      D0=(5) =AVMEMS
1519 008B5 144          DAT0=C A AVMEMS := " fin du buffer
1520 008B8
1521 008B8      *
1522 008B8      * RSTK(1) = LEN (occurrence)
1523 008B8      * RSTK(0) = offset occurrence
1524 008B8      * savlin + 0 = no ligne trouvée
1525 008B8      * savlin + 5 = " ligne trouvée
1526 008B8      * Préparer les registres pour REPLIN
1527 008BF 1F00000      D1=(5) remp
1528 008C3 10B        C=DAT1 10
1529 008C6          R3=C
1530 008C6 1E0000      D1=(4) =OUTBS
1531 008CC 143          A=DAT1 A A(A) := OUTBS
1532 008CF 07          C=RSTK C(A) := offset de l'occurrence
1533 008D1 C2          C=C+A A C(A) := occurrence
1534 008D3 134          D0=C D0 := occurrence

```

Page 036 Editeur, execution des commandes (xcmd.as)  
AREUH ASS. V2.4 Commande Replace

```

1535 0091F AAC          GOC rerr
1536 00922
1537 00922      *
1538 00922      * Préparation de l'appel à RPLIN
1539 00922
1540 00922 1F00000      D1=(5) 5*savlin
1541 00929 147          C=DAT1 A
1542 0092C 134          D0=C D0 := " ancienne ligne
1543 0092F 8E0000      GOSUBL =LIFlen
1544 00935 103          R3=A R3 := length of old line in nibs
1545 00938 D6          C=A A
1546 0093A 06          RSTK=C RSTK := " " "
1547 0093C 136          CD0EX
1548 0093F CA          A=A+C A A(A) := " last nib + 1 of old line
1549 00941
1550 00941 1B00000      D0=(5) =FILADR
1551 00948 146          C=DAT0 A C(A) := " file header
1552 0094B
1553 0094B      *
1554 0094B      * (OUTBS..AVMEMS) = ligne à remplacer
1555 0094B      * A(A) = " address of last nibble + 1 of old line
1556 0094B      * C(A) = " address of file header
1557 0094B      * R3(A) = length of old line in nibs
1558 0094B
1559 0094B 8F00000      GOSBVL =MGOSUB EDIT peut être derrière le fichier
1560 00952 00000      CON(5) =RPLIN
1561 00957 423          GOC rerr Il y a eu erreur. On libère le buf
1562 0095A
1563 0095A      *
1564 0095A      * RSTK = longueur vieille ligne
1565 0095A      * B(A) = longueur de la nouvelle ligne
1566 0095A      * output buffer collapsed
1567 0095A
1568 0095A      *
1569 0095A      * Avant tout, récupérer l'adresse du buffer dans lequel
1570 0095A      * est stocké le tableau de pointeurs sur les lignes.
1571 0095A
1572 0095A      *
1573 0095A      * GOSUBL =synctf ne touche pas à B(A)
1574 0095A
1575 0095A      *
1576 0095A      * Ensuite, actualiser ce fameux tableau.
1577 0095A
1578 0095A      *
1579 0095A      *
1580 0095A      *
1581 0095A      *
1582 0095A      *
1583 0095A      *
1584 0095A      *
1584 0095A 07          C=RSTK C(A) := ancienne longueur
1584 0095A 146          A=B A A(A) := nouvelle longueur
1584 0095A EA          A=A-C A A(A) := (nouvelle - ancienne)
1585 00966
1586 00966 1B00000      D0=(5) savlin savlin + 0 = numéro de la ligne
1587 0096D 146          C=DAT0 A
1588 00970 D5          B=C A B(A) := numéro de la ligne remplacée
1589 00972
1590 00972      *
1591 00972      * Réactualisation du tableau de pointeurs :
1592 00972      * B(A) = numéro de la ligne remplacée
1593 00972      * A(A) = longueur nouvelle - longueur ancienne
1594 00972
1595 00972      *
1596 00972      * GOSUBL =rpllin attention : Ce n'est pas RPLIN !!!

```

Page 037  
AREUH ASS, V2.4

Editeur, execution des commandes (<xcmd.as>  
Commande Replace

```

1635 00978
1636 00978
1637 00978
1638 00978 1800000
1639 00978 15E7
1640 00983 10E
1641 00986
1642 00986 8E0000
1643 0098C 570
1644 0098F 8C0000
1645 00995
1646 00995
1647 00995
1648 00995
1649 00995 1800000
1650 0099C 14E
1651 0099F 140000
1652 009A5 144
1653 009A8
1654 009A8
1655 009A8
1656 009A8
1657 009A8
1658 009A8
1659 009A8
1660 009A8 1F00000
1661 009AF AEC
1662 009B0 1570
1663 009B6 810E
1664 009B9 90A
1665 009BC R1
1666 009BE
1667 009BE
1668 009BE
1669 009BE 1800000
1670 009C5 14E
1671 009C8 E6
1672 009CA 140000
1673 009C9 144
1674 009D3 696E
1675 009D7
1676 009E7 8C0000
1677 009E0
1678 009E0
1679 009E0
1680 009E0
1681 009E0 1E00000
1682 009E4 14E
1683 009E7 84A
1684 009E4 E0

```

- On a fini RPLIN. On peut à présent restaurer R0
- D0=(5) =FUNCRO
- C=DAT0 S restaurer R0(7-0)
- R0=C
- GOSUBL =UPDTR0 mettre à jour R0(7-3) après le move
- GONC R100 pas d'erreur
- GOLONG =vederr System Error
- On a modifié la ligne. C'est donc potentiellement la
- ligne courante finale.
- R100 D0=(5) savlin
- C=DAT0 A ((A) := numéro de cette ligne
- D0=(4) dern
- DATA=C A dern := dernière ligne modifiée
- Le remplacement est fini.
- RC00
- Il faut maintenant décider si on continue ou si on arrête.
- D1=(5) =IMP3
- C=0 B C(B) := 0
- C=DAT1 P C(B) := code de retour (0-1 ou 2-3)
- C=SRB ((A) := code de retour (0 ou 1)
- ?C=0 P
- GOYES R300 [0]
- On continue.
- D0=(5) savlin
- C=DAT0 A C(A) := ligne qu'on vient de traiter
- C=C+1 A
- D0=(4) ldeb
- DATA=C A ldeb := dernière ligne traitée + 1
- GOTO R100
- R300 GOSUBL =ENDPOS
- Si on n'a rien trouvé ou rien modifié, alors "Not Found"
- Tout ce se résume par "dern # 0 ?"
- D0=(5) dern
- C=DAT0 A
- ?C=0 A
- GOYES NotFnd "Not Found"

Page 039  
AREUH ASS, V2.4

Editeur, execution des commandes (<xcmd.as>  
Commande Help

```

1698 009FC
1699 009FC
1700 009FC
1701 009FC
1702 009FC
1703 009FC
1704 009FC
1705 009FC
1706 009FC
1707 009FC
1708 009FC
1709 009FC
1710 009FC
1711 009FC
1712 009FC
1713 009FC
1714 009FC
1715 009FC
1716 009FC
1717 009FC
1718 009FC
1719 009FC
1720 009FC
1721 009FC 8E3D7F
1722 00A02 8E0000
1723 00A03 404
1724 00A0B 3183
1725 00A0F 962
1726 00A12 73
1727 00A14
1728 00A14
1729 00A14
1730 00A14
1731 00A14 8E0000
1732 00A1A AE8
1733 00A1D 8EB67F
1734 00A23 AE4
1735 00A26
1736 00A26
1737 00A26
1738 00A26 8E0000
1739 00A2C 58C
1740 00A2F
1741 00A2F
1742 00A2F
1743 00A2F
1744 00A2F 330000
1745 00A35 CB
1746 00A37
1747 00A37 8E0000

```

- cmHELP
- But: afficher une aide
- Entrée:
- D(A) = LEN(param)
- D1 = param
- R0 = nb de lignes introduites
- R1 : ldebut
- R2 : lfin
- R3(S) = 1 si '?' avant la commande
- Sortie: par BOUCLE
- Abime: tout
- Appelle: noquer, lookah, getchr, valide, poscmd, dspmsg, KEYWT, dspmsg, RPTKY, FINDA
- Niveaux: N/A
- Historique:
- 88/10/23: PD/JT conception & codage
- 88/10/29: PD/JT ajout de la repetition
- 88/10/29: PD/JT ajout du paramètre de Help
- cmHELP GOSUBL noquer
- GOSUBL =lookah
- GOC H500 "help" tout seul (tous les sujets)
- LC(2) ':'
- ?A=C B
- GOYES H500
- Help avec un sujet
- A(B) = le sujet (à vérifier)
- GOSUBL =getchr
- B=A B B(B) := caractère lu
- GOSUBL valide n'abime pas B(B)
- A=B B A(B) := caractère lu
- A(B) = le sujet
- GOSUBL =poscmd
- GONC NotFnd "Not Found"
- Commande trouvée:
- C(A) = D(A) = index de la commande.
- LC(4) (=id)"(=teHL00)
- C=C+D A C(A) := numéro du message
- GOSUBL =dspmsg

Page 038  
AREUH ASS, V2.4

Editeur, execution des commandes (<xcmd.as>  
Commande Replace

```

1685 009EC
1686 009EC
1687 009EC
1688 009EC
1689 009EC 8E0DEF
1690 009FC
1691 009FC
1692 009FC
1693 009F2 8C0000
1694 009F6
1695 009F6 662D
1696 009FC

```

- Tout c'est donc bien passé. Reste à changer la notion
- de "ligne courante"
- GOSUBL =getcou
- Et voilà. Simple, non ?
- GOLONG =BOUCLE
- NotFnd GOTO NotFnd

Page 040  
AREUH ASS, V2.4

Editeur, execution des commandes (<xcmd.as>  
Commande Help

```

1748 00A30 8E0000
1749 00A43
1750 00A43 8C0000
1751 00A49
1752 00A49
1753 00A49
1754 00A49
1755 00A49
1756 00A49 8E37F
1757 00A4F 1800000
1758 00A56 AC2
1759 00A59 1544
1760 00A5D
1761 00A5D 3400000
1762 00A54 D0
1763 00A66 1800000
1764 00A6D 1520
1765 00A71 C2
1766 00A73 8E0000
1767 00A79
1768 00A79
1769 00A79
1770 00A79
1771 00A79
1772 00A79
1773 00A79
1774 00A79
1775 00A79
1776 00A79 8F00000
1777 00A80 570
1778 00A83 D4
1779 00A85 480
1780 00A88 8E0000
1781 00A8E
1782 00A8E
1783 00A8E
1784 00A8E
1785 00A8E 1800000
1786 00A95 1560
1787 00A99 D5
1788 00A9B
1789 00A9B 8F00000
1790 00AA2 23
1791 00AA4 A10
1792 00AA7 03
1793 00AA9 020
1794 00AAC 2A
1795 00AAE 920
1796 00AB1 3A
1797 00AB3 820

```

- GOSUBL =KEYWT
- GOLONG =BOUCLE
- L'utilisateur a demandé une aide générale. On va donc afficher les messages teHL00 à teHL11.
- H500 GOSUBL valide
- D0=(5) curcmd
- C=0 S
- DATA=C S curcmd := 0
- H520 LC(5) (=id)"(=teHL00)
- A=0 A
- D0=(5) curcmd
- A=DAT0 P
- C=C+A A C(A) := teHL00 + curcmd
- GOSUBL =dspmsg afficher le message
- Attente de l'appui sur une touche
- if (key already pressed) /\* repeat \*/
- then get key
- else wait key
- fi
- GOSBVL =RPTKY Y-a-t-il une touche en répétition ?
- GONC H525 Non : il faut utiliser KEYWAIT5
- A=B A Oui : A(A) = touche
- GOC H530 B.E.T.
- H525 GOSUBL =KEYWT Attendre. A(A) = B(A) = touche
- H530
- A(B) = touche pressée
- D0=(5) curcmd
- C=DAT0 P
- B=C A B(0) := curcmd
- GOSBVL =FINDA n'abime ni B(A)
- CON(2) #32 [""]
- REL(3) Hup
- CON(2) #33 [v]
- REL(3) Hdown
- CON(2) #32+112 [g]l[""]
- REL(3) Htop
- CON(2) #33+112 [g]lv
- REL(3) Hbot

Page 041 Editeur, exécution des commandes <xcmd.as>  
AREUH ASS. V2.4 Commande Help

```

1798 00A86 00          NIBHEX 00
1799 00A88
1800 00A88          * Touche normale. Retour à la ligne de commande
1801 00A88
1802 00A88 3C0000          COLONG =BOUCLE
1803 00A8C
1804 00A8E 909          Hup  ?B=0 P
1805 00AC1 32          GOVES H600
1806 00AC3 A0D          B=B-1 P
1807 00AC6 5D1          GONC H600
1808 00AC9 300          Hdown LC(1) =NCHDED
1809 00ACC 989          ?B=C P
1810 00ACF 51          GOVES H600
1811 00AD1 B05          B=B+1 P
1812 00ADA 5F0          GONC H600 B.E.T.
1813 00AD7 A81          Htop  B=0 P
1814 00ADA 6900         GOTO H600
1815 00ADE 300          Hbot  LC(1) =NCHDED
1816 00AE1 A85          B=C P
1817 00AE4
1818 00AE4          * B(0) = nouvelle commande courante
1819 00AE4
1820 00AE4 A84          H600 A=B P A(0) := nouvelle commande courante
1821 00AE7 1B00000         D0=(5) curcmd
1822 00AEE 1500          DAT0=A P
1823 00AF2 6A6F          GOTO H520
1824 00AF6

```

Page 043 Editeur, exécution des commandes <xcmd.as>  
AREUH ASS. V2.4 Commande Join

```

1876 00B1F          * R3 = nombre de lignes à joindre (déjà testé)
1877 00B1F
1878 00B1F 8E96FF          *
1879 00B25          GOSUBL valide
1880 00B25
1881 00B25          *
1882 00B25          * Test de l'accessibilité
1883 00B25          * Changement du 17/06/89 : test du flag "read-only"
1884 00B25 1F00000         D1=(5) =RDONLY
1885 00B2C 1574          C=DAT1 S
1886 00B30 8E0000         GOSUBL =WRITE?
1887 00B36 537          GONC Jerr C(A) = eFACCS, eFPROT ou eFOPEN
1888 00B39
1889 00B39 1B00000         D0=(5) remain
1890 00B40 118          C=R3
1891 00B43 144          DAT0=C A remain := nombre de lignes restantes
1892 00B46 1900         D0=(2) prem
1893 00B4A 119          C=R1
1894 00B4D 144          DAT0=C A prem := numéro de la ligne de début
1895 00B50
1896 00B50          * C(A) := numéro de la ligne à chercher
1897 00B50
1898 00B50 8E0000         GOSUBL =seek abime tous les registres et THP5
1899 00B56
1900 00B56          * D0 = début de la ligne
1901 00B56
1902 00B56 136          CDREX C(A) := première ligne
1903 00B59 108          R0=C R0(A3) := première ligne
1904 00B5C 109          R1=C R1(A) := ligne courante
1905 00B5F
1906 00B5F 1B00000         D0=(5) remain
1907 00B66 146          C=DAT0 A
1908 00B69 108          R3=C R3(A) := nombre de lignes restantes
1909 00B6C
1910 00B6C          * Préparer le buffer
1911 00B6C
1912 00B6C 1B00000         D0=(5) =OUTBS
1913 00B73 142          A=DAT0 A
1914 00B76 D2          C=0 A
1915 00B78 304          LC(1) 4
1916 00B7B CA          A=A+C A
1917 00B7D 164          D0=D0+ 5 D0=(5) =AVHEMS
1918 00B80 140          DAT0=A A
1919 00B83
1920 00B83
1921 00B83          *
1922 00B83          * Invariant de boucle
1923 00B83          * (OUTBS...OUTBS+3) = place pour la longueur LIF
1924 00B83          * (OUTBS+4...AVHEMS) = lignes déjà écrites
1925 00B83          * R3 = nombre de lignes restant à joindre
1926 00B83          * R1 = ligne à joindre

```

Page 042 Editeur, exécution des commandes <xcmd.as>  
AREUH ASS. V2.4 Commande Join

```

1826 00AF6
1827 00AF6          *
1828 00AF6          * cmJOIN
1829 00AF6
1830 00AF6          * But: joindre deux (ou plus) lignes
1831 00AF6          * Entrée:
1832 00AF6          * - D(A) = LEN(param)
1833 00AF6          * - D1 = param
1834 00AF6          * - R0 = nb de lignes introduites
1835 00AF6          * R1 : ldebut
1836 00AF6          * R2 : lfin
1837 00AF6          * R3(S) = 1 si '?' avant la commande
1838 00AF6          * Sortie: par BOUCLE
1839 00AF6          * Abime: tout
1840 00AF6          * Appelle: quer?, abCn, WRITE?, valide, seek, LIFlen,
1841 00AF6          * HENCKL, MOVE+M, FHTLIF, query, TBLJMC, RPLLIN, dellin,
1842 00AF6          * setcou
1843 00AF6          * Niveaux: N/A
1844 00AF6          * Historique:
1845 00AF6          * 89/10/23: PD/JT documentation
1846 00AF6          * 89/10/29: PD/JT conception & codage
1847 00AF6          * 89/06/17: PD/JT test du flag "read-only"
1848 00AF6          * 89/05/07: PD/JT valeur "fin" par défaut si 1 argument
1849 00AF6          *
1850 00AF6          *
1851 00AF6 8EDEGF          *cmJOIN GOSUBL quer? QUERY := R3(S)
1852 00AFC
1853 00AFC          * Analyse une commande de la forme :
1854 00AFC          * [<ligne> [., <ligne>]] J [<nombre>]
1855 00AFC
1856 00AFC
1857 00AFC
1858 00AFC          * Correction du 89/08/07 : s'il y a un argument, alors
1859 00AFC          * la valeur par défaut de "fin" = "début" + 1
1860 00AFC
1861 00AFC 110          A=R0
1862 00AFF 8A8          ?A=0 A
1863 00B02 80          GOVES J050
1864 00B04 111          A=R1 A(A) := ldebut
1865 00B07 5C0          GONC J050 B.E.T.
1866 00B0A
1867 00B0A          * Fin de la correction du 89/08/07.
1868 00B0A
1869 00B0A 1B00000         J050 D0=(5) =COUR
1870 00B11 142          A=DAT0 A A(A) = .
1871 00B14 E4          J050 A=A+1 A lfin = .+1 par défaut, ou ldebut+1
1872 00B16 850          ST=1 0 cas non standard
1873 00B19 8E7F6F         GOSUBL abCn
1874 00B1F
1875 00B1F          * R1 = ligne de début

```

Page 044 Editeur, exécution des commandes <xcmd.as>  
AREUH ASS. V2.4 Commande Join

```

1926 00B83          *
1927 00B83          * prem = numéro de la première ligne
1928 00B83          * R0 = première ligne
1929 00B83 118          J500 C=R3
1930 00B86 CE          C=C-1 A
1931 00B88 560          GONC J510
1932 00B8B 6821         GOTO J600 sortie de la boucle
1933 00B8F 108          J510 R3=C
1934 00B92
1935 00B92          *
1936 00B92          * A-t-on assei de place ?
1937 00B92
1938 00B92 119          C=R1
1939 00B95 134          D0=C
1940 00B9E 66          GOSUBL =LIFlen C(A) := longueur en octets
1941 00BA0 8F00000         C=C+C A C(A) := ... en quartets (sans long)
1942 00BA7 580          GOSBVL =MEMCKL
1943 00BAA          GONC J520 Pas d'erreur
1944 00BAA          * C(A) = "Insufficient Memory"
1945 00BAA 8C0000         Jerr COLONG =xerror
1946 00BB0
1947 00BB0          J520
1948 00BB0
1949 00BB0          *
1950 00BB0          * D0 = début ligne LIF
1951 00BB0
1952 00BB0          *
1953 00BB0          * Préparer les paramètres pour MOVE+M
1954 00BB0          * source = ligne courante
1955 00BB0          * longueur = 2*LEN (ligne courante)
1956 00BB0          * dest = AVHEMS
1957 00BB0 8E0000         GOSUBL =LIFlen C(A) := longueur en octets
1958 00BB6 102          R2=A sauvegarder la longueur totale
1959 00BB9 C6          C=C+C A C(A) := longueur en nibs
1960 00BBB D5          B=C A B(A) := block length
1961 00BB0
1962 00BB0 163          D0=D0+ 4 sauter la longueur LIF
1963 00BB3          AD0EX A(A) := source address
1964 00BB3 1B00000         D0=(5) =AVHEMS
1965 00BBA 146          C=DAT0 A C(A) := dest address
1966 00BCC
1967 00BCC          *
1968 00BCC          * A(A) = source address
1969 00BCC          * B(A) = length of block in nibs
1970 00BCC          * C(A) = dest address
1971 00BCC 8F00000         GOSBVL =MOVE+M abime D0 et D1
1972 00BD4
1973 00BD4          *
1974 00BD4          * Ajouter un espace et mettre à jour AVHEMS
1975 00BD4 C9          C=C+B A C(A) := nouveau AVHEMS

```

```

1976 00B06 134      D0=C      *(AVMEMS)*= *
1977 00B09 3102    LCASC     *
1978 00B00 140     DAT0=C B
1979 00B00 161     D0=D0 *
1980 00B03 136     C00EX
1981 00B06 1B00000  D0=(5) =AVMEMS
1982 00B0D 142     A-DAT0 A A(A) := ancien AVMEMS
1983 00B0F 102     R2=A     R2(A) := ancien AVMEMS
1984 00B0F 144     DAT0=C A AVMEMS := nouveau AVMEMS
1985 00B0F
1986 00B0F
1987 00B0F
1988 00B0F 1B00000  D0=(5) =QUERY
1989 00B0D 1564    C-DAT0 S
1990 00C01 94A     TC=0 S
1991 00C04 77      GOYES J500 Non
1992 00C06
1993 00C06
1994 00C06
1995 00C06 11B     C=R3
1996 00C09 E6     C=C-1 A
1997 00C0B 1A0000  D0=(4) remain
1998 00C11 142     A-DAT0 A
1999 00C14 8A2     7A=C A première jointure ?
2000 00C17 46     GOYES J500 out : alors mode normal
2001 00C19
2002 00C19
2003 00C19
2004 00C19
2005 00C19 119     C=R1
2006 00C1C 1B00000  D0=(5) Jsave
2007 00C23 144     DAT0=C A Jsave* := R1(A)
2008 00C26
2009 00C26 840     ST0=0
2010 00C29 8E00000  GOSUBL =FHTLIF Mettre une longueur LIF plausible
2011 00C2F 436     GOC JerR "Line Too Long"
2012 00C30
2013 00C32 112     A=R2
2014 00C35 4F2     C=0 W
2015 00C38 1B00000  D0=(5) =OUTBS
2016 00C3F 146     C-DAT0 A C(A) := début de la ligne (LIF)
2017 00C42 134     D0=C     D0 := début ligne
2018 00C45 EE     C=A-C A C(A) := long. jusqu'avant la ligne
2019 00C47 81E     CSR0
2020 00C4A CE     C=C-1 A enlever la longueur LIF
2021 00C4C CE     C=C-1 A C(A) := no de colonne
2022 00C4E
2023 00C4E JF00000  D1=(5) prem
2024 00C55 143     A-DAT1 A A(A) := no première ligne
2025 00C58

```

```

2076 00CB7
2077 00CB7
2078 00CB7
2079 00CB7
2080 00CB7
2081 00CB7
2082 00CB7
2083 00CB7
2084 00CB7
2085 00CB7
2086 00CB7
2087 00CB7
2088 00CB7
2089 00CB7 1B00000  D0=(5) =AVMEMS
2090 00CBE 146     C-DAT0 A
2091 00CC1 CE     C=C-1 A
2092 00CC3 CE     C=C-1 A
2093 00CC5 144     DAT0=C A AVMEMS := 2
2094 00CC8
2095 00CC8
2096 00CC8
2097 00CC8 850     ST=1 0 ajuster AVMEMS si besoin est
2098 00CCB 8E00000  GOSUBL =FHTLIF
2099 00CD1 41C     GOC JerR "Line Too Long"
2100 00CD4
2101 00CD4
2102 00CD4
2103 00CD4
2104 00CD4
2105 00CD4
2106 00CD4
2107 00CD4
2108 00CD4
2109 00CD4
2110 00CD4
2111 00CD4
2112 00CD4
2113 00CD4
2114 00CD4
2115 00CD4
2116 00CD4
2117 00CD4
2118 00CD4 1B00000  D0=(5) =OUTBS
2119 00CDB 146     C-DAT0 A
2120 00CDE 134     D0=C
2121 00CE1 8E00000  GOSUBL =LIFlen A(A) := longueur en nibs
2122 00CE7 D8     B=A A B(A) := LIFlen(OUTBS)
2123 00CE9 111     A=R1 A(A) := fin
2124 00CEC 118     C=R0 C(A) := ligne début
2125 00CEF EA     A=A-C A A(A) := tout ce qu'on a lu

```

• Fin de la boucle de recopie des lignes  
 • J600  
 • (OUTBS...OUTBS+3) = place pour la longueur LIF  
 • (OUTBS+4..AVMEMS) = somme des lignes jointes  
 • R0 = première ligne dans le fichier  
 • R1 = ligne juste après la dernière dans le fichier  
 • prem = numéro de la première ligne  
 • remain = nombre initial de lignes à joindre  
 • Enlever le dernier espace ajouté  
 • Mettre en place la longueur LIF  
 • ST=1 0 ajuster AVMEMS si besoin est  
 • GOSUBL =FHTLIF  
 • GOC JerR "Line Too Long"  
 • \*\*\*\*\*  
 • Maintenant, les choses sont en place. Les anciennes lignes  
 • sont toujours dans le fichier, et la nouvelle ligne est  
 • en (OUTBS..AVMEMS) bien formattée.  
 • R0 = première ligne dans le fichier  
 • R1 = première ligne après la dernière jointe  
 • R2 = ?  
 • R3 = 0  
 • prem = numéro de la première ligne  
 • remain = nb de lignes jointes (non comprise la première)  
 • (OUTBS..AVMEMS) = nouvelle ligne  
 • \*\*\*\*\*  
 • I - Suppression des (remain) lignes jointes dans le  
 • tableau de pointeurs.  
 • Suppression des lignes (prem-1, prem+1+remain-1)  
 • On a enlevé (R1-R0)-LIFlen(OUTBS) nibs  
 • \*\*\*\*\*  
 • D0=(5) =OUTBS  
 • C-DAT0 A  
 • D0=C  
 • GOSUBL =LIFlen A(A) := longueur en nibs  
 • B=A A B(A) := LIFlen(OUTBS)  
 • A=R1 A(A) := fin  
 • C=R0 C(A) := ligne début  
 • A=A-C A A(A) := tout ce qu'on a lu

```

2026 00C58 8E00000  GOSUBL =query
2027 00C5E
2028 00C5E 1F00000  D1=(5) Jsave
2029 00C65 143     A-DAT1 A
2030 00C68 101     R1=A R1 := ancienne valeur sauvegardée
2031 00C6E
2032 00C6B 8F00000  GOSUBL =RELJMC
2033 00C72 520     REL(3) J500 [ATTN] ou [O] ou [F][OFF]
2034 00C75 6A0     REL(3) J500 [Y]
2035 00C78 F10     REL(3) J500 [N]
2036 00C7B
2037 00C7B J500
2038 00C7B
2039 00C7B
2040 00C7B
2041 00C7B 119     C=R1 C(A) := ligne courante
2042 00C7E 134     D0=C
2043 00C81 8E00000  GOSUBL =LIFlen A(A) := longueur totale
2044 00C87 136     C00EX C(A) := ligne courante
2045 00C8A C2     C=C+A A C(A) := ligne suivante
2046 00C8C 109     R1=C
2047 00C8F 63FE     GOTO J500
2048 00C93
2049 00C93 661F     JerR GOTO JerR
2050 00C97
2051 00C97
2052 00C97
2053 00C97
2054 00C97
2055 00C97
2056 00C97
2057 00C97
2058 00C97
2059 00C97
2060 00C97
2061 00C97 11A     J500 C=R2 remettre AVMEMS à la valeur avant
2062 00C9A 1B00000  D0=(5) =AVMEMS
2063 00CA1 144     DAT0=C A AVMEMS := valeur avant
2064 00CA4
2065 00CA4 1A00000  D0=(4) remain D0 := nombre de lignes initial
2066 00CA4 142     A-DAT0 A A(A) := nombre de lignes initial
2067 00CA0 11B     C=R3
2068 00CA0 EA     A=A-C A
2069 00CB2 CC     A=A-1 A
2070 00CB4 140     DAT0=A A remain = initial - traitées
2071 00CB7
2072 00CB7
2073 00CB7
2074 00CB7
2075 00CB7

```

```

2126 00CF1 E0     A=A-B A A(A) := longueur de ce qu'on enlevé
2127 00CF3
2128 00CF3 1B00000  D0=(5) prem
2129 00CFA 146     C-DAT0 A
2130 00CFD D5     B=C A
2131 00CFE E5     B=B-1 A B(A) := no de l'ancienne prem. ligne
2132 00DA1
2133 00DA1 1B00000  D0=(5) remain
2134 00DA8 146     C-DAT0 A C(A) := nombre de lignes enlevées
2135 00D0B CE     C=C-1 A
2136 00D0D
2137 00D0D 8E00000  GOSUBL =dellin
2138 00D13
2139 00D13
2140 00D13
2141 00D13
2142 00D13 111     A=R1 A(A) := ligne suivante
2143 00D16 118     C=R0 C(A) := première ligne
2144 00D19 EE     C=A-C A C(A) := length of old line
2145 00D1B 10B     R3=C R3(A) := length of old line
2146 00D1E 1B00000  D0=(5) =FILADR
2147 00D25 146     C-DAT0 A C(A) := file header
2148 00D28
2149 00D28
2150 00D28
2151 00D28
2152 00D28
2153 00D28
2154 00D28 8F00000  GOSUBL =MGOSUB
2155 00D2F 00000  CON(5) =RPLLIN
2156 00D34
2157 00D34
2158 00D34
2159 00D34
2160 00D34
2161 00D34
2162 00D34
2163 00D34 8E00000  GOSUBL =syncbf
2164 00D3A
2165 00D3A
2166 00D3A
2167 00D3A
2168 00D3A 1B00000  D0=(5) prem
2169 00D41 146     C-DAT0 A
2170 00D44 8E273F  GOSUBL setcou
2171 00D4A 8C0000  GOLONG =BOUCLE
2172 00D50

```

• A(A) = " last nibble + 1 of old line  
 • C(A) = " file header  
 • R3(A) = length of old line (in nibs)  
 • (OUTBS..AVMEMS) = new line  
 • \*\*\*\*\*  
 • GOSUBL =MGOSUB  
 • CON(5) =RPLLIN  
 • \*\*\*\*\*  
 • On ne peut ici avoir que deux erreurs :  
 • "Illegal Access" : déjà testé par WRITE?  
 • "Insufficient Memory" : on ne fait que tasser  
 • Donc on ne peut avoir d'erreur...  
 • Le branchement "GOC JerR" est inutile  
 • \*\*\*\*\*  
 • GOSUBL =syncbf  
 • \*\*\*\*\*  
 • Ligne courante := la nouvelle ligne  
 • \*\*\*\*\*  
 • D0=(5) prem  
 • C-DAT0 A  
 • GOSUBL setcou  
 • GOLONG =BOUCLE

```

2174 00050 .....
2175 00050 * cmDEL
2176 00050 *
2177 00050 *
2178 00050 * But: effacer une ou plusieurs ligne dans le fichier
2179 00050 * Entrée:
2180 00050 * - D(A) = LEN(param)
2181 00050 * - D1 = param
2182 00050 * - R0 = nb de lignes introduites
2183 00050 * - R1 = ldebut
2184 00050 * - R2 = lfin
2185 00050 * - R3(S) = 1 si '?' avant la commande
2186 00050 * Sortie: par BOUCLE
2187 00050 * Abime: tout
2188 00050 * Appelle: noquer, set.db, GETFIL, dspmsg, KEYWT, valid?.
2189 00050 * FINDA, seek, syncbf, dellin, OBCOLL, RPLLIN, MVFIL.
2190 00050 * setcou, getdbf, svRIR2, extsek
2191 00050 * Niveaux: N/A
2192 00050 * Algorithmes:
2193 00050 * analyser la ligne de commande
2194 00050 * tester l'accessibilité aux fichiers
2195 00050 * demander confirmation si plus d'une ligne
2196 00050 * si destruction interne
2197 00050 * alors détruire
2198 00050 * sinon
2199 00050 * chercher la fin du fichier externe
2200 00050 * déplacer le bloc dans le fichier externe
2201 00050 * DERN = nombre de lignes détruites
2202 00050 * fin si
2203 00050 * mise à jour du buffer (en modifiant DERN)
2204 00050 * Historique:
2205 00050 * 88/10/31: PD/JT conception & codage
2206 00050 * 88/11/12: PD/JT amélioration du traitement d'erreur
2207 00050 * 88/11/12: PD/JT séparation de set.db
2208 00050 * 88/11/12: PD/JT séparation de getdbf
2209 00050 * 88/11/13: PD/JT séparation de svRIR2
2210 00050 * 88/11/13: PD/JT séparation de extsek
2211 00050 * 88/11/13: PD/JT utilisation de YNO
2212 00050 * 89/06/17: PD/JT test du flag "read-only"
2213 00050 .....
2214 00050 *cmDEL
2215 00050 .....
2216 00050 * Analyse de la ligne de commande
2217 00050 .....
2218 00050 * GOSUBL noquer interdit le '?'
2219 00050 REF74F
2220 00050
2221 00050
2222 00050 * Mettre des paramètres par défaut et tester la validité
2223 00050

```

```

2274 000AF 8E0000 Derr GOSUBL =PURGE n'abime pas C(A)
2275 000B5 8C0000 ERREUR GOLONG =xederr
2276 000BE .....
2277 000BE *
2278 000BB * Demander confirmation
2279 000BB .....
2280 000BB
2281 000BB 1B0000 D060 D0=(5) ldeb
2282 000C2 142 A=DAT0 A A(A) := ldeb
2283 000C5 164 D0=D0+ 5 D0=(5) lfin
2284 000C8 146 C=DAT0 A C(A) := lfin
2285 000FB 8A2 ?A+C A
2286 000CE 93 GOYES D100 détruire sans confirmation
2287 000D0
2288 000D0 *
2289 000D0 * Poser la question à l'utilisateur
2290 000D0
2291 000D0 LC(4) (sid)="+deLE" "Ok to Delete? Y/N:"
2292 000D6 8E0000 GOSUBL =dspmsg
2293 000DC 8E0000 GOSUBL =KEYWT
2294 000E2
2295 000E2 * Cherche la signification de la touche. Eventuellement,
2296 000E2 * autorise la traduction par un traducteur de message.
2297 000E2 8E0000 GOSUBL =YNO
2298 000E8
2299 000E8 * C(0) = 0 si non reconnu, 1 si Y, 2 si N, 3 si 0
2300 000E8
2301 000E8 8F0000 GOSBVL =TBLJMC
2302 000EF C00 REL(3) D070 non reconnu
2303 000F2 510 REL(3) D100 [Y]
2304 000F5 600 REL(3) D070 [N]
2305 000F8 300 REL(3) D070 [0]
2306 000FB
2307 000FB *
2308 000FB * Ouf ! L'utilisateur s'est aperçu de sa méprise...
2309 000FB * Il faut effacer le fichier si on l'a créé spécialement
2310 000FB * pour l'occasion.
2311 000FB 8E0000 D070 GOSUBL =PURGE
2312 000E1 8C0000 GOLONG =BOUCLE pas de destruction
2313 000E7
2314 000E7 *
2315 000E7 * Suppression acceptée par l'utilisateur
2316 000E7
2317 000E7 .....
2318 000E7 * Determination des adresses
2319 000E7 .....
2320 000E7
2321 000E7
2322 000E7
2323 000E7

```

```

2224 00056 8EC83F GOSUBL set.db (., "ligne de début")
2225 00056
2226 00056 *
2227 00056 * Sauver les paramètres de D :
2228 00056 * R1 = ldebut
2229 00056 * R2 = lfin
2230 00056
2231 00056 8EE92F GOSUBL svRIR2
2232 00062
2233 00062 * Deux cas possibles pour le fichier :
2234 00062 * existe n'existe pas
2235 00062 * D (fichier) erreur creation
2236 00062 * D (fichier) + chercher creation
2237 00062 850 ST=1 =+DELETE cas 'D' let pas 'C'/'M'
2238 00065 840 ST=0 =+XCHG cas 'D' let pas 'X'
2239 00068 8E0000 GOSUBL =GETFIL analyse et cherche le fichier
2240 0006E 464 GOC ERREUR heap !
2241 00071
2242 00071 *
2243 00071 * Flag sCREAT = 1 si le fichier a dû être créé
2244 00071
2245 00071
2246 00071 *
2247 00071 * Valider ce qui a été reconnu.
2248 00071 8E014F GOSUBL valid? analyse de la ligne terminée
2249 00077 473 GOC Derr Erreur ! Il faut revenir en arrière
2250 0007A
2251 0007A *
2252 0007A * Tester l'accès au(x) fichier(s)
2253 0007A * Changement du 17/06/89 : test du flag "read-only"
2254 0007A
2255 0007A 1F00000 D1=(5) =RDONLY
2256 00061 1574 C=DAT1 S
2257 00085 8E0000 GOSUBL =WRITE?
2258 00088 532 GONC Derr C(A) = eFACCS, eFPROT ou eFOPEN
2259 0008E
2260 0008E 1F00000 D1=(5) =EXTFIL
2261 00095 147 C=DAT1 A
2262 00098 8AA ?C=0 A
2263 00098 02 GOYES D060 Pas de fichier extérieur
2264 00090 8E0000 GOSUBL =CHKWRT
2265 000A3 8E0000 GOSUBL =WRITE?
2266 000A9 550 GONC Derr C(A) = eFACCS, eFPROT ou eFOPEN
2267 000AC
2268 000AC 4E0 GOC D060 B.E.T. (pas d'erreur)
2269 000AF
2270 000AF
2271 000AF *
2272 000AF * Commande non reconnue. Il faut donc revenir en arrière et
2273 000AF * détruire éventuellement le fichier.

```

```

2324 000E7
2325 000E7 * Calcule addeb, adfin: findeb et nbllg à partir de
2326 000E7 * ldeb et lfin.
2327 000E7 * ldeb = numéro de la première ligne à détruire
2328 000E7 * lfin = numéro de la dernière ligne à détruire
2329 000E7
2330 000E7 8EB02F GOSUBL getdbf
2331 000E0
2332 000E0 *
2333 000E0 * addeb = début de la zone à détruire (début de ldeb)
2334 000E0 * adfin = fin de la zone à détruire (fin de lfin)
2335 000E0 * findeb = adfin - addeb
2336 000E0 * nbllg = lfin - ldeb + 1
2337 000E0
2338 000E0 *
2339 000E0 * A partir de maintenant, deux solutions :
2340 000E0 * - destruction interne (ou destruction simple)
2341 000E0 * - destruction externe (move dans un fichier externe)
2342 000E0 1B0000 D0=(5) =EXTFIL destruction externe ?
2343 000E1 146 C=DAT0 A
2344 000E1 8AE ?C=0 A y=est-il un fichier ?
2345 000E1 83 GOYES D500 oui : destruction externe
2346 000E1C
2347 000E1C
2348 000E1C
2349 000E1C .....
2350 000E1C * Cas 1 = Destruction simple
2351 000E1C .....
2352 000E1C
2353 000E1C 8F0000 GOSBVL =OBCOLL met (OUTBS.,AVMEMS) à 0
2354 000E23
2355 000E23 1B0000 D0=(5) adfin
2356 000E2A 142 A=DAT0 A A(A) := ad. last nib of old line + 1
2357 000E2D 184 D0=D0- 5 D0=(5) addeb
2358 000E30 146 C=DAT0 A C(A) := début
2359 000E33 EE C=A-C A C(A) := longueur
2360 000E35 10B R3=C
2361 000E38
2362 000E38 D0=(5) =FILADR
2363 000E3F 146 C=DAT0 A C(A) := file header
2364 000E42
2365 000E42 *
2366 000E42 * A(A) = address of last nibble + 1 of old line
2367 000E42 * C(A) = address of file header
2368 000E42 * R3(A) = length of old line in nibs
2369 000E42
2370 000E42 *
2371 000E42 GOSBVL =MGOSUB
2372 000E42 (ON(5) =RPLLIN
2373 000E42
2374 000E42 *
2375 000E42 * Il ne peut pas y avoir d'erreur. L'accès au fichier a
2376 000E42 * été déjà testé, et on détruit, donc on tasse : il ne
2377 000E42 * peut pas y avoir "Insufficient Memory".

```

```

Page 053      Editeur, execution des commandes <xcmd.as>
AREUH ASS. V2.4  Commande Delete

2374 00E4E      * Il n'y a donc pas de "GOC erreur"
2375 00E4E
2376 00E4E 6880      GOTO D900
2377 00E52
2378 00E52
2379 00E52      * Cas II : Destruction externe
2380 00E52
2381 00E52
2382 00E52
2383 00E52
2384 00E52      * Chercher la fin du fichier externe
2385 00E52      (adresse de destination)
2386 00E52
2387 00E52 34FFFF      LC(5) -1 chercher la dernière ligne
2388 00E59 1B0000      D0=(5) =EXTFIL
2389 00E60 142      A=DAT0 A A(A) := " file header
2390 00E63 8E0000      GOSUBL =extzek
2391 00E69
2392 00E69      * D0 = " fin du fichier
2393 00E69      * A(A) = numéro de la dernière ligne du fichier + 1
2394 00E69
2395 00E69 136      CD0EX C(A) := " fin du fichier
2396 00E6C 108      R0=C R0(A) := dstadr
2397 00E6F
2398 00E6F      * Debut et fin de la partie à remplacer
2399 00E6F
2400 00E6F 1B0000      D0=(5) addeb
2401 00E76 146      C=DAT0 A
2402 00E79 109      R1=C R1(A) := orgdeb
2403 00E7C 164      D0=D0+ 5 D0=(5) adfin
2404 00E7F 146      C=DAT0 A
2405 00E82 10A      R2=C R2(A) := orgfin
2406 00E85
2407 00E85 1B0000      D0=(5) =FILADR
2408 00E8C 142      A=DAT0 A A(A) := header origine
2409 00E8F
2410 00E8F 1A0000      D0=(4) =EXTFIL
2411 00E95 146      C=DAT0 A C(A) := header destination
2412 00E98
2413 00E98 850      ST=1 =sMOVE
2414 00E9B
2415 00E9B      * A(A) = " header origine (notre fichier)
2416 00E9B      * R1(A) = " début partie à déplacer
2417 00E9B      * R2(A) = " fin partie à déplacer
2418 00E9B      * C(A) = " header destination (fichier externe)
2419 00E9B      * R0(A) = " emplacement d'insertion
2420 00E9B      * sMOVE=1 pour copie avec destruction
2421 00E9B
2422 00E9B 8E0000      GOSUBL =MVFILE
2423 00EA1 560      GONC D600

```

```

Page 055      Editeur, execution des commandes <xcmd.as>
AREUH ASS. V2.4  Commande Delete

2474 00EF5 1A0000      D0=(4) nblig
2475 00EFB 146      C=DAT0 A C(A) := nombre de lignes
2476 00EFE
2477 00EFE      *
2478 00EFE      * A(A) = longueur de ce qui a été enlevé
2479 00EFE      * B(A) = numéro de l'ancienne première ligne
2480 00EFE      * C(A) = nombre de lignes
2481 00EFE 8E0000      GOSUBL =dellin actualise DERN
2482 00F04
2483 00F04
2484 00F04      *
2485 00F04      * Actualise le numéro de la ligne courante.
2486 00F04 1B0000      D0=(5) ldeb
2487 00F0B 146      C=DAT0 A
2488 00F0E 8E0A1F      GOSUBL setcou ", " := ldeb
2489 00F14
2490 00F14 8C0000      GOLONG =BOUCLE
2491 00F1A

```

```

Page 054      Editeur, execution des commandes <xcmd.as>
AREUH ASS. V2.4  Commande Delete

2424 00EA4
2425 00EA4      * S'il y a eu erreur, aucun des deux fichiers n'a bougé
2426 00EA4
2427 00EA4 6A0F      GOTO Derr Purger le fichier externe si besoin
2428 00EA8
2429 00EA8      *
2430 00EA8      * A(A) = nouvelle valeur de FILADR
2431 00EA8      * Réactualiser FILADR et tester s'il faut modifier
2432 00EA8      * le buffer (en fait, simplement DERN est modifié)
2433 00EA8
2433 00EA8 1B0000      D500 D0=(5) =FILADR
2434 00EAF 146      C=DAT0 A C(A) := ancien FILADR
2435 00EB2 140      DAT0=A A FILADR := nouvelle valeur
2436 00EB5 1A0000      D0=(4) =EXTFIL
2437 00EB8 142      A=DAT0 A A(A) := ancien EXTFIL
2438 00EBE 2A6      ?AFC A fichiers différents ?
2439 00EC1 91      GUYES D900 oui : sortie
2440 00EC3
2441 00EC3      *
2442 00EC3      * Non : modifier DERN
2443 00EC3
2443 00EC3 1A0000      D0=(4) nblig
2444 00EC9 146      C=DAT0 A C(A) := lfin - ldeb + 1
2445 00ECC 1A0000      D0=(4) =DERN
2446 00ED2 142      A=DAT0 A A(A) := dernière
2447 00ED5 CA      A=A+C A
2448 00ED7 140      DAT0=A A DERN := lfin - ldeb + 1
2449 00EDA
2450 00EDA
2451 00EDA      *
2452 00EDA      * Mise à jour du buffer et Sortie
2453 00EDA
2454 00EDA
2455 00EDA
2456 00EDA
2457 00EDA
2458 00EDA      *
2459 00EDA 8E0000      GOSUBL =syncbf
2460 00EE0
2461 00EE0
2462 00EE0      *
2463 00EE0      * Paramètres à mettre en place pour "dellin" :
2464 00EE0      * - nombre de lignes (lfin - ldeb + 1)
2465 00EE0      * - longueur de ce qui a été enlevé (adfin - addeb)
2466 00EE0      * - numéro de l'ancienne première ligne (ldeb)
2467 00EE0
2468 00EE0 1B0000      D0=(5) findeb
2469 00EE7 142      A=DAT0 A A(A) := long. de ce qui a été enlevé
2470 00EEA
2471 00EEA 1A0000      D0=(4) ldeb
2472 00EF0 146      C=DAT0 A
2473 00EF3 D5      B=C A B(A) := ldeb
2474 00EF5

```

```

Page 056      Editeur, execution des commandes <xcmd.as>
AREUH ASS. V2.4  Commande Exchange File

2493 00F1A
2494 00F1A
2495 00F1A      *
2496 00F1A      * cmXCHG
2497 00F1A
2498 00F1A      * But: changer le fichier en cours d'édition
2499 00F1A      * Entrée:
2500 00F1A      * - D(A) = LEN(param)
2501 00F1A      * - D1 = " param
2502 00F1A      * - R0 = nb de lignes introduites
2503 00F1A      * R1 = ldebut
2504 00F1A      * R2 = lfin
2505 00F1A      * R3(S) = 1 si '?' avant la commande
2506 00F1A      * Sortie: par BOUCLE
2507 00F1A      * Abime: tout
2508 00F1A      * Appelle: noquer, GETFIL, valid?, PURGE, inibuf, setcou
2509 00F1A      * Niveaux: N/A
2510 00F1A      * Historiques:
2511 00F1A      * 88/11/11: PD/JT conception & codage
2512 00F1A      * 88/11/13: PD/JT utilisation de svRIR2
2513 00F1A      * 89/06/17: PD/JT mise à jour du flag "read-only"
2514 00F1A
2515 00F1A 8C1C2F      InvCmd GOLONG invcmd
2516 00F20 8C952F      InvPrm GOLONG invprm
2517 00F26
2518 00F26 8E9A2F      =cmXCHG GOSUBL noquer
2519 00F2C 118      C=R0 nb lignes
2520 00F2F 8AE      ?C#0 A
2521 00F32 8E      GUYES InvCmd "Invalid Cmd"
2522 00F34
2523 00F34
2524 00F34
2525 00F34
2526 00F34
2527 00F34 850      ST=1 =sDELETE cas D ou X (et pas C/M)
2528 00F37 850      ST=1 =sXCHG cas X (et pas D)
2529 00F3A 8E0000      GOSUBL =GETFIL analyse et cherche le fichier
2530 00F40 402      GOC ErReur ça s'est mal passé
2531 00F43
2532 00F43
2533 00F43
2534 00F43 1B0000      D0=(5) =EXTFIL
2535 00F4A 146      C=DAT0 A
2536 00F4D 2AA      ?C#0 A
2537 00F50 0D      GUYES InvPrm "Invalid Param"
2538 00F52
2539 00F52
2540 00F52
2541 00F52 8EF22F      GOSUBL valid?
2542 00F58 5E0      GONC X050 Pas d'erreur

```

```

2543 00F5B
2544 00F5B
2545 00F5B
2546 00F5B
2547 00F5B
2548 00F5B 8E0000 GOSUBL =PURGE n'abime pas C(A)
2549 00F61 8C0000 ErReuR GOLONG =xterr
2550 00F67
2551 00F67
2552 00F67
2553 00F67
2554 00F67
2555 00F67 1F0000 X050 D1=(5) =EXTFIL
2556 00F6E 147 C=DAT1 A
2557 00F71 1E0000 D1=(4) =FILADR
2558 00F77 145 DAT1=C A FILADR := EXTFIL
2559 00F7A
2560 00F7A
2561 00F7A
2562 00F7A 8E0000 GOSUBL =CHKWRT
2563 00E00 1F0000 D1=(5) =RONLY
2564 00F87 1554 DAT1=C S
2565 00F88
2566 00F88
2567 00F88
2568 00F88 8E0000 GOSUBL =inibuf
2569 00F91
2570 00F91 D2 C=0 A
2571 00F93 8E921F GOSUBL setcou courante := 1
2572 00F99
2573 00F99
2574 00F99
2575 00F99 8C0000 GOLONG =BOUCLE
2576 00F9F

```

- Commande invalide. Il faut revenir en arriere et détruire le fichier s'il vient d'être créé.
- Le fichier est la. Reste a mettre les bons pointeurs aux bons endroits.
- Regarder si on peut modifier le fichier extérieur.
- Initialiser le tableau de pointeurs
- Et retour a la boucle principale.

```

2620 00F9F
2621 00F9F
2622 00F9F D2
2623 00FA1 6600
2624 00FA5
2625 00FA5 301
2626 00FA8
2627 00FA8
2628 00FA8 1B00000
2629 00FAF 1540
2630 00FB3
2631 00FB3
2632 00FB3
2633 00FB3 8EC12F
2634 00FB9
2635 00FB9
2636 00FB9 160
2637 00FC 118
2638 00FC 1540
2639 00FC3 8E730F
2640 00FC9
2641 00FC9
2642 00FC9 340
2643 00FC 8E0000
2644 00FD2 407
2645 00FD5
2646 00FD5
2647 00FD5
2648 00FD5
2649 00FD5
2650 00FD5
2651 00FD5
2652 00FD5
2653 00FD5
2654 00FD5
2655 00FD5
2656 00FD5
2657 00FD5
2658 00FD5
2659 00FD5
2660 00FD5
2661 00FD5 8E3B1F
2662 00FDB
2663 00FDB 1B00000
2664 00FE2 146
2665 00FE5 8E0000
2666 00FE8 136
2667 00FEE 1B00000
2668 00FF5 144
2669 00FF8
2670 00FF8
2671 00FF8
2672 00FF8
2673 00FF8
2674 00FF8
2675 00FF8 1A0000
2676 00FE 1564
2677 01002 1A0000

```

- cmMOVE (n0 A C(0) := 0 (move) GOTO (000)
- cmCOPY LC(1) 1 C(0) := 1 (copie)
- C(0) = 1 (copie) ou 0 (move)
- (000 Dn(5) Csave DAT0=C P
- Interdire C/M avec '?'
- GOSUBL noquer
- Sauvegarder R0, R1 et R2 car GETFIL les abime.
- D0=D0+1 Csave+1 C=R0 C(A)=0..2 DAT0=C P Csave+1 := nb de lignes GOSUBL svP1R2
- Chercher si l'utilisateur a demande un C/M externe
- ST=0 +sDELETE cas 'C' ou 'M' GOSUBL =GETFIL Abime STMX et FUNCxv GOC Erreur
- Interdire tout ce qui peut suivre.
- GOSUBL valide
- D0=(5) =COUR C=DAT0 A GOSUBL =seek D0 := adresse de la ligne courante C=EX D0=(5) curadr DAT0=C A curadr := " ligne courante
- Tester les acces aux fichiers.
- On en profite pour restaurer "copie" qui etait sauvegarde en Csave.
- D0=(4) Csave C=DAT0 S D0=(4) copie

```

2578 00F9F
2579 00F9F
2580 00F9F
2581 00F9F
2582 00F9F
2583 00F9F
2584 00F9F
2585 00F9F
2586 00F9F
2587 00F9F
2588 00F9F
2589 00F9F
2590 00F9F
2591 00F9F
2592 00F9F
2593 00F9F
2594 00F9F
2595 00F9F
2596 00F9F
2597 00F9F
2598 00F9F
2599 00F9F
2600 00F9F
2601 00F9F
2602 00F9F
2603 00F9F
2604 00F9F
2605 00F9F
2606 00F9F
2607 00F9F
2608 00F9F
2609 00F9F
2610 00F9F
2611 00F9F
2612 00F9F
2613 00F9F
2614 00F9F
2615 00F9F
2616 00F9F
2617 00F9F
2618 00F9F
2619 00F9F
2620 00F9F
2621 00F9F
2622 00F9F
2623 00F9F
2624 00F9F
2625 00F9F
2626 00F9F
2627 00F9F

```

- cmMOVE, cmCOPY
- But: deplacer ou copier une partie d'un fichier avant la ligne courante.
- Entrée:
  - D(A) = LEN(param)
  - D1 = " param
  - R0 = nb de lignes introduites
  - R1 = ldebut
  - R2 = lfin
  - R3(S) = 1 si '?' avant la commande
- Sorties: par BOUCLE
- Abime: tout
- Appelle: noquer, GETFIL, valide, set.db, getdbf, svR1R2, CHKWRT, WRITE?
- Niveaux: N/A
- Algorithmes:
  - tester si l'utilisateur a demande un fichier externe
  - si interne
  - alors
  - valider les paramètres (., ldebut)
  - chercher les deux adresses dans le fichier courant
  - sinon
  - valider les paramètres (l, lfin)
  - chercher les deux adresses dans le fichier externe
  - fin si
  - deplacer le bloc
  - si cas "copie" ou (externe et fichiers differents)
  - alors
  - addln (ligne debut = courante, nblignes = nblig)
  - courante := nblig
  - sinon
  - si courante = ldeb
  - alors
  - addln (debut = ldeb, nblignes = nblig)
  - delin (debut = courante+nblig, nblignes = nblig)
  - courante := nblig
  - sinon si courante = lfin
  - addln (debut = courante, nblignes = nblig)
  - delin (debut = ldeb, nblignes = nblig)
  - sinon
  - courante := lfin + 1
  - fin si
  - fin si
- Historiques:
  - 88/11/12: FD/JI conception & codage
  - 89/06/17: PD/JI test du flag "read-only"

```

2678 01008 1544 DAT0=C S copie := valeur sauvegardée
2679 0100C 94E ?C=0 A
2680 0100F A3 GOC=0 C050 Copy
2681 01011
2682 01011
2683 01011
2684 01011
2685 01011 1F00000
2686 01018 1574
2687 0101C 8E0000
2688 01022 500
2689 01025
2690 01025 1F00000
2691 0102C 147
2692 0102F 8AA
2693 01032 B2
2694 01034 8E0000
2695 0103A 8E0000
2696 01040 4C1
2697 01043
2698 01043
2699 01043
2700 01043 8C0000
2701 01049
2702 01049
2703 01049
2704 01049
2705 01049
2706 01049 1F00000
2707 01050 1574
2708 01054 8E0000
2709 0105A 50E
2710 01050
2711 01050
2712 01050
2713 01050 1B00000
2714 01064 D2 C=0 A
2715 01066 1560 C=DAT0 P R0(A) := Csave+1
2716 0106A 108 R0=C
2717 01060
2718 01060 1A0000
2719 01073 146
2720 01076 109
2721 01079 160
2722 0107C 145
2723 0107F 10A
2724 01082
2725 01082
2726 01082
2727 01082 1B00000

```

- Move, Tester les deux acces.
- Changement du 17/06/89 : test du flag "read-only"
- D1=(5) =RONLY C=DAT1 S GOSUBL =WRITE? GOC Erreur C(A) = eFACCS, eFPROT ou eFOEN
- D1=(5) =EXTFIL C=DAT1 A ?C=0 A GOC=0 C070 Pas de fichier extérieur GOSUBL =CHKWRT GOSUBL =WRITE? GOC C070 Pas d'erreur
- Attention ! Le code continue !
- Erreur GOLONG =xterr
- C050
- Changement du 17/06/89 : test du flag "read-only"
- D1=(5) =RONLY C=DAT1 S GOSUBL =WRITE? GOC Erreur C(A) = eFACCS, eFPROT ou eFOEN
- Restaurer R0, R1 et R2. D0 n'a pas été modifié par WRITE?
- C070 D0=(5) (Csave)+1 C=0 A C=DAT0 P R0(A) := Csave+1 R0=C
- D0=(4) ldeb C=DAT0 A R1(A) := ldeb R1=C D0=D0+(lfin)-(ldeb) C=DAT0 A R2(A) := lfin R2=C
- Differentier C/M interne et externe
- D0=(5) =EXTFIL

```

2728 01089 146      C=DAT0 A      C(A) := adresse du fichier externe
2729 0105C 8AA      ?C=0 A
2730 0105F 60      GOYES C100 C/M interne
2731 01091 6910     GOTO C200 C/M externe
2732 01095
2733 01095
2734 01095
2735 01095
2736 01095
2737 01095
2738 01095
2739 01095
2740 01095
2741 01095 8ED40F  *
2742 01098         * C/M interne
2743 01098
2744 01098
2745 01098 8EF5FE  *
2746 010A1         * GOSUBL svR1R2
2747 010A1         *
2748 010A1         * Calcule addeb, adfin, findeb et nblig à partir de
2749 010A1         * ldeb et lfin.
2750 010A1         * ldeb = numéro de la première ligne à détruire
2751 010A1         * lfin = numéro de la dernière ligne à détruire
2752 010A1 8E17FE  *
2753 010A7 65D0    * GOSUBL getdbf
2754 010AB         * GOTO C300 Tout est en place pour MVFILE
2755 010AB
2756 010AB
2757 010AB
2758 010AB
2759 010AB
2760 010AB
2761 010AB
2762 010AB
2763 010AB
2764 010AB 118    *
2765 010AE CE      C=RR
2766 010B0 4A0    C=C-1 A
2767 010B3 CE      GOC C210 nb paramètres = 0
2768 010B5 561    C=C-1 A
2769 010B8 490    GONC C220 nb paramètres = 2
2770 010BB D2      * nb paramètres = 1
2771 010BE E6      GOC C215 nb paramètres = 1
2772 010BF 109    C=0 A
2773 010C2 34FFFFF C=C+1 A C(A) := 1
2774 010C3 10A    R1=C ldebut := 1
2775 010C3 10A    C215 LC(5) -1
2776 010C3 10A    R2=C lfin := Inf par défaut
2777 010CC

```

```

2828 01133
2829 01133 8E0000 *
2830 01139 1F00000 GOSUBL =extsk+ A(A) = numéro trouve, D0 = " ligne
2831 01140 136      D1=(5) adfin
2832 01143 145     CD0EX
2833 01146 1E0000  DAT1=C A      adfin := " ligne trouvée
2834 0114C 141     D1=(4) lfin
2835 0114F         DAT1=A A      lfin := numéro trouve
2836 0114F
2837 0114F
2838 0114F
2839 0114F
2840 0114F
2841 0114F
2842 0114F
2843 0114F
2844 0114F D5
2845 01151 1C0
2846 01154 147     C=DAT1 A      C(A) := ldeb
2847 01157 8AA     ?C=0 A
2848 0115A 88      GOYES c990 retour
2849 0115C EE      C=A-C A      C(A) := lfin - ldeb
2850 0115E 1E0000  D1=(4) nblig
2851 01164 145     DAT1=C A      nblig := lfin - ldeb
2852 01167
2853 01167 D4
2854 01169 1E0000  A=B A      A(A) := addeb
2855 0116F 147     D1=(4) addeb
2856 01172 EE      C=DAT1 A      C(A) := addeb
2857 01174 1E0000  C=A-C A      C(A) := adfin - addeb
2858 0117A 145     D1=(4) findeb
2859 0117D         DAT1=C A
2860 0117D
2861 0117D
2862 0117D
2863 0117D
2864 0117D
2865 0117D
2866 0117D
2867 0117D
2868 0117D
2869 0117D
2870 0117D
2871 0117D
2872 0117D
2873 0117D
2874 0117D
2875 0117D
2876 0117D
2877 0117D

```

- \* Ca y est !
- \* - ldeb = numéro de ligne de début (vérifié ds le fichier)
- \* - lfin = numéro de ligne de fin (vérifié dans le fichier)
- \* - addeb = " ligne de début dans le fichier externe
- \* - adfin = " ligne de fin dans le fichier externe
- \* - findeb = adfin - addeb
- \* - nblig = lfin - ldeb + 1
- \* Le code peut maintenant continuer dans le tronç commun.

```

.....
* C/M : Partie commune (déplacer le bloc)
.....
C300
*
* addeb = " début de la zone à détruire (début de ldeb)

```

```

2772 010CC
2773 010CC
2774 010CC
2775 010CC
2776 010CC
2777 010CC
2778 010CC
2779 010CC
2780 010CC
2781 010CC
2782 010CC
2783 010CC
2784 010CC
2785 010CC
2786 010CC
2787 010CC 111
2788 010CF 8A8    A=R1 A      A(A) := ldebut
2789 010D0 40    ?A=0 A
2790 010D4 11A    GOYES invPrm "Invalid Parm"
2791 010D7 88A    C=R2 A      C(A) := lfin
2792 010DA C0    ?A=C A
2793 010DC 8CD90F invPrm GOLONG invprM
2794 010E2
2795 010E2 6662  c990 GOTO C990
2796 010E6
2797 010E6
2798 010E6
2799 010E6
2800 010E6 8E41FE C250 GOSUBL svR1R2 sauver (R1, R2) en (ldeb, lfin)
2801 010EC
2802 010EC 119    C=R1 A      C(A) := ldeb
2803 010EF 1B00000 D0=(5) =EXTFIL
2804 010F6 142    A=DATE A      A(A) := " file header
2805 010F9 8E0000 GOSUBL =extsk+ A(A) = numéro trouve, D0 = " ligne
2806 010FF 1F00000 D1=(5) addeb
2807 01105 136    CD0EX
2808 01109 145    DAT1=C A      addeb := " ligne trouvée
2809 0110C 134    D0=C
2810 0110F 1E0000 D1=(4) ldeb
2811 01115 141    DAT1=A A      ldeb := numéro trouve
2812 01116
2813 01118 170    D1=D1+ (lfin)-(ldeb)
2814 01118 147    C=DAT1 A      C(A) := lfin
2815 0111E D5     B=C A      B(A) := lfin
2816 01120 E5     B=B+1 A
2817 01122 440    GOC C250 branch if lfin = #FFFFF
2818 01125 E6     C=C-1 A      C(A) := lfin + 1
2819 01127
2820 01127 D0
2821 01129 1F00000 B=A A      B(A) := ldeb
2822 01130 143    D1=(5) =EXTFIL
2823 01133         A=DAT1 A      A(A) := " file header
2824 01133
2825 01133
2826 01133
2827 01133

```

- \* A(A) = " file header
- \* B(A) = ldeb
- \* D0 = " ligne de début
- \* C(A) = lfin + 1 ou bien #FFFFF

```

2878 0117D
2879 0117D
2880 0117D
2881 0117D
2882 0117D
2883 0117D
2884 0117D
2885 0117D
2886 0117D 840
2887 01180 1B00000 ST=0 =sMOVE cas "copie" par défaut
2888 01187 1564    D0=(5) copie
2889 01188 94E    C=DAT0 S      C(S) := 1 si "copie", 0 si "move"
2890 0118E 50     ?C=0 S      cas "copie" ?
2891 01190 850    GOYES C310 out
2892 01193         ST=1 =sMOVE non : alors cas "move"
2893 01193 1A0000 C310 D0=(4) addeb
2894 01199 146    C=DAT0 A      C(A) := addeb
2895 0119C 109    R1=C R1 := " début partie à déplacer
2896 0119F 164    D0=D0+ 5      D0=(5) adfin
2897 011A2 146    C=DAT0 A      C(A) := adfin
2898 011A5 10A    R2=C R2 := " début partie à déplacer
2899 011A8 1A0000 D0=(4) curadr
2900 011AE 146    C=DAT0 A      C(A) := " ligne courante
2901 011B1 108    R0=C R0 := " destination address
2902 011B4 1A0000 D0=(4) =FILADR
2903 011B4 146    C=DAT0 A      A(A) := " header destination
2904 011B0 1A0000 D0=(4) =EXTFIL
2905 011C3 142    A=DAT0 A      C(A) := 0 ou " header origine
2906 011C5 8AC    ?A#0 A      C/M externe ?
2907 011C9 40     GOYES C320 non : A(A) est donc valide.
2908 011CB DA     A=C A      A(A) := " header origine
2909 011CD
2910 011CD
2911 011CD
2912 011CD
2913 011CD
2914 011CD
2915 011CD
2916 011CD
2917 011CD 8E0000 GOSUBL =MVFILE
2918 011D3 580    GONC C330 Pas d'erreur
2919 011D6 8C0000 GOLONG =xederr
2920 011DC
2921 011DC
2922 011DC
2923 011DC
2924 011DC
2925 011DC
2926 011DC
2927 011DC

```

- \* adfin = " fin de la zone à détruire (fin de lfin)
- \* findeb = adfin - addeb
- \* nblig = nombre de lignes traitées
- \* curadr = adresse de la ligne courante
- \* FILADR = adresse du fichier courant
- \* EXTFIL = 0 ou adresse du fichier externe



Page 065  
AREUH ASS. V2.4

Editeur, exécution des commandes 'xcmd.as'  
Commandes Copy / Move

```

2928 0110C 1B00000 D0=(5) =FILADR
2929 011E3 144 DAT0=C A FILADR réactualisé (destination)
2930 011E6 1A0000 D0=(4) =EXTFIL
2931 011EC 140 DAT0=A A EXTFIL réactualisé (origine)
2932 011EF
2933 011EF
2934 011EF
2935 011EF
2936 011EF 8E0000 GOSUBL =synchf
2937 011F5
2938 011F5
2939 011F5
2940 011F5
2941 011F5 1B00000 D0=(5) copie
2942 011FC 1564 C=DAT0 S
2943 01200 94E ?C#0 S
2944 01203 91 GOYES C400 "copie"
2945 01205
2946 01205 1A0000 D0=(4) =FILADR
2947 01208 146 C=DAT0 A C(A) := " fichier courant
2948 0120E 1A0000 D0=(4) =EXTFIL
2949 01214 142 A=DAT0 A A(A) := " fichier externe
2950 01217 8A2 ?A=C A
2951 0121A F3 GOYES C500 M interne
2952 0121C
2953 0121C
2954 0121C
2955 0121C
2956 0121C
2957 0121C
2958 0121C
2959 0121C
2960 0121C
2961 0121C
2962 0121C 1B00000 D0=(5) =COUR
2963 01223 146 C=DAT0 A
2964 01226 D5 B=C A B(A) := no nouvelle première ligne
2965 01228
2966 01228 1A0000 D0=(4) nbilig
2967 0122E 146 C=DAT0 A C(A) := lfin - ldeb + 1
2968 01231
2969 01231 1A0000 D0=(4) findeb
2970 01237 142 A=DAT0 A A(A) := lg de ce qui est rajouté
2971 0123A
2972 0123A 8E0000 GOSUBL =addlin
2973 01240
2974 01240
2975 01240
2976 01240 1B00000 D0=(5) nbilig
2977 01247 142 A=DAT0 A A(A) := nbilig

```

- Le buffer a très certainement été déplacé durant l'opération. Remettre son adresse à jour :
- GOSUBL =synchf
- Les déplacements sont faits. Il faut donc mettre à jour le buffer, et c'est très très compliqué !
- C interne/externe ou M externe
- C400
- Ajouter (lfin - ldeb + 1) lignes juste avant la ligne "courante".
- actualiser courante.

Page 067  
AREUH ASS. V2.4

Editeur, exécution des commandes 'xcmd.as'  
Commandes Copy / Move

```

3028 012CC 146 C=DAT0 A C(A) := courante
3029 012CF C2 C=C+A A C(A) := courante + nbilignes
3030 012D1 6170 GOTO C900
3031 012D5
3032 012D5 160 C500 D0=D0+ (lfin)-(ldeb)
3033 012D8 146 C=DAT0 A C(A) := lfin
3034 012DB 8BA ?A=C A courante += lfin
3035 012DE 95 GOYES C540
3036 012E0
3037 012E0
3038 012E0
3039 012E0 1B00000 D0=(5) =COUR
3040 012E7 146 C=DAT0 A
3041 012EA D5 B=C A B(A) := ligne début
3042 012EC 1A0000 D0=(4) findeb
3043 012F2 142 A=DAT0 A A(A) := long de ce qui a été rajouté
3044 012F5 1A0000 D0=(4) nbilig
3045 012FB 146 C=DAT0 A C(A) := nb lignes
3046 012FE
3047 012FE
3048 012FE
3049 012FE
3050 012FE
3051 012FE 8E0000 GOSUBL =addlin
3052 01304
3053 01304 1B00000 D0=(5) ldeb
3054 01308 146 C=DAT0 A
3055 0130E D5 B=C A B(A) := ldeb
3056 01310 1B00000 D0=(5) nbilig
3057 01317 146 C=DAT0 A C(A) := nbilignes
3058 0131A 1A0000 D0=(4) findeb
3059 01320 142 A=DAT0 A A(A) := long de ce qui a été enlevé
3060 01323
3061 01323
3062 01323
3063 01323
3064 01323
3065 01323 8E0000 GOSUBL =dellin
3066 01329
3067 01329 1B00000 D0=(5) =COUR
3068 01330 146 C=DAT0 A
3069 01333
3070 01333 6F00 GOTO C900
3071 01337
3072 01337
3073 01337
3074 01337
3075 01337
3076 01337 1B00000 D0=(5) lfin
3077 0133E 146 C=DAT0 A

```

- courante après le bloc (courante - lfin)
- D0=(5) =COUR
- B(A) := ligne début
- A(A) := long de ce qui a été rajouté
- C(A) := nb lignes
- A(A) = longueur de ce qui a été rajouté (adfin - addeb)
- B(A) = no de la nouvelle première ligne (courante)
- C(A) = nombre de lignes (nbilig)
- A(A) = longueur de ce qui a été enlevé (adfin - addeb)
- B(A) = no de l'ancienne première ligne (ldeb)
- C(A) = nombre de lignes (nbilig)
- GOSUBL =dellin
- Courante dans le bloc (ldeb = courante = lfin)

Page 066  
AREUH ASS. V2.4

Editeur, exécution des commandes 'xcmd.as'  
Commandes Copy / Move

```

2978 0124A 1A0000 D0=(4) =COUR
2979 01250 146 C=DAT0 A C(A) := courante
2980 01253 C2 C=C+A A C(A) := nouvelle valeur de courante
2981 01255 6DE0 GOTO C900
2982 01259
2983 01259
2984 01259
2985 01259
2986 01259
2987 01259 1B00000 C500 D0=(5) =COUR
2988 01260 142 A=DAT0 A A(A) := courante
2989 01263 1A0000 D0=(4) ldeb
2990 01269 146 C=DAT0 A C(A) := ldeb
2991 0126C 8BE ?A=C A courante += ldeb
2992 0126F 66 GOYES C520
2993 01271
2994 01271
2995 01271
2996 01271 1B00000 D0=(5) ldeb
2997 01273 146 C=DAT0 A
2998 01278 D5 B=C A B(A) := ligne début
2999 0127D 1A0000 D0=(4) findeb
3000 01283 142 A=DAT0 A A(A) := long de ce qui a été rajouté
3001 01286 1A0000 D0=(4) nbilig
3002 0129C 146 C=DAT0 A C(A) := nb lignes
3003 0129F
3004 0129F
3005 0129F
3006 0129F
3007 0129F
3008 0129F 8E0000 GOSUBL =addlin
3009 01299
3010 01295 1B00000 D0=(5) =COUR
3011 0129C 146 C=DAT0 A
3012 0129F D5 B=C A
3013 012A1 1B00000 D0=(5) nbilig
3014 012A3 146 C=DAT0 A C(A) := nbilignes
3015 012AB C1 B=B+C A B(A) := courante + nb lignes
3016 012AD 1A0000 D0=(4) findeb
3017 012B3 142 A=DAT0 A A(A) := long de ce qui a été enlevé
3018 012B6
3019 012B6
3020 012B6
3021 012B6
3022 012B6
3023 012B6 8E0000 GOSUBL =dellin
3024 012BC
3025 012BC 1B00000 D0=(5) nbilig
3026 012C3 142 A=DAT0 A A(A) := nbilignes
3027 012C6 1A0000 D0=(4) =COUR

```

- M interne
- Courante avant le bloc (courante - ldeb)
- A(A) = longueur de ce qui a été rajouté (adfin - addeb)
- B(A) = no de la nouvelle première ligne (ldeb)
- C(A) = nombre de lignes (nbilig)
- GOSUBL setcou pour vérification

Page 068  
AREUH ASS. V2.4

Editeur, exécution des commandes 'xcmd.as'  
Commandes Copy / Move

```

3078 01341 E6 C=C-1 A
3079 01343
3080 01343
3081 01343
3082 01343
3083 01343
3084 01343
3085 01343
3086 01343
3087 01343 8E97DE GOSUBL setcou pour vérification
3088 01349 8C0000 C900 GOLONG =BOUCLE
3089 0134F
3090 0134F END

```

- Attention ! Le code continue !
- C(A) = nouvelle valeur de "courante"