

```

0001 00000 TITLE Module principal main.as
0002 00000
0003 00000
0004 00000
0005 00000
0006 00000
0007 00000
0008 00000
0009 00000
0010 00000
0011 00000
0012 00000
0013 00000
0014 00000
0015 00000
0016 00000
0017 00000
0018 00000
0019 00000
0020 00000
0021 00000
0022 00000
0023 00000
0024 00000
0025 00000
0026 00000
0027 00000
0028 00000
0029 00000
0030 00000
0031 00000
0032 00000
0033 00000
0034 00000
0035 00000
0036 00000
0037 00000
0038 00000
0039 00000
0040 00000
0041 00000
0042 00000
0043 00000
0044 00000
0045 00000
0046 00000
0047 00000
0048 00000
0049 00000
0050 00000

```

- Allocation du bit restant dans le ram-réservee
- A mettre dans def.as lors de l'intégration dans JPC Rom

```

+RES:JPC EQU 214991
+INEDIT EQU 21000 bit de T/XEDIT
+ASPMASK EQU 20010

```

- Version A = 88/10/16
- distribution à Bultennure
- L. P. J. T. S. R marchent convenablement
- Le mode HF-0x n'a plus besoin de "L" pour délimiter les deux paramètres
- Version A1 = 88/10/22 -> 88/10/...
- distribution à Laurent Istria
- correction de l'accès au message "Y/N/O"
- le flag 1 signale quand on est en insertion
- dernier bit de la Reserved Ram utilisé par T/XEDIT
- Version A2 = 88/10/... -> 88/11/05
- distribution à la réunion PPC de Novembre 1988
- les commandes J et H ont été écrites
- les commandes sont accessibles par un message
- suppression du buffer en cas d'erreur "chaîne générique invalide"
- la commande "240" posait un problème (ne reconnaissait pas EOL)
- D fonctionne en interne (i.e. sans fichier externe)
- J et R n'acceptent plus les lignes à 64 K
- Version A3 = 88/11/05 - 88/11/13
- distribution à Jean-Jacques Dhénin
- D fonctionne en interne et externe
- correction des accès aux fichiers en Rom ou Secure
- amélioration des traitements d'erreur
- la commande X a été écrite
- les commandes C et M fonctionnent !
- utilisation de l'id #E1
- traduction des touches avec "KEYWT"
- introduction du message "Insufficient Memory"
- écriture du traducteur de messages
- Version A4 = 88/11/13 -> 88/11/20
- distribution à Laurent Istria, Jean-Jacques Dhénin, Eric Gengoux
- il n'y a plus de sortie brutale sur "eMEM"
- visualisation de la ligne courante avec [ENDLINE]
- Version A5 = 88/11/20 - 88/11/26
- distribution à Jean-Jacques Dhénin
- correction de ...C (début dest = début source)

```

0100 00062
0101 00062 240 CON(3) (TxEn05)-(TxTbSt)
0102 00065 00000 REL(5) =EDITe
0103 0006A D NIBHEX D
0104 0006B
0105 0006B F40 CON(3) (TxEn06)-(TxTbSt)
0106 0006E 00000 REL(5) =XEDITe
0107 00073 D NIBHEX D
0108 00074
0109 00074 TxTbSt
0110 00074 D TxEn01 CON(1) 13
0111 00075 6494C454 NIBASC 'FILEPOS'
0007D 05F435
0112 00083 07 CON(2) 0+t
0113 00085 B TxEn02 CON(1) 11
0114 00086 7454E4C4 NIBASC 'GENLEN'
0008E 54E4
0115 00092 17 CON(2) 1+t
0116 00094 B TxEn03 CON(1) 11
0117 00095 7454E405 NIBASC 'GENPOS'
0009D F435
0118 000A1 27 CON(2) 2+t
0119 000A3 F TxEn04 CON(1) 15
0120 000A4 7454E425 NIBASC 'GENRPLC4'
000AC 05C43442
0121 000A4 37 CON(2) 3+t
0122 000B6 9 TxEn05 CON(1) 9
0123 000B7 45544494 NIBASC 'EDIT'
000BF 45
0124 000C1 47 CON(2) 4+t
0125 000C3 9 TxEn06 CON(1) 9
0126 000C4 85544494 NIBASC 'XEDIT'
000CC 45
0127 000CE 57 CON(2) 5+t
0128 000D0 1FF NIBHEX 1FF
0129 000D3
0130 000D3 =EDITd
0131 000D3 8F00000 =XEDITd GOSBVL =FILDc+
0132 000DA 8D00000 GOVLNG =DROFDc
0133 000E1
0134 000E1 8F00000 numck GOSBVL =NUMCK
0135 000E8 8D00000 respnr GOVLNG =RESPTR
0136 000EF
0137 000EF =EDITp
0138 000EF =XEDITp
0139 000EF
0140 000EF
0141 000EF
0142 000EF
0143 000EF 7D10 GOSUB fichip

```

- Note : Lors de l'intégration dans JPC Rom, les parses de XEDIT et TEDIT sont les mêmes que celle de ADCREATE.

```

0051 00000
0052 00000
0053 00000
0054 00000
0055 00000
0056 00000
0057 00000
0058 00000
0059 00000
0060 00000
0061 00000
0062 00000
0063 00000
0064 00000
0065 00000
0066 00000 54443445 NIBASC 'EDIT'
0067 00010 202E NIBHEX 202E
0068 00014 00 CON(2) 0
0069 00016 0000 CON(4) 0
0070 0001A 000000 CON(6) 0
0071 00020 00000 REL(5) =FILEnd
0072 00025
0073 00025 xid EQU #E1
0074 00025 t EQU #E2
0075 00025
0076 00025 IE CON(2) xid
0077 00027 07 CON(2) t Lowest Token
0078 00029 57 CON(2) 5+t Highest Token
0079 0002B 37200 REL(5) find Next Lev
0080 00030 F NIBHEX F
0081 00031 4400 REL(4) 1+T/TbSt
0082 00035 0000 REL(4) =MSGTAB1 Offset to Message Table
0083 00039 70100 REL(5) POLHND Pol1
0084 0003E
0085 00040 000 CON(3) (TxEn01)-(TxTbSt)
0086 00041 00000 REL(5) =FILEPOS
0087 00045 F NIBHEX F
0088 00047
0089 00047 110 CON(3) (TxEn02)-(TxTbSt)
0090 0004A 00000 REL(5) =GENLEN
0091 0004F F NIBHEX F
0092 00050
0093 00050 020 CON(3) (TxEn03)-(TxTbSt)
0094 00053 00000 REL(5) =GENPOS
0095 00055 F NIBHEX F
0096 00055
0097 00059 F0x CON(3) (TxEn04)-(TxTbSt)
0098 0005C 00000 REL(5) =GENRPLC4
0099 00061 F NIBHEX F

```

```

0144 000F3
0145 000F3 8F00000 GOSBVL =EOLc
0146 000FA 4DE SOC respnr
0147 000FD 77EF GOSUB respnr
0148 00101 7C10 GOSUB virgup
0149 00105 8F00000 GOSBVL =STRGCK
0150 0010C 8BDF GOTO respnr
0151 00110
0152 00110 8F00000 fichip GOSBVL =FSPECp
0153 00117 500 RTNCK
0154 0011A 8D00000 GOVLNG =FSPECe
0155 00121
0156 00121 8F00000 virgup GOSBVL =NTOKEN
0157 00128 8F00000 GOSBVL =COMCK+
0158 0012F 400 RTNCK Cy = 1 ==> tCOMMA trouvé
0159 00132 8D00000 GOVLNG =HSPARe
0160 00139
0161 00139
0162 00139
0163 00139
0164 00139
0165 00139 8D00000 errp GOVLNG =SYNTXe
0166 00140
0167 00140
0168 00140
0169 00140
0170 00140 969 POLHND ?B=0 B
0171 00143 D0 GOYES POL010
0172 00145 3100 LC(2) =pkYDF
0173 00149 951 ?B=C B
0174 0014C 83 GOYES WKYDEF
0175 0014E 00 RTNSXh
0176 00150 110 POL010 C=R3
0177 00153 135 D1=C
0178 00156 112 A=R2
0179 00159 1C0 D1-D1- (VERSen)-(VERSt)-2
0180 0015C 137 CDIEX
0181 0015F 8B5 ?A=C A
0182 00162 C1 GOYES POL050
0183 00164 135 D1=C
0184 00167 10B R3=C
0185 0016A 3D7314A3 VERSt LCASc 'XED:A7'
00172 44548502
0186 0017A 1500 VERSen DAT1=C (VERSen)-(VERSt)-2
0187 0017E 00 POL050 RTNSXh
0188 00180
0189 00180 MNEMF EQU 8
0190 00180 MODF EQU 15
0191 00180 COMF EQU 24
0192 00180

```

- Fin de la partie extraite de ADCREATE

```

0193 00180 .....
0194 00180 * Auteurs: Stephane Barisien 03/85 - Interception
0195 00180 * de KEYDF et champs assembleur.
0196 00180 * 1ere modification: M.MARTINET et P.DAVID 10/85 -
0197 00180 * transformation en FIELD ON | OFF.
0198 00180 * 2eme modification: M.MARTINET 12/85 - mise en
0199 00180 * place de la bascule touche CALC et suppres-
0200 00180 * sion token FIELD ON | OFF.
0201 00180 * 3eme modification: J.Taillandier et P.David
0202 00180 * 03/86, reunion de JPC-LEX
0203 00180 .....
0204 00180
0205 00180 6A10 HASF1 GOTO HASF
0206 00184
0207 00184 110 hKYDEF A=R0 (A) = code logique
0208 00187 3102 LC(2) 02
0209 00188 962 7A=C B
0210 0018E 2F GOYES HASF1
0211 00190 * LC(2) =kcVIEW
0212 00190 * 7A=C B
0213 00190 * GOYES hCURS_1
0214 00190 * LC(2) =kcLFT
0215 00190 * 7A=C B
0216 00190 * GOYES hCALC_1
0217 00190 3100 LC(2) =kcCALC
0218 00194 962 7A=C B
0219 00197 9E GOYES HASF1
0220 00199 00 RTNSM
0221 0019B
0222 0019B hASF
0223 0019B *
0224 0019B * Est-on sous editeur ? Test modifie le 03/10/23 par PD/JT
0225 0019B *
0226 0019B 1B199F2 D0=(5) =RESJPC
0227 001A2 1520 A=DAT0 P
0228 001A6 308 LC(1) =INEDIT
0229 001A9 0E06 A=ASC P
0230 001AD 908 7A=0 P
0231 001B0 24 GOYES rtn
0232 001B2 *
0233 001B2 * R0(A) = touche pressee
0234 001B2 *
0235 001B2 110 A=R0
0236 001B5 3100 LC(2) =kcCALC Est-ce
0237 001B9 966 7A=C B la touche [F][CALC] ?
0238 001BC 83 GOYES FX0 Non: execution suite prgm.
0239 001BE * Touche [F][CALC] : inversion de l'etat du bit ASF
0240 001BE 1A199F D0=(4) =RESJPC
0241 001C4 1520 A=DAT0 P
0242 001C8 302 LC(1) =ASFMSK

```

```

0293 00263 E6 C=C+1 A de la touche
0294 00265 1A0000 D0=(4) =DEFADR espace
0295 00268 14C DAT0=C B avec
0296 0026E 161 D0=D0+ 2 le
0297 00271 304 LCHEX. 4 bon
0298 00274 15C0 DAT0=C 1 nombre
0299 00278 7210 G0SUB SKPTBL de
0300 0027C 02020202 NIBASC ' CARCTERES BLANCS
0301 00280 02 NIBASC '
0302 0028E 07 SKPTBL C=RSTK Sortie de table
0303 00290 160 D0=D0+ 1
0304 00293 144 DAT0=C A
0305 00296 050 ST=1 0
0306 00299 021 rtnxm0 XM=0 ST[0]=1 => definit. touche
0307 0029C 03 RTNCC Puis interception du Poll.
0308 0029E
0309 0029E Find
0310 0029E
0311 0029E tFIND EQU 75
0312 0029E
0313 0029E 1E CON(2) =id
0314 002A0 E4 CON(2) tFIND
0315 002A2 E4 CON(2) tFIND
0316 002A4 00000 CON(5) 0 Next Lex
0317 002A9
0318 002A9 F NIBHEX F Pas de speed table
0319 002AA 7100 REL(4) 1+FindTxTbSt
0320 002AE 0000 REL(4) =MSGTBL2 Offset to Message Table
0321 002B2 00000 CON(5) 0 Poll handler
0322 002B7
0323 002B7 *
0324 002B7 * Main Table
0325 002B7 *
0326 002B7 000 CON(3) (FindTxEn0)-(FindTxTbSt)
0327 002BA 00000 REL(5) =FIND
0328 002BF 1 CON(1) #1 Stat. Non programmable
0329 002C0 *
0330 002C0 *
0331 002C0 * Text Table
0332 002C0 *
0333 002C0 FindTxTbSt
0334 002C0 FindTxEn0
0335 002C0 7 CON(1) 7
0336 002C1 6494E444 NIBASC 'FIND'
0337 002C9 B4 CON(2) tFIND
0338 002CB *
0339 002CB 1FF NIBHEX 1FF
0340 002CE

```

```

0243 001CB 08 B=A A
0244 001CD 0EF2 C=CAA A C(0) := ASHMSK ou 0
0245 001D1 9AA 7C=0 P
0246 001D4 C0 GOYES asf10 Il faut le mettre à 1
0247 001D6 * Il faut mettre le bit à 0
0248 001D6 300 A=ASC P A(0) := bit ASF à 0
0249 001D9 0E06 G0NC asf20 B.E.T.
0250 001DD 590 * Il faut mettre le bit à 1
0251 001E0 asf10 LC(1) =ASFMSK
0252 001E0 302 A=ASC P A(0) := bit ASF à 1
0253 001E3 0EAF tF2M DATA=A F
0254 001E7 1500 7A=0 0 Touche inhibee, pas de definition
0255 001EB 240 GOTO tFnm0 et fin d'execution
0256 001EE 6A00
0257 001F2
0258 001F2 00 rtn RTNSM
0259 001F4 1A199F FX0 D0=(4) =RESJPC Execution du programme
0260 001F4 15A0 A=DAT0 1 Est-on en mode field on ?
0261 001FE 302 LC(1) =ASFMSK
0262 00201 0E06 A=ASC F
0263 00205 908 7A=0 P
0264 00208 AE GOYES rtn Non: fin de programme
0265 0020A 1A0000 D0=(4) =DSPBFS Oui: continu
0266 00210 32000 LC(3) (=DSPBFE)-(=DSPBFS)-1
0267 00215 AB5 B=C X
0268 00218 14A SP05 A=DAT0 B
0269 0021B 968 7A=0 B
0270 0021E F1 GOYES SP08
0271 00220 3102 LCASC ' '
0272 00224 966 7A=C B
0273 00227 D0 GOYES SP06
0274 00229 161 D0=D0+ 2
0275 0022C A30 B=B-1 X
0276 0022F 50E G0NC SP05
0277 00232 00 rtnl RTNSM
0278 00234 31A2 SP06 LCASC ' ' Y a-t-il des remarques ?
0279 00238 962 7A=C B
0280 00238 7E GOYES rtn Oui: fin de programme
0281 0023D 1A0000 SP08 D0=(4) =CURSOR Verification
0282 00243 14A A=DAT0 B position du curseur
0283 00246 3170 LC(2) (MNEHF)-1 en fonction
0284 0024A 9EA 7A=C B des
0285 0024D 41 GOYES SP10 differents
0286 0024F 31E0 LC(2) (MODF)-1 champs:
0287 00253 9EA 7A=C B MNEHF: memo-field
0288 00256 BA GOYES SP10 MODF: modification-field
0289 00258 3171 LC(2) (COMF)-1 COMF: comment-field
0290 0025C 9E6 7A=C B
0291 0025F 3D GOYES rtnl
0292 00261 E2 SP10 C=C-A A Redefinition

```

```

0342 002CE .....
0343 002CE * FIND0
0344 002CE *
0345 002CE *
0346 002CE * But: analyser et reconnaître la syntaxe de FIND
0347 002CE * Syntaxe: FIND (chaine) [,<debut> [,<fin>]]
0348 002CE * Entrées:
0349 002CE * - D1 = ' chaîne Ascii
0350 002CE * - D0 = ' tokens
0351 002CE * - D(A) = AVHEME
0352 002CE * Sorties:
0353 002CE * - D1 = 0
0354 002CE * - Ables: A=C, B(15-5), R0-R1, R3, 10-2, 17, 111,
0355 002CE * - FINE00, D0, D1
0356 002CE * - Niveaux: 5 (STRGCK)
0357 002CE * - Appelles: STRGCK, COMCK+, ck211#, RESPTR
0358 002CE * - Historique:
0359 002CE * - 09/06/09: P.D., reconception & recodage
0360 002CE .....
0361 002CE respTr
0362 002CE GOVLNG =RESPTR
0363 002CE SD00000
0364 002D5 =FIND0
0365 002D5 G0SBVL =STRGCK
0366 002D5 2F00000
0367 002DC *
0368 002DC * A = token de ce qui suit l'expression
0369 002DC * D1 = " derrière le texte correspondant à A
0370 002DC *
0371 002DC 3F00000 G0SBVL =COMCK+
0372 002E3 5AE G0NC respTr Pas de ' ' : c'est fini !
0373 002E6 181 D0=D0- 2 ck211# sortira une virgule
0374 002E9 *
0375 002E9 * Une virgule a été reconnue, donc il faut un ou
0376 002E9 * deux numeros de ligne.
0377 002E9 * C'est le rôle de ck211#, empruntée à JPC Rom et
0378 002E9 * plus précisément à D/PBLIST/RENUMREM
0379 002E9 *
0380 002E9 *
0381 002E9 *
0382 002E9 * ATTENTION ! LE CODE CONTINUE !
0383 002E9 *
0384 002E9 *
0385 002E9 .....
0386 002E9 *
0387 002E9 * ck211#
0388 002E9 *
0389 002E9 * But: parse " (line#1) [ , (line#2) ] "
0390 002E9 * Entrée:
0391 002E9 * - D1 = ' blancs optionnels avant
0392 002E9 * Sortie:

```

```

*0000 00000
*0001 00000
*0002 00000
*0003 00000
*0004 00000
*0005 00000
*0006 00000
*0007 00000
*0008 00000
*0009 00000
*0010 00000
*0011 00000
*0012 00000
*0013 00000
*0014 00000
*0015 00000
*0016 00000
*0017 00000
*0018 00000
*0019 00000
*0020 00000
*0021 00000
*0022 00000
*0023 00000
*0024 00000
*0025 00000
*0026 00000
*0027 00000
*0028 00000
*0029 00000
*0030 00000
*0031 00000
*0032 00000
*0033 00000
*0034 00000
*0035 00000
*0036 00000
*0037 00000
*0038 00000
*0039 00000
*0040 00000
*0041 00000
*0042 00000
*0043 00000
*0044 00000
*0045 00000
*0046 00000
*0047 00000
*0048 00000
*0049 00000
*0050 00000
*0051 00000
*0052 00000
*0053 00000
*0054 00000
*0055 00000
*0056 00000
*0057 00000
*0058 00000
*0059 00000
*0060 00000
*0061 00000
*0062 00000
*0063 00000
*0064 00000
*0065 00000
*0066 00000
*0067 00000
*0068 00000
*0069 00000
*0070 00000
*0071 00000
*0072 00000
*0073 00000
*0074 00000
*0075 00000
*0076 00000
*0077 00000
*0078 00000
*0079 00000
*0080 00000
*0081 00000
*0082 00000
*0083 00000
*0084 00000
*0085 00000
*0086 00000
*0087 00000
*0088 00000
*0089 00000
*0090 00000
*0091 00000
*0092 00000
*0093 00000
*0094 00000
*0095 00000
*0096 00000
*0097 00000
*0098 00000
*0099 00000
*0100 00000

```

```

+TEDItp 000EF Rel 0137 -
T-En01 00074 Rel 0110 - 0085
T-En02 00085 Rel 0113 - 0089
T-En03 00094 Rel 0116 - 0093
T-En04 000A2 Rel 0119 - 0097
T-En05 000B6 Rel 0122 - 0101
T-En06 000C3 Rel 0125 - 0105
T-TbSt 00074 Rel 0109 - 0081 0085 0089 0093 0097 0101 0105
VERLen 0017A Rel 0186 - 0179 0186
VFRst 0016A Rel 0185 - 0179 0186
=XEDItd 000D3 Rel 0131 -
=XEDIte Extrn Ukn - 0106
=XEDItp 000EF Rel 0138 -
asf10 001E0 Rel 0252 - 0246
asf20 001E7 Rel 0254 - 0250
ck11i# 002FD Rel 0407 - 0402
ck21i# 002E9 Rel 0401 -
errp 00139 Rel 0165 -
fichip 00110 Rel 0152 - 0143
find 0029E Rel 0309 - 0079
hasf 0019B Rel 0222 - 0205
hasf1 00180 Rel 0205 - 0210 0219
hYDEF 00184 Rel 0207 - 0174
+id 000E1 Abs 0073 - 0076 0313
+kcCALC Extrn Ukn - 0217 0236
numcf 000E1 Rel 0134 -
+pkYDF Extrn Ukn - 0172
respTr 002CE Rel 0362 - 0372 0404
respTr 000E8 Rel 0135 - 0146 0147 0150
rtn 001F2 Rel 0258 - 0231 0264 0280
rtn1 00232 Rel 0277 - 0291
rtn=0 00299 Rel 0306 - 0256
+COMMA Extrn Ukn - 0411
+LINE# Extrn Ukn - 0408
t 00070 Abs 0074 - 0077 0078 0112 0115 0118 0121 0124 0127
tFIND 0004B Abs 0311 - 0314 0315 0337
virgup 00121 Rel 0156 - 0148

```

Source : main.as
Object : obj/main.ao
Listing : list/main.al
Date : Sat Aug 12 17:40:48 1989

```

+ASFMSX 00002 Abs 0009 - 0242 0248 0252 0261
+COMCF Extrn Ukn - 0403
+COMCF+ Extrn Ukn - 0157 0371
+CURSGR Extrn Ukn - 0281
+COMF 00019 Abs 0191 - 0289
+DEFADR Extrn Ukn - 0294
+DROPOC Extrn Ukn - 0132
+DSPBFE Extrn Ukn - 0266
+DSPBFS Extrn Ukn - 0255 0266
+EOLCX Extrn Ukn - 0145
+FILEDC Extrn Ukn - 0101
+FILEPOS+ Extrn Ukn - 0096
+FINDE Extrn Ukn - 0327
+FINDE+ 0000E Rel 0355 -
+FSPECE Extrn Ukn - 0154
+FSPECE+ Extrn Ukn - 0150
+FileNd Extrn Ukn - 0071
+FXD 001F4 Rel 0255 - 0210
+FileNd 0002C Rel 0417 -
+FindT-En01 002C0 Rel 0134 - 0104
+FindT-TbSt 002C0 Rel 0330 - 0319 0324
+GENLEH+ Extrn Ukn - 0098
+GENPOS+ Extrn Ukn - 0094
+GENSFCE+ Extrn Ukn - 0090
+INEDIT 00000 Abs 0000 - 0200
+IVPARE Extrn Ukn - 0410
+LSTFE 00318 Rel 0415 - 0410
+MSGTBL1 Extrn Ukn - 0082
+MSGTBL2 Extrn Ukn - 0320
+MSPARE Extrn Ukn - 0159
+MNFEM 00008 Abs 0189 - 0203
+MOF 0000F Abs 0190 - 0206
+NTOKEN Extrn Ukn - 0156
+NTOKNL Extrn Ukn - 0407
+NUMCK Extrn Ukn - 0134
+OUT3TK Extrn Ukn - 0410
+POLR10 00150 Rel 0176 - 0171
+POLR50 0017E Rel 0107 - 0182
+POLHND 00140 Rel 0170 - 0083
+RESJPC 2F991 Abs 0007 - 0226 0240 0259
+RESPTR Extrn Ukn - 0135 0363
+STRGCK Extrn Ukn - 0149 0366
+SYNTX+ Extrn Ukn - 0165
+SKPTBL 0020E Rel 0302 - 0299
+SP05 00218 Rel 0265 - 0276
+SP06 00234 Rel 0278 - 0273
+SP08 00230 Rel 0201 - 0270
+SP10 00261 Rel 0292 - 0285 0288
+TEDItd 000D3 Rel 0130 -
+TEDIte Extrn Ukn - 0102

```

Errors : 000

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

```

0001 00000      *
0002 00000      * Fichier g n r  automatiquement par amg
0003 00000      * et modifi  automatiquement par chnsg
0004 00000      *
MBASE EQU 16
NULL EQU (MBASE)*0 EDIT
+XMEM EQU (MBASE)*1 Insufficient Memory
+ICMD EQU (MBASE)*2 Invalid Cmd
+IPAT EQU (MBASE)*3 Invalid Pattern
+ICMP EQU (MBASE)*4 No Room for Pattern
+PLIN EQU (MBASE)*5 Line . Cmd:
+PEOF EQU (MBASE)*6 Eof. Cmd:
+EOF EQU (MBASE)*7 [Eof]
+DELE EQU (MBASE)*8 Ok to Delete? Y/N:
+CONF EQU (MBASE)*9 Yes/No/Quit ?
+KEYS EQU (MBASE)*10 YNO
+VCMO EQU (MBASE)*11 CODEHJLMPORSTX
+HL00 EQU (MBASE)*12 Copy: [b[e]] C [<file>]
+HL01 EQU (MBASE)*13 Delete: [b[e]] D [<file>[*]]
+HL02 EQU (MBASE)*14 Exit: E
+HL03 EQU (MBASE)*15 Help: H [<cmd>]
+HL04 EQU (MBASE)*16 Insert: [I] I
+HL05 EQU (MBASE)*17 Join: [b[e]] [?] J [n]
+HL06 EQU (MBASE)*18 List: [b[e]] L [n]IN]
+HL07 EQU (MBASE)*19 Move: [b[e]] M [<file>]
+HL08 EQU (MBASE)*20 Print: [b[e]] P [n]IN]
+HL09 EQU (MBASE)*21 Quit: Q
+HL10 EQU (MBASE)*22 Replace: [b[e]] [?] R/str1/str2[
+HL11 EQU (MBASE)*23 Search: [b[e]] [?] S/str1/[
+HL12 EQU (MBASE)*24 Text: [I] T
+HL13 EQU (MBASE)*25 Exchange File: X [<file>]

BB42 EQU 42 EDIT
BB43 EQU 43 Insufficient Memory
BB44 EQU 44 Invalid Cmd
BB45 EQU 45 Invalid Pattern

+MSGTBL1
CON(2) (MBASE)*1 Lowest message #
CON(2) (MBASE)*25 Highest message #

* EDIT
CON(2) 16
CON(2) NULL Message # 16
CON(1) 4
NIBASC 'EDIT '
00011 02
CON(1) 12
0047 00013 C
0048 00014
0049 00014 * Insufficient Memory

```

```

0096 00090 C2023406 NIBASC '. Cmd: '
0098 00098 46A3
0097 0009C C
CON(1) 12
0098 0009D
0099 0009D * Eof. Cmd:
CON(2) 24
0100 0009D 21 CON(2) +tePEOF Message # 22
0101 0009F 61 CON(1) 2
0102 000A1 2 CON(1) 2
0103 000A2 54F666C2 NIBASC 'Eof. Cmd'
000AA 02340646
0104 000B2 A3 NIBASC '.'
0105 000B4 C CON(1) 12
0106 000B5
0107 000B5 * [Eof]
0108 000B5 01 CON(2) 16
0109 000B7 71 CON(2) +teEOF Message # 23
0110 000B9 4 CON(1) 4
0111 000BA B554F666 NIBASC '[Eof]'
000C2 D5
0112 000C4 C CON(1) 12
0113 000C5
0114 000C5 * Ok to Delete? Y/N:
CON(2) 44
0115 000C5 C2 CON(2) 44
0116 000C7 81 CON(2) +teDELE Message # 24
0117 000C9 8 CON(1) 11
0118 000CA F CON(1) 15
0119 000CB F4850247 NIBASC 'Ok to De'
000D3 F6024456
0120 000D8 C6564756 NIBASC 'lete? Y?'
000E3 F90295F2
0121 000E8 1 CON(1) 1
0122 000EC E4A3 NIBASC 'N:'
0123 000F0 C CON(1) 12
0124 000F1
0125 000F1 * Yes/No/Quit ?
CON(2) 33
0126 000F1 12 CON(2) 33
0127 000F3 91 CON(2) +teCONF Message # 25
0128 000F5 B CON(1) 11
0129 000F6 C CON(1) 12
0130 000F7 955637F2 NIBASC 'Yes/No/Q'
000FF E4F6F215
0131 00107 57964702 NIBASC 'Quit ?'
0010F F3
0132 00111 C CON(1) 12
0133 00112
0134 00112 * YNO
CON(2) 12
0135 00112 C0 CON(2) 12
0136 00114 A1 CON(2) +teEYS Message # 26
0137 00116 C CON(1) 2
0138 00117 95E415 NIBASC 'YNO'

```

```

0050 00014 E2 CON(2) 46
0051 00016 11 CON(2) +XMEM Message # 17
0052 00018 B CON(1) 11
0053 00019 F CON(1) 15
0054 0001A 94E63757 NIBASC 'Insuffic'
00022 66669636
0055 0002A 9656E647 NIBASC 'ient Mem'
00032 020456D6
0056 0003A 2 CON(1) 2
0057 0003B F62797 NIBASC 'orv'
0058 00041 C CON(1) 12
0059 00042
0060 00042 * Invalid Cmd
CON(2) 15
0061 00042 F0 CON(2) 15
0062 00044 21 CON(2) +teICMD Message # 18
0063 00046 E CON(1) 14
0064 00047 CE CON(2) 236
0065 00049 2 CON(1) 2
0066 0004A 34D646 NIBASC 'Cmd'
CON(1) 12
0067 00050 C
0068 00051
0069 00051 * Invalid Pattern
CON(2) 11
0070 00051 B0 CON(2) 11
0071 00053 31 CON(2) +teIPAT Message # 19
0072 00055 E CON(1) 14
0073 00056 CE CON(2) 236
0074 00058 D CON(1) 13
0075 00059 A2 CON(2) BB42
0076 0005B C CON(1) 12
0077 0005C
0078 0005C * No Room for Pattern
CON(2) 34
0079 0005C 22 CON(2) 34
0080 0005E 41 CON(2) +teICMP Message # 20
0081 00060 B CON(1) 11
0082 00061 B CON(1) 11
0083 00062 E4F60225 NIBASC 'No Room '
0006A F6F6D602
0084 00072 66F62702 NIBASC 'for '
CON(1) 13
0085 0007A D CON(2) BB42
0086 0007B A2 CON(1) 12
0087 0007D C
0088 0007E
0089 0007E * Line . Cmd:
CON(2) 31
0090 0007E F1 CON(2) 31
0091 00080 51 CON(2) +tePLIN Message # 21
0092 00082 4 CON(1) 4
0093 00083 C496E656 NIBASC 'Line '
0008B 02
0094 0008D F2 NIBHEX F2
0095 0008F 5 CON(1) 5

```

```

0139 0011D C CON(1) 12
0140 0011E
0141 0011E * CODEHJLMPORSTX
CON(2) 35
0142 0011E 32 CON(2) 35
0143 00120 B1 CON(2) +teVCMO Message # 27
0144 00122 B CON(1) 11
0145 00123 D CON(1) 13
0146 00124 34445484 NIBASC 'CODEHJLM'
0012C 34A4C4D4
0147 00134 05152535 NIBASC 'PORSTX'
0013C 4585
0148 00140 C CON(1) 12
0149 00141
0150 00141 * Copy: [b[e]] C [<file>]
CON(2) 34
0151 00141 22 CON(2) 34
0152 00143 C1 CON(2) +teHL00 Message # 28
0153 00145 5 CON(1) 5
0154 00146 34F60797 NIBASC 'Copy: '
0014E A302
0155 00152 D CON(1) 13
0156 00153 B2 CON(2) BB43
0157 00155 2 CON(1) 2
0158 00156 3402B5 NIBASC 'C ['
CON(1) 13
0159 0015C D CON(1) 13
0160 0015D C2 CON(2) BB44
0161 0015F 0 CON(1) 0
0162 00160 D5 NIBASC ']'
0163 00162 C CON(1) 12
0164 00163
0165 00163 * Delete: [b[e]] D [<file>[*]]
CON(2) 44
0166 00163 C2 CON(2) 44
0167 00165 D1 CON(2) +teHL01 Message # 29
0168 00167 7 CON(1) 7
0169 00168 4456C656 NIBASC 'Delete: '
00170 4756A302
0170 00178 D CON(1) 13
0171 00179 B2 CON(2) BB43
0172 0017B 2 CON(1) 2
0173 0017C 4402B5 NIBASC 'D ['
CON(1) 13
0174 00182 D CON(1) 13
0175 00183 C2 CON(2) BB44
0176 00185 3 CON(1) 3
0177 00186 B5B205D5 NIBASC '[*]]'
CON(1) 12
0178 0018E C
0179 0018F
0180 0018F * Exit: E
CON(2) 20
0181 0018F 41 CON(2) 20
0182 00191 E1 CON(2) +teHL02 Message # 30
0183 00193 6 CON(1) 6
0184 00194 54879647 NIBASC 'Exit: E'

```

```

0185 001A2 C          CON(1) 12
0186 001A3
0187 001A3          * Help: H [ <cmd> ]
0188 001A3 52        CON(2) 37
0189 001A5 F1        CON(2) +teHL03 Message # 31
0190 001A7 B          CON(1) 11
0191 001A8 E          CON(1) 14
0192 001A9 8456C607  NIBASC 'Help: H '
0193 001B9 85C336D6  NIBASC '[ <cmd> ]'
0194 001C7 C          CON(1) 12
0195 001C8
0196 001C8          * Insert: [ ] I
0197 001C8 12        CON(2) 33
0198 001CA 02        CON(2) +teHL04 Message # 32
0199 001CC B          CON(1) 11
0200 001CD C          CON(1) 12
0201 001CE 94E63756  NIBASC 'Insert: '
0202 001DE 2747A302  NIBASC '[ ] I '
0203 001E5 84
0204 001E8 C          CON(1) 12
0205 001E9
0206 001E9 82        * Join: [b[e]] [?] J [n]
0207 001E8 12        CON(2) 40
0208 001ED 5          CON(2) +teHL05 Message # 33
0209 001EE A4F696E6  NIBASC 'Join: '
0210 001FA D          CON(1) 13
0211 001FB B2        CON(2) BB43
0212 001FD 8          CON(1) 8
0213 001FE 85F3D502  NIBASC '[?] J [n]'
0214 00206 A40285E6  NIBASC ']'
0215 00210 C          CON(1) 12
0216 00211
0217 00211          * List: [b[e]] L [n][N]
0218 00211 B1        CON(2) 27
0219 00213 22        CON(2) +teHL06 Message # 34
0220 00215 5          CON(1) 5
0221 00216 C4963747  NIBASC 'List: '
0222 00222 D          CON(1) 13
0223 00223 B2        CON(2) BB43
0224 00225 0          CON(1) 0
0225 00226 F4        NIBASC 'L'
0226 00228 D          CON(1) 13

```

```

0273 00299 B          CON(1) 11
0274 0029A F          CON(1) 15
0275 0029B 85F3D502  NIBASC '[?] R/st'
0276 002A3 25F23747  002A3 25F23747
0276 002AB 2713F237  NIBASC 'r1/str2['
0277 002B3 47272385
0277 002B8 1          CON(1) 1
0278 002BC F2D5      NIBASC ']/]'
0279 002C0 C          CON(1) 12
0280 002C1
0281 002C1          * Search: [b[e]] [?] S/str[/]
0282 002C1 33        CON(2) 51
0283 002C3 72        CON(2) +teHL11 Message # 39
0284 002C5 7          CON(1) 7
0285 002CE 35561627  NIBASC 'Search: '
0286 002D6 D          CON(1) 13
0287 002D7 B2        CON(2) BB43
0288 002D9 B          CON(1) 11
0289 002DA B          CON(1) 11
0290 002DB 85F3D502  NIBASC '[?] S/st'
0291 002E8 2785F205  002E3 35F23747
0292 002F3 C          NIBASC 'r[/]'
0293 002F4           CON(1) 12
0294 002F4
0295 002F4 C1        * Text: [ ] T
0296 002F6 B2        CON(2) 28
0297 002F8 A          CON(2) +teHL12 Message # 40
0298 002F9 45568747  CON(1) 10
0299 00309 050245    NIBASC 'Text: [ ]'
0300 0030F C          NIBASC ']'
0301 00310           CON(1) 12
0302 00310
0303 00310 D2        * Exchange File: X <file>
0304 00312 92        CON(2) 45
0305 00314 B          CON(2) +teHL13 Message # 41
0306 00315 F          CON(1) 11
0307 00316 54873686  CON(1) 15
0308 00325 026496C6  NIBASC 'Exchange'
0309 00336 0          NIBASC 'File: X'
0310 00337 02        CON(1) 0
0311 00339 D          NIBASC ' '
0312 0033A C2        CON(1) 13
0313 0033C C          CON(2) BB44
0314 0033D           CON(1) 12
0315 0033D          * Pattern

```

```

0227 00229 D2        CON(2) BB45
0228 0022E C          CON(1) 12
0229 0022C
0230 0022C          * Move: [b[e]] M [file>]
0231 0022C 22        CON(2) 34
0232 0022E D2        CON(2) +teHL07 Message # 35
0233 00230 5          CON(1) 5
0234 00231 D4F6E756  NIBASC 'Move: '
0235 00239 A302
0236 0023D 0          CON(1) 13
0237 0023E B2        CON(2) BB43
0238 0023F C          CON(1) 13
0239 00241 C4963747  NIBASC 'M [ '
0240 00242 0          CON(1) 13
0241 00244 0          CON(2) BB44
0242 00246 D6        CON(1) 0
0243 0024D C          NIBASC ']'
0244 0024E           CON(1) 12
0245 0024E          * Print: [b[e]] P [n][N]
0246 0024E D1        CON(2) 29
0247 00250 42        CON(2) +teHL08 Message # 36
0248 00252 6          CON(1) 6
0249 00253 052796E5  NIBASC 'Print: '
0250 00258 47A302
0251 00251 0          CON(1) 13
0252 00252 B2        CON(2) BB43
0253 00254 0          CON(1) 0
0254 00255 05        NIBASC 'P'
0255 00257 D          CON(1) 12
0256 00258 D2        CON(2) BB45
0257 0025A C          CON(1) 12
0258 0025B
0259 0025B          * Quit: Q
0260 0025B 41        CON(2) 20
0261 0025D 52        CON(2) +teHL09 Message # 37
0262 0025F 6          CON(1) 6
0263 00270 15573647  NIBASC 'Quit: Q'
0264 00278 A30215
0265 0027E C          CON(1) 12
0266 0027F
0267 0027F          * Replace: [b[e]] [?] R/str1/str2[/]
0268 0027F 24        CON(2) 66
0269 00281 52        CON(2) +teHL10 Message # 38
0270 00283 8          CON(1) 8
0271 00284 25560765  NIBASC 'Replace:'
0272 00286 153654A0
0273 00294 02        NIBASC ' '
0274 00296 D          CON(1) 13
0275 00297 B2        CON(2) BB44

```

```

0316 0033D 41        CON(2) 20
0317 0033F A2        CON(2) BB42 Message # 42
0318 00341 6          CON(1) 6
0319 00342 05164747  NIBASC 'Pattern'
0320 0034A 5627E6
0321 00350 C          CON(1) 12
0322 00351
0323 00351 41        * [b[e]]
0324 00353 B2        CON(2) 20
0325 00355 0          CON(2) BB43 Message # 43
0326 00356 0          CON(1) 0
0327 00356 05208556  NIBASC '[b[e]] '
0328 0035E 05D502
0329 00364 C          CON(1) 12
0330 00365 C
0331 00365 21        * <file>
0332 00366 C2        CON(2) 18
0333 00369 5          CON(2) BB44 Message # 44
0334 0036A C36696C6  CON(1) 5
0335 00372 56E3      NIBASC '<file>'
0336 00376 C          CON(1) 12
0337 00377
0338 00377 41        * [n][N]
0339 00379 D2        CON(2) 20
0340 00378 6          CON(2) BB45 Message # 45
0341 00378 6          CON(1) 6
0342 0037C 0285E605  NIBASC '[n][N]'
0343 00384 B5E405
0344 0038A C          CON(1) 12
0345 0038B
0346 0038B          NIBHEX FF Table terminator

```

```
BB42      0002A Abs 0033 - 0075 0086 0317  
BB43      0002B Abs 0034 - 0156 0171 0211 0223 0236 0251 0272 0287 0324  
BB44      0002C Abs 0055 - 0160 0175 0240 0312 0331  
BB45      0002D Abs 0036 - 0227 0255 0338  
FileNd    0003D Rel 0344  
=MSGTBL1 00000 Rel 0038  
MBASE     00010 Abs 0005 - 0006 0007 0008 0009 0010 0011 0012 0013 0014  
          0015 0016 0017 0018 0019 0020 0021 0022 0023  
          0024 0025 0026 0027 0028 0029 0030 0031 0039  
          0040  
NULL      00010 Abs 0006 - 0044  
=eXMEH    00011 Abs 0007 - 0051  
=teCONF   00019 Abs 0015 - 0127  
=teDELE   00018 Abs 0014 - 0116  
=teEOF    00017 Abs 0013 - 0109  
=teHL00   0001C Abs 0015 - 0152  
=teHL01   0001D Abs 0019 - 0167  
=teHL02   0001E Abs 0020 - 0182  
=teHL03   0001F Abs 0021 - 0189  
=teHL04   00020 Abs 0022 - 0198  
=teHL05   00021 Abs 0023 - 0207  
=teHL06   00022 Abs 0024 - 0219  
=teHL07   00023 Abs 0025 - 0232  
=teHL08   00024 Abs 0026 - 0247  
=teHL09   00025 Abs 0027 - 0260  
=teHL10   00026 Abs 0028 - 0267  
=teHL11   00027 Abs 0029 - 0283  
=teHL12   00028 Abs 0030 - 0296  
=teHL13   00029 Abs 0031 - 0304  
=teICMD   00012 Abs 0002 - 0062  
=teICMP   00014 Abs 0010 - 0030  
=teIPAT   00013 Abs 0009 - 0071  
=teFEYS   0001A Abs 0016 - 0136  
=tePEOF   00016 Abs 0012 - 0101  
=tePLIN   00015 Abs 0011 - 0091  
=teVCMD   0001B Abs 0017 - 0143
```

Source : msg1.as
Object : obj/msg1.a
Listing : list/msg1.al
Date : Sat Aug 12 17:40:51 1989

```
0001 00000  
0002 00000  
* Fichier g n r  automatiquement par amg  
* et modifi  automatiquement par chmig  
*  
0003 00000  
0004 00000  
0005 00000  
0006 00000  
0007 00000  
0008 00000  
0009 00000  
=MSGTBL2 0010 00000 20  
          0011 00002 20  
          0012 00004  
          0013 00004  
* FIND  
          0014 00004 01  
          0015 00006 10  
          0016 00008 4  
          0017 00009 6494E444  
          00011 02  
          0018 00013 C  
          0019 00014  
          0020 00014  
* Not Found  
          0021 00014 21  
          0022 00016 20  
          0023 00018 8  
          0024 00019 E4F64702  
          00021 64F657E6  
          0025 00029 46  
          0026 00028 C  
          0027 0002C  
          0028 0002C FF  
          MBASE EQU 1  
          NULL EQU (MBASE)+0 FIND  
          =eNFND EQU (MBASE)+1 Not Found  
          CON(2) (MBASE)+1 Lowest message #  
          CON(2) (MBASE)+1 Highest message #  
          CON(2) 16  
          CON(2) NULL Message # 1  
          CON(1) 4  
          NIBASC 'FIND '  
          CON(1) 12  
          CON(2) 24  
          CON(2) =eNFND Message # 2  
          CON(1) 8  
          NIBASC 'Not Foun'  
          NIBASC 'd'  
          CON(1) 12  
          NIBHEX FF Table terminator
```

Errors : 000
Areuh Assembler/Linker V2.4. (c) P. David & J. Taillandier 1988 Paris, France

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

Source : msg2.as
Object : obj/msg2.a
Listing : list/msg2.al
Date : Sat Aug 12 17:40:53 1989
Errors : 000

```

0001 00000 TITLE Editeur, décomposition des cmd <xdec.as>
0002 00000
0003 00000
0004 00000
0005 00000
0006 00000
0007 00000
0008 00000
0009 00000
0010 00000
0011 00000
0012 00000
0013 00000
0014 00000
0015 00000
0016 00000
0017 00000
0018 00000
0019 00000
0020 00000
0021 00000
0022 00000
0023 00000
0024 00000
0025 00000
0026 00000
0027 00000
0028 00000
0029 00000
0030 00000
0031 00000
0032 00000
0033 00000
0034 00000
0035 00000
0036 00000
0037 00000
0038 00000
0039 00000
0040 00000
0041 00000
0042 00000
0043 00000
0044 00000
0045 00000
0046 00000
0047 00000
0048 00000
0049 00000
0050 00000

```

• Variables temporaires
 tmp EQU (*SRTCH)-1
 val EQU (tmp)-5 5 q dans expr
 signe EQU (tmp)-6 1 q dans expr
 sauve EQU *SMTRO 20 q pendant recherche
 debut EQU (*SMTRO)+20 5 q pendant "expr"

• skip
 • But: sauter les blancs
 • Entree:
 • - D1 = chaîne
 • - D(A) = nb de caractères
 • Sortie:
 • Abime: A(B), C(B)
 • Appelle: lookah, getchr
 • Niveaux: 1
 • Historique:
 • 88/05/14: PD/JT conception & codage
 • 88/05/15: PD/JT isolement dans un module séparé

```

0030 00000 S102 =skip LCASC *
0031 00004 7E10 skip10 GOSUB =lookah
0032 00008 400 = RTNC Plus de caractère
0033 00008 966 = ZARC B
0034 0000E 00 RTNYES
0035 00010 7300 GOSUB =getchr
0036 00014 SFE GONC skip10 B.E.T.

```

• lookah, getchr
 • But: lire un caractère, et passer au suivant (getchr)
 • Entree:
 • - D1 = chaîne
 • - D(A) = nb de caractères
 • Sortie:
 • - Cy = 1 : fin de la chaîne
 • - Cy = 0, A(B) = caractère lu
 • Abime: A(B)
 • Appelle:
 • Niveaux: 0

```

0101 0003F 147 C=DAT1 A
0102 00042 D7 D=C A D(A) := longueur
0103 00044 1000 D1=(2) =PCMD
0104 00048 147 C=DAT1 A
0105 0004B 135 D1=C D1 := caractère à lire
0106 0004E
0107 0004E
0108 0004E
0109 0004E 7EAF
0110 00052 700F GOSUB =skip
0111 00056 400 GOSUB =lookah
0112 00059 RTNC Cy = 1 : rien trouvé
0113 00059
0114 00059
0115 00059 D2 C=0 A
0116 0005B 108 R0=C nblignes := 0
0117 0005E
0118 0005E
0119 0005E
0120 0005E 94A ?C=0 S Si si, ça tient dans C(S) !
0121 00061 B4 G0YES dec100 Mode HP-UX
0122 00063
0123 00063
0124 00063
0125 00063
0126 00063
0127 00063
0128 00063
0129 00063
0130 00063 70F0 GOSUB linenb
0131 00067 404 GOC dec050 ldebut non reconnu
0132 0006A
0133 0006A
0134 0006A
0135 0006A
0136 0006A 101 R1=A R1 := ldebut
0137 0006D 7623 GOSUB nbli** nblignes**
0138 00071
0139 00071 788F GOSUB =skip
0140 00075 70AF GOSUB =lookah
0141 00079 4E2 GOC dec050 EOL
0142 0007C 31C2 LC(2)
0143 00080 966 ZARC B
0144 00083 71 G0YES dec020 virgule non reconnue
0145 00085
0146 00085 7E8F GOSUB =getchr passer la virgule
0147 00088
0148 00088
0149 00088
0150 00088

```

• Y-a-t'il quelque chose dans la ligne de commande ?
 • GOSUB =skip
 • GOSUB =lookah
 • RTNC Cy = 1 : rien trouvé
 • Il y a quelque chose.
 • C=0 A
 • R0=C nblignes := 0
 • Mode HP-71 ou HP-UX ?
 • ?C=0 S Si si, ça tient dans C(S) !
 • G0YES dec100 Mode HP-UX
 • Mode HP-71
 • [ldebut [[.] lfin]] [?] [cmd] [param]
 • GOSUB linenb
 • GOC dec050 ldebut non reconnu
 • [ldebut [[.] lfin]] [?] [cmd] [param]
 • R1=A R1 := ldebut
 • GOSUB nbli** nblignes**
 • GOSUB =skip
 • GOSUB =lookah
 • GOC dec050 EOL
 • LC(2)
 • ZARC B
 • G0YES dec020 virgule non reconnue
 • GOSUB =getchr passer la virgule
 • [ldebut [[.] lfin]] [?] [cmd] [param]

```

0051 00017
0052 00017
0053 00017
0054 00017
0055 00017
0056 00017 SAB
0057 0001A 00
0058 0001C CF
0059 0001E 14B
0060 00021 1C1
0061 00024 01
0062 00026
0063 00026 2AB
0064 00029 00
0065 0002B 14B
0066 0002E 01
0067 00030
0068 00030
0069 00030
0070 00030
0071 00030
0072 00030
0073 00030
0074 00030
0075 00030
0076 00030
0077 00030
0078 00030
0079 00030
0080 00030
0081 00030
0082 00030
0083 00030
0084 00030
0085 00030
0086 00030
0087 00030
0088 00030
0089 00030
0090 00030
0091 00030
0092 00030
0093 00030
0094 00030
0095 00030 IF00000
0096 00037 1574
0097 0003B
0098 0003B
0099 0003B
0100 0003B 1000

```

• Historique:
 • 88/05/14: PD/JT conception & codage
 • 88/05/15: PD/JT isolement dans un module séparé

```

0056 00017 SAB =getchr ?D=0 A
0057 0001A 00 RTNYES
0058 0001C CF D=D-1 A
0059 0001E 14B A=DAT1 B
0060 00021 1C1 D1=D1- 2
0061 00024 01 RTN Cy = 0
0062 00026
0063 00026 2AB =lookah ?D=0 A
0064 00029 00 RTNYES
0065 0002B 14B A=DAT1 B
0066 0002E 01 RTN Cy = 0

```

• keddec
 • But: décomposer une ligne de commande
 • Entree:
 • - LCMD, PCMD positionnés
 • Sortie:
 • - Cy = 1 : rien trouvé
 • - Cy = 0 : trouve une commande (éventuellement nulle)
 • A(B) = cmd (en majuscule, 0xff si pas de commande)
 • D(A) = longueur de PARAM
 • D1 = adresse de PARAM
 • R0 = nblignes (0, 1 ou 2)
 • R1 = ldebut
 • R2 = lfin
 • R3 = query (1 si '?' reconnu avant la commande)
 • Attention: PCMD et LCMD ne sont pas bien positionnés en
 • sortie. Pour cela, il faut que chaque commande appelle
 • setCMD pour stocker D1 et D(A) dans CMD.
 • Abime: A, C, D, D1
 • Appelles: skip, getchr, lookah, linenb, nbli**, expr
 • Niveaux: 3
 • Historique:
 • 88/05/15: PD/JT conception & codage
 • 88/10/16: PD/JT retrait de la " " obligatoire dans HP-UX

```

0095 00030 IF00000 =keddec D1=(5) =MODE71
0096 00037 1574 <=DAT1 5 <(5) := mode71 pour tout à l'heure
0097 0003B
0098 0003B
0099 0003B
0100 0003B 1000 D1=(2) =LCMD

```

• Positionne D(A) et D1 pour utilisation de getchr et lookah

```

0151 00089 77D0 GOSUB linenb
0152 0008D 531 GONC dec030 lfin reconnu
0153 00090 330000 LC(4) =MSPAR "5."
0154 00096 68B2 GOTO erreur
0155 0009A
0156 0009A 76C0 dec020 GOSUB linenb
0157 0009E 490 GOC dec050 lfin non reconnu
0158 000A1
0159 000A1
0160 000A1
0161 000A1
0162 000A1 102 dec030 R2=A R2 := lfin
0163 000A4 7FE2 GOSUB nbli** nblignes**
0164 000A8 6560 dec050 GOTO dec200
0165 000AC
0166 000AC
0167 000AC
0168 000AC
0169 000AC
0170 000AC
0171 000AC
0172 000AC
0173 000AC
0174 000AC
0175 000AC
0176 000AC
0177 000AC
0178 000AC
0179 000AC 1B00000
0180 000B3 142
0181 000B6 1B00000
0182 000BD 140
0183 000C0
0184 000C0
0185 000C0 7631 GOSUB expr
0186 000C4 494 GOC dec200 ldebut non reconnu
0187 000C7 101 R1=A ldebut
0188 000CA 79C2 GOSUB nbli** nblignes**
0189 000CE
0190 000CE
0191 000CE
0192 000CE 111
0193 000D1 1B00000 D0=(5) debut debut := R1
0194 000D8 140 DAT0=A A
0195 000DB
0196 000DB
0197 000DB
0198 000DB
0199 000DB
0200 000DB

```

• Mode HP-UX
 • dec100
 • [ldebut [[.] lfin]] [?] [cmd] [param]
 • ajout du 88/10/07 :
 • ligne de départ de la recherche = courante
 • D0=(5) =COUR
 • A=DAT0 A
 • D0=(5) debut debut := COUR
 • DAT0=A A
 • Fin de l'ajout
 • GOSUB expr
 • GOC dec200 ldebut non reconnu
 • R1=A ldebut
 • GOSUB nbli** nblignes**
 • ajout du 88/10/07 :
 • ligne de départ de la recherche = ldebut
 • A=R1
 • D0=(5) debut debut := R1
 • DAT0=A A
 • fin de l'ajout
 • [ldebut [[.] lfin]] [?] [cmd] [param]

```

0201 0000B 774F      GOSUB =lookah
0202 0000F 4E2       GOC   dec200
0203 000E2 31C2      LCASC  " "
0204 000E6 966       ?A#C  B
0205 000E9 71       GOYES dec120 virgule non reconnue
0206 000EB 782F      GOSUB =getchr
0207 000EF
0208 000EF
0209 000EF
0210 000EF
0211 000EF 7701      GOSUB expr
0212 000F3 531       GONC  dec130
0213 000F6 350000    LC(4) =#SPAR Missing parameter
0214 000FC 6552      GOTO  erreur
0215 00100
0216 00100 76F0      dec120 GOSUB expr
0217 00104 490       GOC   dec200 nombre non reconnu
0218 00107
0219 00107
0220 00107
0221 00107
0222 00107 102      dec130 R2=A      Ifin
0223 0010A 7982      GOSUB nbli** nblignes**
0224 0010E
0225 0010E
0226 0010E
0227 0010E
0228 0010E
0229 0010E
0230 0010E
0231 0010E
0232 0010E 7EEE      dec200 GOSUB =skip
0233 00112
0234 00112
0235 00112
0236 00112
0237 00112 AC2      C=B S query := 0
0238 00115
0239 00115 700F      GOSUB =lookah
0240 00119 421       GOC   dec300 EOL
0241 0011C 31F3      LCASC  "?"
0242 00120 966       ?A#C  B
0243 00123 90       GOYES dec300
0244 00125 7EEE      GOSUB =getchr pligne**
0245 00129 B46      C=C+1 S query := 1
0246 0012C 10B      dec300 R3=C query := 0 ou 1
0247 0012F
0248 0012F
0249 0012F
0250 0012F

```

```

0301 00173 21       GOYES ln010
0302 00175 7E9E      GOSUB =getchr
0303 00179 1B00000    D0=(5) =COUR return courante :
0304 00180 142       A=DAT0 A
0305 00183 01       PTN Cy = 0
0306 00185 3132      ln010 LCASC "g"
0307 00189 966       ?A#C  B
0308 0019C 21       GOYES ln020
0309 0019E 755E      GOSUB =getchr
0310 00192 1B00000    D0=(5) =DERN return dernière :
0311 00199 142       A=DAT0 A
0312 0019C 01       PTN Cy = 0
0313 0019E
0314 0019E
0315 0019E
0316 0019E
0317 0019E
0318 0019E
0319 0019E
0320 0019E
0321 0019E
0322 0019E
0323 0019E
0324 0019E
0325 0019E
0326 0019E
0327 0019E
0328 0019E
0329 0019E
0330 0019E
0331 0019E
0332 0019E
0333 0019E
0334 0019E
0335 0019E
0336 0019E
0337 0019E
0338 0019E
0339 0019E
0340 0019E
0341 0019E 742E      =nombre GOSUB =lookah
0342 001A2 400       RTNC Non, il n'y a rien...
0343 001A5 9F00000    GOSBVL =ORANGE
0344 001AC 400       RTNC Non, il n'y a rien...
0345 001AF D1       B=0 A
0346 001B1 717E      nbr010 GOSUB =lookah
0347 001B5 404       GOC   nbr090 EOL
0348 001B8 8F00000    GOSBVL =ORANGE
0349 001BF 463       GOC   nbr090 not in {0..3}
0350 001C2 715E      GOSUR =getchr

```

```

0251 0012F 7DCE      GOSUB =skip
0252 00133 7FEE      GOSUB =lookah
0253 00137 4B0       GOC   dec320 EOL
0254 0013A
0255 0013A
0256 0013A
0257 0013A 3183      LCASC  " ; "
0258 0013E 966       ?A#C  B
0259 00141 B0       GOYES dec350
0260 00143 AE0       dec320 A=0 B
0261 00146 A6C      A=A-1 B cmd := 0xff
0262 00149 481       GOC   dec900 fin de decomp (B.E.T.)
0263 0014C
0264 0014C
0265 0014C
0266 0014C 77CE      dec350 GOSUB =getchr A(B) := cmd (caractère existe)
0267 00150 AE8       B=A B protéger A(B)
0268 00153 79AE      GOSUB =skip
0269 00157 DA       A=C A
0270 00159 DC       ABEX A A(A) := cmd ; B(A) := len(param)
0271 0015B 8F00000    GOSBVL =CONVUC A(B) := cmd en majuscule
0272 00162
0273 00162
0274 00162
0275 00162
0276 00162
0277 00162 03      dec900 RTNCC Un seul point de retour: c'est beau
0278 00164
0279 00164
0280 00164
0281 00164
0282 00164
0283 00164
0284 00164
0285 00164
0286 00164
0287 00164
0288 00164
0289 00164
0290 00164
0291 00164
0292 00164
0293 00164
0294 00164
0295 00164
0296 00164
0297 00164 789E      linenb GOSUB =skip
0298 00168 7ABE      GOSUB =lookah
0299 0016C 31E2      LCASC  " "
0300 00170 966       ?A#C  B

```

```

0351 001C6
0352 001C6 D9       C=B A vérification que B < 65536
0353 001C8 24       P= 4
0354 001CA A82      C=0 P
0355 001CD 20       P= 0
0356 001CF 8A5      ?B#C ivarg
0357 001D2 A1       GOYES ivarg
0358 001D4
0359 001D4 DC       C=0 A D, on peut encore multiplier par 10
0360 001D6 3103      LCASC  "0"
0361 001DA 8EE      C=A-C B C(A) := A*(A(B))
0362 001DD C5       B=B+B A
0363 001DF D4       A=B A A(A) := 20
0364 001E1 C5       B=B+B A
0365 001E3 C5       B=B+B A B(A) := 30
0366 001E5 C8       B=B+B A B(A) := 100
0367 001E7 C1       B=B+C A B(A) := 100 + C(A)
0368 001E9 57C      GONC   nbr010 B.E.T
0369 001EC 3300000    ivarg LC(4) =#IVARG
0370 001F2 6F51      GOTO  erreur
0371 001F6 D4       nbr090 A=B A
0372 001F8 03       RTNCC Cy = 0
0373 001FA
0374 001FA
0375 001FA
0376 001FA
0377 001FA
0378 001FA
0379 001FA
0380 001FA
0381 001FA
0382 001FA
0383 001FA
0384 001FA
0385 001FA
0386 001FA
0387 001FA
0388 001FA
0389 001FA
0390 001FA
0391 001FA
0392 001FA
0393 001FA 7680      expr GOSUB expelm
0394 001FE 400       RTNC nombre non reconnu
0395 00201 1B00000    expr00 D0=(5) val val := A(A)
0396 00208 140       DATE=A A
0397 0020B 71FD      GOSUB =skip
0398 0020F 731E      GOSUB =lookah
0399 00213 446       GOC   expr99 sortin enfin
0400 00216 AC2      C=0 S

```



```

#401 #0219 3182 LCASC '*'
#402 #0210 952 7A#C B
#403 #0220 EA G0YES expr10 C(S) := #0 pour '*'
#404 #0222 A4E C<C-1 S C(S) := #F pour '*'
#405 #0225 3102 LCASC '*'
#406 #0229 956 7A#C B
#407 #022C C4 G0YES expr99 sortir si ni '*', ni '#'
#408 #022E 1800000 expr10 D0=(5) signe
#409 #0235 1544 DATA#C S signe := 0 si '*', #F si '#'
#410 #0239 7A0D G0SUB #getchr passer l'opérateur
#411 #0230 7340 G0SUB expelm
#412 #0241 500 G0NC expr20 ok...
#413 #0244 3300000 LC(4) #eSYNTAX "5."
#414 #024A 6701 G0TO erreur
#415 #024E 1800000 expr20 D0=(5) signe
#416 #0255 1564 C=DATA# S
#417 #0259 1900 D0=(2) val
#418 #0250 146 C=DATA# A
#419 #0260 94A 7C#0 S
#420 #0263 C0 G0YES expr30 '*'
#421 #0265 DE ACEX A
#422 #0267 EA A#A-C A '*'
#423 #0269 579 G0NC expr00 si tout va bien
#424 #026C 470 G0C lvarg B.E.T.
#425 #026F CA A#A-C A
#426 #0271 5F8 G0NC expr00 si tout va bien
#427 #0274 677F lvarg G0TO B.E.T.
#428 #0276 G0YES
#429 #0278 1800000 expr99 D0=(5) val
#430 #027F 142 A=DATA# A
#431 #0282 03 RTNCC Ok, ca a marché...
#432 #0284
#433 #0284
#434 #0284
#435 #0284
#436 #0284
#437 #0284
#438 #0284
#439 #0284
#440 #0284
#441 #0284
#442 #0284
#443 #0284
#444 #0284
#445 #0284
#446 #0284
#447 #0284
#448 #0284
#449 #0284
#450 #0284

```

- expelm
- But: analyser un élément d'expression à la syntaxe HF-UX
- Entree:
 - D1 = ' chaîne
 - D(0) = nb de caractères
 - debut = numéro de la première ligne de la recherche
- Sortie:
 - Cy = 1 : non reconnu
 - Cy = 0, A(A) = nombre reconnu
- Abime:
- Appelle: skip, lookah, GETSRC, getchr
- Niveaux:
- Historique:
 - 88/05/24: PD/JI conception & codage
 - 88/10/07: PD/JI ajout de "debut" dans la recherche
 - 88/10/15: PD/JI modification pour nouveau GETSRC

```

#501 #030C
#502 #030C 1800000 D0=(5) debut
#503 #0313 142 A=DATA# A
#504 #0316 E4 A#A+1 A A(A) := debut + 1
#505 #0318 1800000 D0=(5) #DERN
#506 #031F 146 C=DATA# A C(A) := dernière
#507 #0322 886 7A#C A
#508 #0325 00 G0YES ex1035
#509 #0327 8E0000 G0SUBL #SRCLIN
#510 #032D 482 G0C ex1040 match found
#511 #0330
#512 #0330 D0
#513 #0332 E4 ex1035 A#0 A
#514 #0334 1800000 D0=(5) debut A(A) := 1
#515 #0338 146 C=DATA# A C(A) := debut
#516 #033E 886 7A#C A
#517 #0341 00 G0YES notfnd
#518 #0343 8E0000 G0SUBL #SRCLIN
#519 #0349 C0 G0C ex1040 match found
#520 #034C
#521 #034C 3300000 notfnd LC(4) (=id) (=NFND) "Not Found"
#522 #0352 6000 erreur G0TO #xederr aie aie aie...
#523 #0356
#524 #0356
#525 #0356
#526 #0356
#527 #0356
#528 #0358 101
#529 #0359 8E0000
#530 #035F 111
#531 #0362
#532 #0362 AF2
#533 #0365 1800000 C#0 W on en profite pour nettoyer
#534 #036C 146 D0=(5) sauve R0(A)
#535 #036F 108 C=DATA# A
#536 #0372 164 R0#C
#537 #0375 146 D0#D0# 5 R1(A)
#538 #0378 109 R1#C
#539 #037B 164 D0#D0# 5 D1
#540 #037E 146 C=DATA# A
#541 #0381 135 D1#C
#542 #0384 164 D0#D0# 5 D(A)
#543 #0387 146 C=DATA# A
#544 #038A 07 D#C A
#545 #038C
#546 #038C 24 P# 4 restaurer 5 niveaux
#547 #038E 8F0000 G0SBVL #RSTK#R
#548 #0395
#549 #0395 03 RTNCC on a trouvé !
#550 #0397

```

- Match found
- A(A) = ligne trouvée
- ex1040 R1=A Sauvegarde de A(A)
- G0SUBL #ENDPOS Achever le buffer (arghhh)
- A#R1 Restauration de A(A)
- C#0 W on en profite pour nettoyer
- D0=(5) sauve R0(A)
- C=DATA# A
- R0#C
- D0#D0# 5 R1(A)
- R1#C
- D0#D0# 5 D1
- C=DATA# A
- D1#C
- D0#D0# 5 D(A)
- C=DATA# A
- D#C A
- P# 4 restaurer 5 niveaux
- G0SBVL #RSTK#R
- RTNCC on a trouvé !

```

#451 #0284
#452 #0284
#453 #0284 7870 expelm G0SUB #skip
#454 #0280 7A9D G0SUB #lookah
#455 #028C 31E2 LCASC '*'
#456 #0290 966 7A#C B
#457 #0293 21 G0YES ex1010
#458 #0295 7E7D G0SUB #getchr
#459 #0299 1800000 D0=(5) #COUR return courante :
#460 #02A0 142 A=DATA# A
#461 #02A3 01 RTN Cy = 0
#462 #02A5 3142 ex1010 LCASC '*'
#463 #02A9 956 7A#C B
#464 #02AC 21 G0YES ex1020
#465 #02AE 756D G0SUB #getchr
#466 #02B2 1800000 D0=(5) #DERN return dernière :
#467 #02B9 142 A=DATA# A
#468 #02BC 01 RTN Cy = 0
#469 #02BE 31F2 ex1020 LCASC '*'
#470 #02C2 952 7A#C B
#471 #02C5 58 G0YES ex1030
#472 #02C7 65DE G0TO #nombre "default"
#473 #02C8
#474 #02C8
#475 #02C8
#476 #02C8
#477 #02C8
#478 #02C8 24 P# 4 restaurer 5 niveaux
#479 #02CD 8F0000 G0SBVL #RSTK#R
#480 #02D4
#481 #02D4 1800000 D0=(5) sauve qui peut !
#482 #02D8 118 C#R0 R0(A)
#483 #02DE 144 DATA# A
#484 #02E1 164 D0#D0# 5 R1(A)
#485 #02E4 119 C#R1
#486 #02E7 144 DATA# A
#487 #02EA
#488 #02EA 8E0000 G0SUBL #GETSRC compiler la chaîne de recherche
#489 #02FA 416 G0C erreur
#490 #02FD 702D G0SUB #getchr passer le délimiteur (88/10/15)
#491 #02F7
#492 #02F7
#493 #02F7
#494 #02F7
#495 #02F7 1800000 D0=(5) sauve#10
#496 #02FE 137 #DIE#
#497 #0301 144 DATA# A
#498 #0304 164 D0#D0# 5
#499 #0307 08 C#C A
#500 #0309 144 DATA# A

```

- Recherche : /.../
- R0#R# = caractéristiques du buffer
- D1 et D(A) réactualisés
- D0=(5) sauve#10
- #DIE#
- DATA# A
- D0#D0# 5
- C#C A
- DATA# A

```

#551 #0397
#552 #0397
#553 #0397
#554 #0397
#555 #0397
#556 #0397
#557 #0397
#558 #0397
#559 #0397
#560 #0397
#561 #0397
#562 #0397
#563 #0397
#564 #0397
#565 #0397 128 nbli# CR#EX
#566 #039A E6 C#C+1 A
#567 #039C 128 CR#EX
#568 #039F 01 RTN
#569 #03A1

```

- nbli#
- But: incrémenter R0
- Entree:
 - R0
- Sortie:
 - R0 := R0+1
- Abime: (C#)
- Niveaux: 0
- Historique:
 - 88/05/15: PD/JI conception & codage
- nbli# CR#EX
- C#C+1 A
- CR#EX
- RTN

```

0571 003A1
0572 003A1
0573 003A1
0574 003A1
0575 003A1
0576 003A1
0577 003A1
0578 003A1
0579 003A1
0580 003A1
0581 003A1
0582 003A1
0583 003A1
0584 003A1
0585 003A1
0586 003A1
0587 003A1
0588 003A1
0589 003A1
0590 003A1
0591 003A1
0592 003A1
0593 003A1
0594 003A1
0595 003A1
0596 003A1
0597 003A1
0598 003A1
0599 003A1
0600 003A1
0601 003A1
0602 003A1
0603 003A1
0604 003A1
0605 003A1
0606 003A1
0607 003A1
0608 003A1
0609 003A1
0610 003A1
0611 003A1
0612 003A1
0613 003A1
0614 003A1
0615 003A1
0616 003A1
0617 003A1
0618 003A1
0619 003A1
0620 003A1

```

.....

- GETFIL
- But: lire un nom de fichier, chercher ce fichier ou le créer selon les flags.
- Entree:
 - - D1 = chaîne
 - - D(A) = nb de caractères
 - - sDELET = 0 : cas C/M
 - - sDELET = 1 : cas D ou X
 - - sXCHG = 1 : cas X
 - - sXCHG = 0 : cas D
- Sortie:
 - Cy = 1 : erreur
 - C(3-0) = numéro d'erreur
 - Cy = 0 : pas d'erreur
 - EXTFIL = adresse du header du fichier
 - sCREAT = 1 : fichier crée
 - Abime: A-D, R0-R3, D0, D1, S0-S9, STMTxx, FUNCxx
 - Appelle: ADHEAD, FILX0\$, FINDF+, CREFT+
 - Niveaux: 6 (CREFT+)
 - Algorithme:
 - création d'une chaîne sur la M.S. contenant le nom
 - (la chaîne s'arrête sur '*', ':' ou EOL)
 - si DELET = 1
 - alors
 - si XCHG = 1
 - alors plus := true
 - sinon plus := (caractère lu == '*')
 - fin si
 - fin si
 - FILX0\$
 - FINDF+
 - si il existe
 - alors
 - si DELET=1 et not plus
 - alors erreur (File Exists)
 - fin si
 - sinon
 - si DELET=0
 - alors erreur (File Not Found)
 - sinon CREFT+
 - fin si
 - vérifier que le type est TEXT
 - retourner l'adresse dans EXTFIL
 - Historique:
 - 85/10/31: PD/JT conception & codage
 - 88/11/11: PD/JT modification pour la commande 'X'

```

0671 0040B 2A2
0672 0040E 5F
0673 00410 134
0674 00413
0675 00413
0676 00413
0677 00413 79EB
0678 00417
0679 00417
0680 00417
0681 00417
0682 00417
0683 00417
0684 00417
0685 00417
0686 00417
0687 00417
0688 00417 859
0689 0041A F1
0690 0041C 07A
0691 0041F A1
0692 00421
0693 00421
0694 00421
0695 00421
0696 00421 710C
0697 00425 3182
0698 00429 966
0699 0042C D0
0700 0042E
0701 0042E
0702 0042E
0703 0042E 75EB
0704 00432 7ACB
0705 00436 85A
0706 00439
0707 00439
0708 00439
0709 00439 137
0710 0043C 1F00000
0711 00443 145
0712 00446 174
0713 00449 DB
0714 0044B 145
0715 0044E
0716 0044E
0717 0044E
0718 0044E
0719 0044E
0720 0044E

```

?

- A
- GOYES rtncc sortie sans erreur
- D0=C sauvegarde temporaire dans D0
- Le nom du fichier a été reconnu et est dans la M.S.
- GOSUB =skip
- Reconnaitre le '*' seulement si 'D' (et pas C/M ou X)
- si sDELET = 1
- alors
- si sXCHG = 1
- alors plus := true
- sinon plus := '*' reconnu ? true : false
- fin si
- fin si
- ST=0 =sDELET cas 'D' ?
- GOYES GF150 non
- ST=1 =sXCHG cas 'X' ?
- GOYES GF150 out, donc plus := true (deja vrai)
- On sait que XCHG = 0. Donc sPLUS = 0
- l'instruction "ST=0 sPLUS" est inutile"
- GOSUB =lookah
- LCASC '*'
- ?A=C B caractère suivant = '*' ?
- GOYES GF150 non
- '*' reconnu
- GOSUB =getchr passer le '*'
- GOSUB =skip
- ST=1 sPLUS plus := true
- Sauver les meubles !!! (D1 et D(A))
- GF150 CODEX
- D1=(5) =STMT00
- DAT=C A STMT00=0 := D1
- D1=D1+ 5
- C=D A
- DAT=C A STMT00+05 := D(A)
- D0 = haut de la M.S.
- R1 = bas de la M.S.
- sDELET = 1 si commande 'D', 0 si commande 'C' ou 'M'
- sPLUS = 1 si commande 'D' et '*' reconnu

```

0621 003A1
0622 003A1
0623 003A1
0624 003A1
0625 003A1
0626 003A1
0627 003A1
0628 003A1
0629 003A1 84B
0630 003A4 1B00000
0631 003AB D2
0632 003AD 144
0633 003B0
0634 003B0 1B00000
0635 003B7 146
0636 003BA D5
0637 003BC 164
0638 003BF 146
0639 003C2 134
0640 003C5 109
0641 003C8
0642 003C8 7A5C
0643 003CC 483
0644 003CF 3582B302
0645 003D7 25
0646 003D9 8F00000
0647 003E0 542
0648 003E3
0649 003E3
0650 003E3
0651 003E3 703C
0652 003E7
0653 003E7 181
0654 003EA 136
0655 003ED 80
0656 003F0 80
0657 003F2 136
0658 003F5 148
0659 003F8 5FC
0660 003FB
0661 003FB 330000
0662 00401 02
0663 00403
0664 00403 03
0665 00405
0666 00405
0667 00405
0668 00405
0669 00405 136
0670 00408 111

```

.....

- sDELET EQU 09 flag D (par opposition à C/M)
- sXCHG EQU 10 cas D ou X ? (après : '*' reconnu)
- sCREAT EQU 11 fichier créé par nos soins
- sPLUS EQU =sXCHG flag plus utilisé apres son test
- GETFIL ST=0 =sCREAT fichier non encore créé
- D0=(5) =EXTFIL par défaut, EXTFIL := 0
- C=0 A
- DAT=0 A
- D0=(5) =AVMHS
- C=DAT0 A
- B=C A B(A) := AVMHS
- D0=D0+ 5 D0=(5) =AVMHE
- C=DAT0 A
- D0 := début de la M.S.
- R1=C pour ADHEAD tout à l'heure
- GF00 GOSUB =lookah
- GOC GF100 EOL
- LCASC '*' ensemble (espace, ':', '*')
- P= 3=2-1
- GOSBVL =MEMBER n'abime que C(MP)
- GONC GF100 byte in set
- Mettre le caractère dans la M.S. (pointée par D0)
- GOSUB =getchr passer le caractère
- Ce qui suit est très fortement inspiré de STKCHR (#18505)
- D0=D0+ 2
- CODEX
- ?C=B A
- GOYES GF050 Memerr
- CODEX
- DAT0=A B
- GONC GF010 B.E.T.
- GF00 LC(4) =eMEM
- RTNSC Beeeeeeep !
- rtncc RTNSC
- Fin de la chaîne. Si la longueur est nulle, alors sortir sans avoir rien reconnu.
- GF100 CODEX C(A) := haut de la M.S.
- A=R1 A(A) := bas de la M.S.

```

0721 0044E
0722 0044E
0723 0044E
0724 0044E
0725 0044E
0726 0044E
0727 0044E
0728 0044E 136
0729 00451 135
0730 00454 1B00000
0731 0045B 146
0732 0045E D7
0733 00460 850
0734 00463
0735 00463
0736 00463
0737 00463
0738 00463
0739 00463
0740 00463 8F00000
0741 0046A
0742 0046A 8F00000
0743 00471 4A0
0744 00474
0745 00474 330000
0746 0047A 02
0747 0047C
0748 0047C
0749 0047C
0750 0047C
0751 0047C
0752 0047C
0753 0047C
0754 0047C
0755 0047C
0756 0047C
0757 0047C
0758 0047C
0759 0047C
0760 0047C
0761 0047C
0762 0047C
0763 0047C
0764 0047C
0765 0047C
0766 0047C 1B00000
0767 00483 1507
0768 00487 16F
0769 0048A AFB
0770 0048D 812

```

- STMT0(4-0) = sauvegarde de D1
- STMT0(5-5) = sauvegarde de D(A)
- Construire le M.S. header et chercher le fichier
- CODEX
- D1=C D1 = haut de la M.S.
- D0=(5) =AVMHS
- C=DAT0 A
- D=C A D(A) := AVMHS
- ST=1 0 rtn desired
- D1 = top of M.S.
- R1 = bottom of M.S.
- D(A) = AVMHS
- S0=1
- GOSBVL =ADHEAD D1 = string header
- GOSBVL =FILX0\$
- GOC GF160 pas d'erreur
- ivfsc LC(4) =sSPEC
- RTNSC
- GF160
- Spécificateur valide.
- A(W) = file name ou 0
- D(S) = F if no device specified
- 0 if :MIN
- 1 if :PORT
- 7 if :CARD
- La vérification du spécificateur sera faite par FINDF.
- Maintenant, il faut chercher le fichier.
- A(W) = le nom
- D(S) et D(B) = device specifier
- Sauvegarde de A(W) et D(S)-D(B)
- dans STMT0(10-25) et STMT0(28-26)
- D0=(5) 10=sSTMT0
- DAT0=A W
- D0=D0+ 16
- C=D W
- CSLC C(2-0) = D(S) + D(B)

```

0771 00430 15C2      DATA=C 3
0772 00434
0773 00434
0774 00494
0775 00494
0776 00494 8F00000
0777 0049B 402
0778 0049E
0779 0049E
0780 0049E
0781 0049E
0782 0049E
0783 0049E 869
0784 004A1 70
0785 004A3 86A
0786 004A6 60
0787 004A8 6080
0788 004AC
0789 004AC 330000
0790 004BC 02
0791 004B4 330000
0792 004BA 02
0793 004BC
0794 004BC
0795 004BC
0796 004BC
0797 004BC 869
0798 004BF 5F
0799 004C1
0800 004C1
0801 004C1
0802 004C1
0803 004C1 1B00000
0804 004C3 15E2
0805 004CC 816
0806 004CF AF7
0807 004D2
0808 004D2 02
0809 004D4 3192
0810 004D8 8F00000
0811 004DF 400
0812 004E2
0813 004E2 09
0814 004E4
0815 004E4
0816 004E4
0817 004E4
0818 004E4 8F00000
0819 004EB 400
0820 004EE

```

• Chercher le fichier
 • GOSBVL +FIND+
 GOC GF300 fichier n'existe pas
 • Le fichier existe
 • D1 = " header du fichier
 GF200 ?ST=0 =>DELET pour les commandes 'C' et 'M'
 G0YES GF205 le fichier doit exister
 ?ST=0 #PLUS si D et pas +, il doit pas exister
 G0YES fevist pas plus => erreur
 GF205 GOTO GF900 rallonge
 fevist LC(4) =>FEVST "File Exists"
 RTNSC
 fafnd LC(4) =>FNFND "File Not Found"
 RTNSC
 • Le fichier n'existe pas
 GF300 ?ST=0 =>DELET commande 'C' ou 'M' ?
 G0YES fnfnd oui : "File Not Found"
 • Le fichier n'existe pas, mais il faut le créer pour la
 commande "D"
 D=(5) 16*10=>STMR0
 C=DATA 3
 CSRC
 D=C W D(W) => comme apres FILX0\$
 C=0 A
 LC(2) 37*4 header + EOF mark
 GOSBVL +MEMCKL
 RTNC Not Enough Memory
 C=0 A C(A) => amount to check (MEMCKL)
 • C(A) = taille du buffer en nbs
 • D = conditions de sortie de FILX0\$
 GOSBVL =>CRETF+ Théoriquement, le lex ne doit pas bouger
 RTNC erreur !

```

=ADHEAD Extrn Ukn - 0740
=AVMEMS Extrn Ukn - 0634 0730
=CHKTXT Extrn Ukn - 0855
=CONVUC Extrn Ukn - 0271
=COUR Extrn Ukn - 0179 0303 0459
=CRETF+ Extrn Ukn - 0818
=DERN Extrn Ukn - 0310 0466 0505
=DRANGE Extrn Ukn - 0343 0348
=ENDPOS Extrn Ukn - 0529
=EXTFIL Extrn Ukn - 0630 0849
=FILX0$ Extrn Ukn - 0742
=FIND+ Extrn Ukn - 0776
File=nd 00559 Rel 0868 -
=GETFIL 00341 Rel 0629 -
=GETSRC Extrn Ukn - 0488
GF010 00308 Rel 0642 - 0559
GF030 00308 Rel 0651 - 0656
GF100 00405 Rel 0659 - 0543 0647
GF150 00439 Rel 0709 - 0659 0631 0699
GF160 0047C Rel 0748 - 0743
GF200 0049E Rel 0783 -
GF205 004AS Rel 0787 - 0784
GF300 004BC Rel 0797 - 0777
GF900 00529 Rel 0847 - 0787
GF910 00542 Rel 0860 -
Ivarg 00274 Rel 0427 - 0424
=LCMD Extrn Ukn - 0100
=MEMBER Extrn Ukn - 0646
=MEMCKL Extrn Ukn - 0810
=MODE71 Extrn Ukn - 0095
=PCMD Extrn Ukn - 0103
=R=RSK Extrn Ukn - 0479
=RSK+R Extrn Ukn - 0547
=SCRICH Extrn Ukn - 0086
=SRCLIN Extrn Ukn - 0509 0518
=STMR0 Extrn Ukn - 0011 0012 0710 0766 0803 0827 0860
debut Unkn Ukn 0012 - 0181 0193 0502 0514
dec020 0009A Rel 0156 - 0144
dec030 000A1 Rel 0162 - 0152
dec050 000AS Rel 0164 - 0131 0141 0157
dec100 000AC Rel 0170 - 0121
dec120 00100 Rel 0216 - 0205
dec130 00107 Rel 0222 - 0212
dec200 0010E Rel 0232 - 0164 0136 0202 0217
dec300 0012C Rel 0246 - 0240 0243
dec320 00143 Rel 0260 - 0253
dec350 0014C Rel 0266 - 0259
dec900 00162 Rel 0277 - 0262
=FEVST Extrn Ukn - 0789
=FSPEC Extrn Ukn - 0745

```

```

0821 004EE 858
0822 004F1
0823 004F1
0824 004F1
0825 004F1
0826 004F1 119
0827 004F4 1F00000
0828 004FB 1537
0829 004FF 135
0830 00502 1517
0831 00506 17F
0832 00509 351000004
0833 00511 1505
0834 00515 17F
0835 00518 174
0836 0051B 02
0837 0051D CE
0838 0051F 1503
0839 00523
0840 00523 119
0841 00526 135
0842 00529
0843 00529
0844 00529
0845 00529
0846 00529
0847 00529
0848 00529 137
0849 0052C 1F00000
0850 00533 145
0851 00536
0852 00536
0853 00536
0854 00536 135
0855 00539 8E00000
0856 0053F 400
0857 00542
0858 00542
0859 00542
0860 00542 1B00000
0861 00549 146
0862 0054C 115
0863 0054F 164
0864 00552 146
0865 00555 07
0866 00557 00
0867 00559
0868 00559

```

ST=1 =>CREAT fichier cree !
 • R1 = " début du nouveau fichier
 • manque le nom et le type
 C=R1
 D1=(5) 10*STMR0
 A=DATA1 W
 D1=C
 DATA1=A W nom du fichier
 D1=D1- 16
 LCHEX 40001 text + copy code
 DATA1=C 8
 D1=D1- 16
 D1=D1- 5
 C=0 A
 C=C-1 A C(C-0) => #FFFF
 DATA1=C 4 EOF mark
 C=R1 C(A) => " début du fichier
 D1=C " début du fichier
 • D1 = " header du fichier trouvé ou créé
 • STMR0(9-0) = sauvegarde de D1 et D1A)
 •
 GF900
 C=D1EX
 D1=(5) =>EXTFIL
 DATA1=C A
 • Vérifier que le fichier est bien TEXT
 D1=C
 GOSBVL =CHKTXT
 RTNC C(A) => #FTYPE
 • Restaurer les neubles (D1 et D1A)
 GF910 D=(5) =>STMR0
 /=DATA A
 D1=C
 D0=D0- 8
 /=DATA 8
 D=C A
 RTNC "sortie sans erreur"
 END

```

=FEVST Extrn Ukn - 0791
=IARG Extrn Ukn - 0369
=MEM Extrn Ukn - 0661
=MSPAR Extrn Ukn - 0153 0213
=NFND Extrn Ukn - 0521
=SYNTAX Extrn Ukn - 0413
erreur 00352 Rel 0522 - 0154 0214 0370 0414 0489
ex1010 002AS Rel 0462 - 0457
ex1020 002BE Rel 0469 - 0464
ex1030 002CB Rel 0474 - 0471
ex1035 00330 Rel 0512 - 0508
ex1040 00356 Rel 0528 - 0510 0519
expelm 00284 Rel 0453 - 0393 0411
expr 001FA Rel 0393 - 0185 0211 0216
expr00 00201 Rel 0395 - 0423 0426
expr10 0022E Rel 0408 - 0403
expr20 0024E Rel 0415 - 0412
expr30 0026F Rel 0425 - 0420
expr99 00276 Rel 0429 - 0399 0407
fevist 004AC Rel 0789 - 0786
fnfnd 0048A Rel 0791 - 0798
ggetchr 00017 Rel 0056 - 0035 0146 0206 0244 0266 0302 0309 0350 0410
+ 0458 0465 0490 0651 0703
+ 0521
Ivarg Extrn Ukn - 0369 - 0357 0427
lvfproc 00474 Rel 0745 -
looksh 00026 Rel 0053 - 0031 0110 0140 0201 0239 0252 0298 0341 0346
+ 0398 0454 0642 0696
linemb 00164 Rel 0297 - 0130 0151 0156
lnb010 00185 Rel 0306 - 0301
lnb020 0019E Rel 0313 - 0308
nombre 0019E Rel 0341 - 0472
nbl1++ 00397 Rel 0565 - 0137 0163 0188 0223
nbr010 00181 Rel 0346 - 0368
nbr030 001F6 Rel 0371 - 0347 0349
notfnd 0034C Rel 0521 - 0517
rtnc 00403 Rel 0664 - 0672
=CREAT 00008 Abs 0625 - 0629 0821
=DELET 00009 Abs 0623 - 0688 0783 0797
=CHKG 0000A Abs 0624 - 0627 0690
skip 00000 Rel 0030 - 0109 0139 0232 0251 0268 0297 0397 0453 0677
+ 0704
=PLUS 0000A Abs 0627 - 0705 0705
sauve Unkn Ukn 0011 - 0481 0495 0533
signe Unkn Ukn 0009 - 0408 0415
skipR 00004 Rel 0031 - 0036
tmp Unkn Ukn 0006 - 0002 0009
val Unkn Ukn 0008 - 0395 0417 0429
=xxddc 00030 Rel 0095 -
=xxderr Extrn Ukn - 0522

```

```

0001 00000 TITLE Editeur, boucle principale <xbcl.as>
0002 00000
0003 00000
0004 00000
0005 00000
0006 00000
0007 00000
0008 00000
0009 00000
0010 00000
0011 00000
0012 00000
0013 00000
0014 00000
0015 00000
0016 00000
0017 00000
0018 00000
0019 00000
0020 00000
0021 00000
0022 00000
0023 00000
0024 00000
0025 00000
0026 00000
0027 00000
0028 00000
0029 00000
0030 00000
0031 00000
0032 00000
0033 00000
0034 00000
0035 00000
0036 00000
0037 00000
0038 00000
0039 00000
0040 00000
0041 00000
0042 00000
0043 00000
0044 00000
0045 00000
0046 00000
0047 00000
0048 00000
0049 00000
0050 00000

```

TITLE Editeur, boucle principale <xbcl.as>

- Variables utilisées par les fonctions et par XEDIT
- Tout le reste de TRFMFBF est donc disponible

```

=MODE71 EQU 00+=TRFMFBF 1q si mode = HP-71
=QUERY EQU 01+=TRFMFBF 1q si mode = "?" pour S/R
=FCNTHP EQU 02+=TRFMFBF 58 quartets pour le scratch

```

- Variables utilisées par XEDIT seulement.

```

=BUFID EQU 02+=TRFMFBF 3q
=BUFAOR EQU 05+=TRFMFBF 5q
=FILADR EQU 10+=TRFMFBF 5q
=COUR EQU 15+=TRFMFBF 5q no de ligne courante
=DERN EQU 20+=TRFMFBF 5q no de la dernière ligne
=PCMD EQU 25+=TRFMFBF 5q pteur dans la cmd
=LCMD EQU 30+=TRFMFBF 5q longueur de la cmd
=FLAG1 EQU 35+=TRFMFBF 1q état du flag 1 à l'entrée
=RONLY EQU 36+=TRFMFBF 1q fichier read-only

```

- Tout le reste (à savoir 23 quartets) est disponible
- pour des variables temporaires dans XEDIT

```

=XEDTHP EQU 37+=TRFMFBF 23q : variables temporaires

```

- En particulier, la fonction GENRPLC\$ et la commande R
- utilisent 10 quartets pour stocker les coefficients "a" et
- "b" utilisés lors du remplacement. Voir la documentation
- CALCAB.
- Cette zone peut être utilisée par toute routine
- n'utilisant pas REPLIN ou REPLFIL.

```

=coeffa EQU (=SCRCH)-1A
=coeffb EQU (=coeffa)+5

```

- Diverses zones de scratch.

```

=TMP5 EQU =LDCSPC
=TMP3 EQU =STSAVE
=TMP1 EQU (=SCRCH)-1

```

```

0101 00030 8000000 GOVLNG <EXPECX Cy = 0 en sortie
0102 00037
0103 00037
0104 00037
0105 00037
0106 00037
0107 00037
0108 00037
0109 00037
0110 00037
0111 00037
0112 00037
0113 00037
0114 00037
0115 00037
0116 00037
0117 00037
0118 00037 00000
0119 00030 00000
0120 00041 850
0121 00044 6010
0122 00048
0123 00048 00000
0124 00040 00000
0125 00052 840
0126 00055 1F00000
0127 0005C 34C0000
0128 00063 8F00000
0129 0006A
0130 0006A 860
0131 0006D 01
0132 0006F 301
0133 00072 1F00000
0134 00079 1550
0135 0007D
0136 0007D
0137 0007D
0138 0007D
0139 0007D 8E0000
0140 00083
0141 00083
0142 00083
0143 00083
0144 00083 799F
0145 00087 443
0146 0008A 8F00000
0147 00091 137
0148 00094 1F00000
0149 0009B 145
0150 0009E C2

```

- XEDITe, TEDITe
- But: éditeur de texte
- Entree:
- le nom du fichier
- une chaîne de commande optionnelle
- Abime: tout
- Appelle: ????
- Niveaux: 7 niveaux sont autorisés pour les statements
- Historique:
- 88/05/13: PD/JT conception & codage
- 89/06/17: PD/JT retrait des paramètres largeur/hauteur

```

REL(5) =TEDITd
REL(5) =TEDITp
=TEDITw ST=1 0 Mode HP-71 := true
GOTO XED000
REL(5) =XEDITd
REL(5) =XEDITp
=XEDITw ST=0 0 Mode HP-71 := false
XED000 D1=(5) =TRFMFBF
LC(5) 60 50 quartets
GOSBVL =WIPOUT TRFMFBF rempli avec des 0
ZST=0 0 mode HP-71 ?
GOYES XED010 non : on saute
LC(1) 1
D1=(5) =MODE71
DAT1=C P MODE71 := 1
XED010
Analyse du fichier
GOSUBL =FILE1 Evalue la spécification
Analyse de la chaîne de commande
GOSUB nstexp Expression suivante
GOC XED050 EOL trouvé
GOSBVL =POPIS
CDIEX
D1=(5) =AVHEME
DAT1=C A AVHEME := " bas de la M.S.
C=C*A A ((A) := " début de la chaîne

```

- Constantes globales :

```

=FLIMSK EQU 20010 flag 1 = 1 dans les flags 3-8
=NCHDEU EQU 13 nombres de commandes (0-13)

```

```

poparg

```

- But : dépiler un nombre de la M.S. et le vérifier
- Entree:
- D1 = " top of M.S.
- Sortie :
- A(A) = C(A) = nombre dépilé (0 < nb <= 255)
- Abime: A, B(S), B(A), C(A), D(A)
- Appelle: RNDAXH
- Niveaux: 4 (RNDAXH)
- Historique:
- 88/05/13: PD/JT conception & codage

```

poparg GOSBVL =RNDAXH
GONC ivarg L = 0
7A=0 A
GOYES ivarg
C=0 A
C=A B
7A=C A
RINYES
ivarg GOVLNG =IVAERR

```

- nextxp
- But: évaluer la prochaine expression
- Entree:
- DR = " chaîne binétriee (sur tCOMMA ou #EOL)
- Sorties:
- Cy = 1 si #EOL reconnu
- Cy = 0 et expression sur la M.S. (A/W) = top 16 nibs)
- Abime: tous les registres, fonction scratch
- Appelle: EOLC\$, EXPECX
- Niveaux: 4 (<EXPECX)
- Historiques:
- 88/05/13: PD/JT conception & codage

```

nextxp A:DAT0 B
D0=000 =
GOSBVL =EOLC$
RPNV
#EOL trouvé

```

```

0151 000A0 CE
0152 000A2 CE
0153 000A4 1E0000
0154 000AA 145
0155 000AD AF2
0156 000B0 06
0157 000B2 81E
0158 000B5 1000
0159 000B9 145
0160 000BC
0161 000BC
0162 000BC
0163 000BC
0164 000BC 851
0165 000BF 852
0166 000C2 8E0000
0167 000C3
0168 000C8
0169 000C2
0170 000C8 8E0000
0171 000CE 1F00000
0172 000D5 1554
0173 000D9
0174 000D9 1E0000
0175 000DF 145
0176 000E2
0177 000E2 135
0178 000E5
0179 000E5 2E0000
0180 000EB 530
0181 000EE
0182 000EE 8000000 bserr GOVLNG =BSERR
0183 000F5
0184 000F5
0185 000F5
0186 000F5
0187 000F5
0188 000F5
0189 000F5
0190 000F5
0191 000F5 2E0000
0192 000FB
0193 000FB 02
0194 000FD E5
0195 000FF 1F00000
0196 00106 145
0197 00109
0198 00109
0199 00109
0200 00109

```

```

C=C-1 A
C=C-1 A
D1=(4) =PCMD
DAT1=C A PCMD := " premier caractère
C=0 W
C=A A
CSR8 (A) := longueur en octets
D1=(2) =LCMD
DAT1=C A LCMD := longueur en octets
Recherche du fichier, et copie éventuelle
XED050
ST=1 1 Copie éventuelle
ST=1 2 Création éventuelle
GOSUBL =FILE2
Le fichier est trouvé. On a son adresse dans C(A)
GOSUBL =CHKWRT Modifiable ?
D1=(5) =RONLY
DAT1=C S RONLY := 0 (write) 1..3 (read-only)
D1=(4) =FILADR Adresse de l'en-tete
DAT1=C A FILADR := adresse de l'en-tete
D1=C D1 := " file header
GOSUBL =CHKTXE Est-ce un fichier TEXT ?
GONC XED080 Oui

```

- bserr GOVLNG =BSERR
- XED080
- inibuf initialise le tableau de pointeurs dans le texte.
- Cette routine commence par désallouer le buffer dont le
- numéro est dans BUFID. Cela ne nous gêne pas car nous
- avons fait un WIPOUT sur toutes les variables au début.
- GOSUBL =inibuf
- C=0 A
- C+=1 A
- D1=(5) =COUR courante := 1
- DAT1=C A
- Mettre à 0 le flag 1 et sauvegarder son ancien état.

```

0201 00109 1E0000      DI=(4) =FLGREG.
0202 0010F 1530       A-DATI P
0203 00113 A88        B-A P      B(0) := ancienne valeur du flag 1
0204 00116 300        LC(1) =FLIMSA
0205 00119 0E0E       A-ASC P      A(0) := flags 0-3 sans le 1
0206 0011D 1510       DATI=A P
0207 00121
0208 00121 302        LC(1) =FLIMSA
0209 00124 0E05       C-C5B P      C(0) := flag 1 seulement
0210 00128 1E0000     DI=(4) =FLAG1
0211 0012E 1550       DATI=C P
0212 00132
0213 00132 6A80       GOTO =BOUCLE
0214 00136
0215 00136
0216 00136
0217 00136
0218 00136
0219 00136
0220 00136
0221 00136
0222 00136
0223 00136
0224 00136
0225 00136
0226 00136
0227 00136
0228 00136
0229 00136
0230 00136
0231 00136
0232 00136
0233 00136
0234 00136
0235 00136
0236 00136
0237 00136 1B0000     D0=(5) =DERN
0238 0013D 142       A-DAT0 A      A(A) := dernière
0239 00140 1900       D0=(2) =COUR
0240 00144 146       C-DAT0 A      C(A) := courante
0241 00147 8B2       ?>A nul500
0242 0014A 75        GOYES nul500 attendre le relachement de [ENDLINE]
0243 0014C
0244 0014C
0245 0014C
0246 0014C 8F0000     GOSBVL =OBCOLL C(A) := OUTBS
0247 00153 135       DI=C          DI := AVHERS
0248 0015E
0249 0015E 1B0000     D0=(5) =COUR
0250 0015D 142       A-DAT0 A      A(A) := courante

```

- vieuIn
- But: affiche la ligne en cours ou "[Eof]" tant que la touche [ENDLINE] est appuyée.
- Entree:
 - - COUR = numéro de la ligne courante
 - - DERN = numéro de la dernière ligne
- Sortie:
 - Abime: A-D, R0-R3, D0, D1, SIMTR0, AVHERS, TMP5
 - Appelle: OBCOLL, NumD01, AVS2DS, seek, dsplin, CRLFND
 - dspsmg, KYDN?
 - Niveaux: 6 (CRLFND)
 - Historique:
 - 88/11/13: PD/JT conception & codage
- vieuIn
- "numéro: ligne" ou "[Eof]" ?
- - Afficher le numero de ligne suivi d'un " :"

```

0301 001C9
0302 001C8 8E0000     GOSUBL =vedec
0303 001C6 581       GONC BCL200 Commande non nulle
0304 001C9
0305 001C9
0306 001C9
0307 001C9
0308 001C9
0309 001C9 85A       ?ST=0 10 Ce n'est pas le cas
0310 001CC 60        GOYES BCL100 donc affichage du prompt
0311 001CE
0312 001CE
0313 001CE
0314 001CE
0315 001CE 746F     GOSUB vieuIn
0316 001D2
0317 001D2
0318 001D2
0319 001D2
0320 001D2 8E0000   BCL100 GOSUBL =prompt
0321 001D8 85A       ST=1 10
0322 001D8 64EF     GOTO BCL000
0323 001D0
0324 001D0 31FF     BCL200 LC(2) -1 commande nulle ?
0325 001E3 962       ?A=C B
0326 001E6 1D        GOYES cmNULL
0327 001E8
0328 001E8
0329 001E8
0330 001E8
0331 001E8 1B0000     D0=(5) =FUNCR0
0332 001EF 08        C=D A
0333 001F1 144       DAT0=C A      FUNCR0 + 0 := D(A)
0334 001F4 164       D0=D0+ 5
0335 001F7 137       COIEX
0336 001FA 144       DAT0=C A      FUNCR0 + 5 := D1
0337 001FD 164       D0=D0+ 5
0338 00200 118       C=R0
0339 00203 144       DAT0=C A      FUNCR0 + 10 := R0
0340 00206 164       D0=D0+ 5
0341 00209 119       C=R1
0342 0020C 144       DAT0=C A      FUNCR0 + 15 := R1
0343 0020F
0344 0020F 8E0000   GOSUBL =poscmd
0345 00215 546       GONC invcmd "Invalid Cmd"
0346 00218
0347 00218
0348 00218
0349 00218
0350 00218

```

-
- Commande nulle. Si on vient d'afficher notre prompt, c'est le cas spécial de l'utilisateur qui appuie sur [ENDLINE] pour avoir la ligne courante à l'affichage.
- ?ST=0 10 Ce n'est pas le cas
- GOYES BCL100 donc affichage du prompt
- C'est le cas. Afficher la ligne tant que [ENDLINE] est appuyée.
- GOSUB vieuIn
- Afficher un prompt
- BCL100 GOSUBL =prompt
- ST=1 10
- GOTO BCL000
- BCL200 LC(2) -1 commande nulle ?
- ?A=C B
- GOYES cmNULL
- Une commande a été fournie. La chercher dans le message
- listant les commandes valides.
- D0=(5) =FUNCR0
- C=D A
- DAT0=C A FUNCR0 + 0 := D(A)
- D0=D0+ 5
- COIEX
- DAT0=C A FUNCR0 + 5 := D1
- D0=D0+ 5
- C=R0
- DAT0=C A FUNCR0 + 10 := R0
- D0=D0+ 5
- C=R1
- DAT0=C A FUNCR0 + 15 := R1
- GOSUBL =poscmd
- GONC invcmd "Invalid Cmd"
- La commande a été reconnue et acceptée
- Ce qui suit est un "hack" basé sur le fait qu'il y a moins de 16 commandes dans l'éditeur. Tout tient donc sur un seul quartet.

```

0251 00160
0252 00160 8E0000     GOSUBL =NumD01 place A(A) en FUNCR0
0253 00166 31A3      LCASC " "
0254 0016A 140       DATI=C B
0255 0016D 171       DI=D1+ 2      +D1+ := " "
0256 00170 AEO       A=0 B
0257 00173 A6C       A-A-1 B      A(A) := #FF
0258 00176 149       DATI=A B      +D1 := #FF
0259 00179
0260 00179 8F0000     GOSBVL =AVS2DS
0261 00180
0262 00180
0263 00180
0264 00180 1B0000     D0=(5) =COUR
0265 00187 146       C-DAT0 A      C(A) := courante
0266 0018A 8E0000     GOSUBL =seek D0 := longueur LIF
0267 00190 8E0000     GOSUBL =dsplin
0268 00196 8F0000     GOSBVL =CRLFND
0269 0019D 6000      GOTO nul1900
0270 001A1
0271 001A1 330000     nul1800 LC(4) =id)*(=teEOF)
0272 001A7 7021      GOSUB =dspsmg
0273 001AB
0274 001AB
0275 001AB
0276 001AB
0277 001AB 8F0000     nul1900 GOSBVL =KYDN?
0278 001B2 58F       GONC nul1900
0279 001B5 01        RTN
0280 001B7
0281 001B7
0282 001B7
0283 001B7
0284 001B7
0285 001B7
0286 001B7
0287 001B7
0288 001B7
0289 001B7
0290 001B7
0291 001B7
0292 001B7
0293 001B7
0294 001B7
0295 001B7 8C0000     cmNULL GOLONG =cmNULL Rallonge
0296 001B0
0297 001B0 84A       =BOUCLE ST=0 10
0298 001C0
0299 001C0
0300 001C0

```

- BOUCLE
- But: la boucle principale de l'éditeur
- Entree:
 - - PCHD = pointeur sur le premier caractère de la ligne
 - - LCHD = longueur de cette ligne de commande
- Historique:
 - 88/05/14: PD/JT conception & codage
 - 88/05/15: PD/JT séparation de "vedec" et "prompt"
 - 88/11/06: PD/JT sigillage selon le message "teVCHD"
 - 88/11/13: PD/JT ajout de vieuIn
- Décomposition de la commande

```

0351 00218
0352 00218 80D0     P=C 0 P := numéro de la commande
0353 0021C
0354 0021C 1B0000     D0=(5) =FUNCR0
0355 00223 146       C-DAT0 A
0356 00226 D7        D=C A
0357 00228 164       D0=D0+ 5
0358 0022B 146       C-DAT0 A
0359 0022E 135       DI=C A
0360 00231 164       D0=D0+ 5
0361 00234 146       C-DAT0 A
0362 00237 108       R0=C A
0363 0023A 164       D0=D0+ 5
0364 0023D 146       C-DAT0 A
0365 00240 109       RI=C A
0366 00243
0367 00243
0368 00243
0369 00243
0370 00243 8F0000     GOSBVL =TELJMP switch (PY)
0371 0024A 640       REL(3) cmCOPY C
0372 0024D F40       REL(3) cmDEL D
0373 00250 000       REL(3) cmEXIT E
0374 00253 550       REL(3) cmHELP H
0375 00256 000       REL(3) cmINS I
0376 00259 940       REL(3) cmJOIN J
0377 0025C 000       REL(3) cmLIST L
0378 0025F 730       REL(3) cmMOVE M
0379 00262 000       REL(3) cmPRINT P
0380 00265 000       REL(3) cmEXIT O
0381 00268 C10       REL(3) cmREPL R
0382 0026B 000       REL(3) cmSRCH S
0383 0026E 000       REL(3) cmTEXT T
0384 00271 910       REL(3) cmXCHG X
0385 00274
0386 00274
0387 00274
0388 00274
0389 00274
0390 00274
0391 00274 600       REL(3) invcmd
0392 00277 300       REL(3) invcmd
0393 0027A
0394 0027A 330000     invcmd LC(4) =id)*(=teICMD) Invalid Command
0395 00280 6D20      GOTO =vederr
0396 00284
0397 00284
0398 00284
0399 00284 8C0000     cmREPL GOLONG =cmREPL
0400 0028A 8C0000     cmXCHG GOLONG =cmXCHG

```

- On ne sait jamais. L'information vient ici d'un message, qui peut être traduit. C'est donc potentiellement un risque. Il vaut mieux assurer un comportement stable plutôt que de risquer un "Memory Lost".
- REL(3) invcmd
- REL(3) invcmd
- invcmd LC(4) =id)*(=teICMD) Invalid Command
- GOTO =vederr
- Quelques rallonges...
- cmREPL GOLONG =cmREPL
- cmXCHG GOLONG =cmXCHG

```

0401 00290 SC0000 cmCOPY GOLONG =cmCOPY
0402 00296 SC0000 cmMOVE GOLONG =cmMOVE
0403 0029C SC0000 cmDEL GOLONG =cmDEL
0404 002A2 SC0000 cmJOIN GOLONG =cmJOIN
0405 002A8 SC0000 cmHELP GOLONG =cmHELP
0406 002AE .....
0407 002AE * xederr, dspmsg
0408 002AE *
0409 002AE * But: afficher une erreur et repartir à la boucle (erreur)
0410 002AE * ou revenir à l'appelant (dspmsg)
0411 002AE * Entree:
0412 002AE * - C(3-0) = numero d'erreur
0413 002AE * Sortie:
0414 002AE * - par BOUCLE (erreur)
0415 002AE * - par RTN (dspmsg)
0416 002AE * Abime: A-D, D0, D1, R0
0417 002AE * Appelle: MFWRN
0418 002AE * Niveaux: 2 (dspmsg) ou 3 (erreur)
0419 002AE * Note: Un traitement special est effectue pour eMEM car
0420 002AE * MFWRN ne revient pas si C(3-0) = eMEM. Nous remplacons
0421 002AE * donc eMEM par eMEM qui est notre "Insuffisant Mem" à
0422 002AE * nous memes que d'abord que.
0423 002AE * Historique:
0424 002AE * 88/05/14: PD/JT conception & codage
0425 002AE * 88/07/10: PD/JT separation de dspmsg
0426 002AE * 88/11/13: PD/JT traitement de eMEM
0427 002AE .....
0428 002AE
0429 002AE
0430 002AE 7520 =xederr GOSUB mem?
0431 002AE 2A P= 31010
0432 002AE *
0433 002AE * 1 BEEP
0434 002AE * 0 store ERRN
0435 002AE * 1 display message only (no "WRN Lxxx:")
0436 002AE * 0 DELAY respecte
0437 002AE *
0438 002AE 8F0000 GOSBVL =MFWRN
0439 002AE D0 A
0440 002AE 1F0000 D1=(5) =LCMD
0441 002AE 14E DAT=C A LCMD := 0
0442 002AE 65FE GOTO =BOUCLE
0443 002AE
0444 002AE 7900 =dspmsg GOSUB mem?
0445 002AE 27 P= 31111
0446 002AE *
0447 002AE * 0 no BEEP
0448 002AE * 1 do not store ERRN
0449 002AE * 1 display message only (no "WRN Lxxx:")
0450 002AE * 1 no DELA

```

```

=AVHEHE Extern Ukn - 0148
=AVS2DS Extern Ukn - 0260
=BOUCLE =01BD Rel 0297 - 0213 0442
=BSERR Extern Ukn - 0182
=BUFADR Unkn Ukn 0015 -
=BUFID Unkn Ukn 0014 -
BCL000 =01C0 Rel 0298 - 0322
BCL100 =01D2 Rel 0320 - 0310
BCL200 =01DF Rel 0324 - 0303
=CHKTXT Extern Ukn - 0179
=CHKWRT Extern Ukn - 0170
=COUR Unkn Ukn 0018 - 0195 0239 0249 0264
=CRLFND Extern Ukn - 0268
=DERN Unkn Ukn 0019 - 0237
=EGLXCK Extern Ukn - 0099
=EXPEXC Extern Ukn - 0101
=FCNTMP Unkn Ukn 0009 -
=FILADR Unkn Ukn 0017 - 0174
=FILE1 Extern Ukn - 0139
=FILE2 Extern Ukn - 0166
=FLIMSK =0002 Abs 0053 - 0204 0208
=FLAG1 Unkn Ukn 0022 - 0210
=FLGREG Extern Ukn - 0201
=FUNCR0 Extern Ukn - 0331 0354
FILEnd =02F6 Rel 0405 -
=IVAFER Extern Ukn - 0079
=KVD07 Extern Ukn - 0277
=LCMD Unkn Ukn 0021 - 0158 0440
=LDOSPC Extern Ukn - 0045
=MFWRN Extern Ukn - 0438 0452
=MODE71 Unkn Ukn 0007 - 0133
=NCMDED =0000 Abs 0054 -
=Num201 Extern Ukn - 0252
=ORCOLL Extern Ukn - 0246
=PCMD Unkn Ukn 0020 - 0153
=FOPI5 Extern Ukn - 0146
=QUERY Unkn Ukn 0008 -
=ROONLY Unkn Ukn 0023 - 0171
=RNDAXH Extern Ukn - 0071
=SCRTOH Extern Ukn - 0039 0047
=STSAVE Extern Ukn - 0046
=TBLJMP Extern Ukn - 0370
=TEDITd Extern Ukn - 0118
=TEDITe =0041 Rel 0120 -
=TEDITp Extern Ukn - 0119
=TMP1 Unkn Ukn 0047 -
=TMP3 Unkn Ukn 0046 -
=TMP5 Unkn Ukn 0045 -
=TRFMFB Extern Ukn - 0067 0008 0009 0014 0015 0017 0018 0019 0020
- 0021 0022 0023 0029 0126

```

```

0451 002D1 *
0452 002D1 8D0000 =GOLONG =MFWRN
0453 002D8 .....
0454 002D8 * mem?
0455 002D8 *
0456 002D8 * But: MFWRN ne revenant pas s'il y a une erreur eMEM, il
0457 002D8 * faut tester spécialement ce cas et remplacer eMEM par
0458 002D8 * notre message d'erreur
0459 002D8 * Entree:
0460 002D8 * - C(4-0) = message d'erreur
0461 002D8 * Sortie:
0462 002D8 * - C(4-0) = message d'erreur modifié si eMEM
0463 002D8 * - P = 0
0464 002D8 * Abime: A(A), C(A), P
0465 002D8 * Appelle:
0466 002D8 * Niveaux: 0
0467 002D8 * Historique:
0468 002D8 * 88/11/13: PD/JT conception & codage
0469 002D8 .....
0470 002D8
0471 002D8
0472 002D8
0473 002D8 00 mem? A0 A
0474 002D8 23 P= 1
0475 002D8 A9A A=C WP
0476 002D8 20 P= 0
0477 002D8 02 C=0 A
0478 002D8 3100 LC(2) =eMEM
0479 002D8 0E AC=C A
0480 002D8 2A5 AC=C A
0481 002D8 00 RTNYES
0482 002D8 30000 LC(4) =id / (eMEM), notre erreur à nous
0483 002D8 01 RTN
0484 002D8
0485 002D8 END

```

```

=WIPOUT Extern Ukn - 0128
=XEDITd Extern Ukn - 0123
=XEDITe =0052 Rel 0125 -
=XEDITp Extern Ukn - 0124
=XEDTMP Unkn Ukn 0029 -
XED000 =0055 Rel 0126 - 0121
XED010 =0070 Rel 0135 - 0131
XED050 =00BC Rel 0163 - 0145
XED080 =00F5 Rel 0134 - 0180
bserr =00EE Rel 0182 -
cmCOPY Extern Ukn - 0401
cmDEL Extern Ukn - 0403
cmEXIT Extern Ukn - 0373 0380
cmHELP Extern Ukn - 0405
cmINS Extern Ukn - 0375
cmJOIN Extern Ukn - 0404
cmLIST Extern Ukn - 0377
cmMOVE Extern Ukn - 0402
cmNULL Extern Ukn - 0295
cmPNT Extern Ukn - 0379
cmREPL Extern Ukn - 0399
cmSRCH Extern Ukn - 0382
cmTEXT Extern Ukn - 0383
cmXCHG Extern Ukn - 0400
ccoeffa Unkn Ukn 0029 - 0040
ccoeffb Unkn Ukn 0040 -
cmCOPY 00290 Rel 0401 - 0371
cmDEL 0029C Rel 0403 - 0372
cmHELP 002A8 Rel 0405 - 0374
cmJOIN 002A2 Rel 0404 - 0376
cmMOVE 00296 Rel 0402 - 0378
cmNULL =01B7 Rel 0295 - 0326
cmREPL =0024 Rel 0399 - 0381
cmXCHG =002A Rel 0400 - 0384
dsplin Extern Ukn - 0267
dspmsg =00CB Rel 0444 - 0272
eMEM Extern Ukn - 0478
eMEM Extern Ukn - 0482
id Extern Ukn - 0271 0394 0482
id Extern Ukn - 0191
inibuf Extern Ukn -
invcmd =027A Rel 0394 - 0345 0391 0332
ivarg =0019 Rel 0079 - 0072 0074
mem? =0208 Rel 0473 - 0430 0444
nu1300 =01A1 Rel 0271 - 0242
nu1300 =01AB Rel 0277 - 0269 0278
ntexp =00C0 Rel 0097 - 0144
pocmd Extern Ukn - 0344
prompt Extern Ukn - 0320
pperg =0000 Rel 0071 -
scef Extern Ukn - 0266

```