```
00: MOVE.L  TPILE, SP
    BSR     EXCHTRAP      ⑥ 39.7
    OR.B    #8, $484.W
[ou: MOVE.B  #14, $484.W ]
    LEA     X68000, A0
    MOVEQ   #1, D0
    Processor 1
    MOVE.B  -2(A0, D0*2), D0
    Process FLAG30
    MOVE.B  D0, (A0)
```

modif "68000"/"68030" suivant le processeur

{"0" "3"}

"Exception 680" } erreur n°57
X68000:"30", 57

~~BSR HDEM~~

```
[hd280 :] bsr hidem
          move #4,-(sp)          xbios(4)
          trap #14
          move d0,resol          resol = { 0 basse
                                           1 moy
                                           2 haute   <- résolution du bureau
          move d0,resolb
          move.b d0,resolc       <- resolution0
          lea colres+6,a3        } lit et met les
                                   couleurs
          bsr setcol4            <- { MOVE D0,D5
                                      MOVEM d3/d4/d5,(a3)  (x)
          move d0,-(a3)          }
          subq #2,a3             } valeurs initiales des couleurs
          move d3,-(a3)          }
          subq #2,a3             }
          movem d3/d4,-(a3)      }

          bsr setbreak           localise getshift_ad pour break

          dw $a000               } init de la ligne a
          move.l 8(a0),intina      (inutil intina=intin)
                  TPILE
          move.l (a6),sp

          bsr ylimtm  ) (x)

          moveq #78,d3           }
          clr aintin             } aes 78,0   souris = flèche
          bsr aesc               }
          bsr xdecn              déconnecte les lignes (met T1173 = 0..)
          move.l #RER,$BC.W      [trap #15 = RER]                - XPSDIV-1
          move.l
          lea $8C.W,A0
          move.l #trap3,(A0)+    trap#3 = verna    } perd 24 cycles /BSR.K   5580  KA77
          move.l #trap4,(a0)+    trap#4 = windes   }                         19761 MG62
          move.l #trap5,(a0)+    trap#5 = wadr     }                         ~ KT69
          move.l #trap6,(a0)+    = cuexpr

          lea td000,a0           } 4000 blancs ds td000
          move #4000,d0          <- CLR TDEL-TD000(a0)  vide buffer d'affiche (x)
          bsr dambl              }
          lea
          CLR.B FILETYP,a0       inhibe verify
          clr.b (a0)+
          clr.b (a0)
          MOVE #-1,BINSTR        inhibe { B_END
                                        { B_TRACE
```

```
BSR   XCLRT1     { met écran éditeur et vide écran
                 { vide écran
JSR   XESAV      sauvegard écran prg vide
BSR   SINIT1     vide la source
BSR   EXCEP      exceptions
CLR   TPRINT
MOVE.L TPILE,SP
BSR   INOMI      préparation de la table des mots clefs
;BSR  TSTA       si on veut tester le nb de clefs par somme type
BSR   INITR
BSR   XCUROFF
CLR   THLCLE     mot clef vide
BSR   RESTOREA   lit secteur 1 piste 0 (après protection)
BSR   SAKKI      code claviers
```

```
MOVEQ #NBCANAUX ,d0
LEA   CANAUX,A0
GA24: CLR (A0)+
      DBRA D0,GA24
```

} les canaux sont fermés

```
PEA   DTA
MOVE  #$1A,-(SP)
TRAP  #1
ADDQ  #6,SP
```

} fixe l'adresse de stockage des données fichier ⊗
(pour exist sinon utilise basepage+128)
DTA : tampon de 44 octets

↓
HOT

vide la table (1 seul appel)

XINIT: LEA GEVNT, A0
MOVEQ #GARBRN-GEVNT/2, d0
GAG85: CLR (A0)+
DBRA D0, GAG85

BSR XREPDEF ⑥1c met le répertoire par défaut

```
HD288: BSR   VINITO              ) mode éditeur
HD29A: BSR   X9
HOTB:  BSR   VDKEY               retour éditeur sans aller à la ligne
                    ← BSR SAVER   mode "edit source" enlevé
       BSR   DAMPR1              ) remet taura
       BSR   PLMOD1                          b
                                             y
       BSR   DAMED               damier éditeur

       MOVE  #$4E71, escape      "NOP"
       MOVE  #$4E71, pack        "
       BSR   MET70               } interrupt. 200Hz
       AND   #$2300, SR
                    BSR VEREXCEP   vérifie que les exceptions n'ont pas été falsifiées
       ~~BSR   HIDEM~~
       BSR   ~~S3E0AC~~ closerr     ferme fichier
       CLR.B FVERPRN             ) vérif. d'imprimante : une fois pour tous seulement
       CLR.B TESCV
       MOVE  RESOL, RESOL1
       CLR.B NIVEAU7             { 0 inter permises       ] pour pack
                                 { -1 " interdites et non demandées
                                 { 1  "    "     et demandées
       CLR.L TFONS               pas d'erreur dans decode

       BSR   YBREAK0             "break Ø"

       ~~BSR   AESVDIPB~~          met blocs AES et VDI    ⊗
       ~~MOVE  UAES0, UAES~~       ⊗
```

```
⊗ lea   tcursbm,a0
  move  (a0)+,d0
  move  #4,(a0)+
  move  d0,(a0)+

        LEA    TDNM,A0      "source"
        LEA    TDPR,A1
        MOVE   (A1)+,D0
        CMP.B  (A1),D0
        BEQ    HD29B
        LEA    TD...,A0     "edit"
HD29B:  BSR    XPTET
        MOVE   #$4E71,ESCAPE    enable escape
                NOP

ZB9BD:  MOVE.L TPILE,SP
        BSR    XCUROFF

HD289:  BSR    VINITO       remet mode éditeur et variable à 0

                            boucle éditeur

HD29A:  BSR    VDKEY        vide clavier
        BSR    PLMOD1

HD29C:  CLR    YTRACE
        CLR    TFRDG        0 in/edit, 1 run, -1 debug
        CLR    TFRUN        pour erreurs {0 mode source/éditeur, ≠0 calcul/run/immédiat

        MOVE.L TDHAUT,D0
        ADDQ   #8,D0
        CMP.L  TVARB,D0
        BCC    HD2..
        MOVE.L TVARF,D0
        ADDQ   #8,D0
        CMP.L  TCOND,D0
        BCC    HD2..
        MOVE.L TVALA6,A0
        ADD    #200,A0
        CMP.L  SP,A0
        BCS    HD29C
HD2..:  lea    TDCLR,A0
        BSR    XN15
        BSR    X9
        BSR    INSTR
HD2..:  BSR    INSTR
HD2..:  MOVE.L TVARB,D0
        CMP.L  TVARF,D0
        BEQ    HD29C

⊗       BSR    X9

        MOVE.L TPILE,SP
GEB45:  MOVE   #"*"*256,TBUF    *0 sur TBUF
        BSR    INSER            mode insertion
        BNE    HD290            → non
        BSR    PLGNT2           met [ > ]0 sur TBUF
HD290:  BSR    GET29            {A5 = deb de ligne
                                {A4 = pos curseur
                                 A3 = pos fin
        CMP.B  #"~",X(A5)
        BNE    HD290A
        BSR    X6
        BRA    HD290
HD290A: LEA    TBUF,A0
        CMP.L  A5,A3
        BNE    HD291            → ligne non vide
HD290B: BSR    XN15
        BRA    HD292

        MOVE.B TBASE+3,TBASE+1   remet base (si erreur) de MUSIC
        MPROT14
        BSR    XBEND    { exécute B_END
                        { ... inhibé
                        p41cc
```

```
JTRACE:
BINSTR: D.B 0        bit7  1 inhibé
BEND  : D.B 0              0 inactif
```

exécute B_END si actif, puis l'inhibe

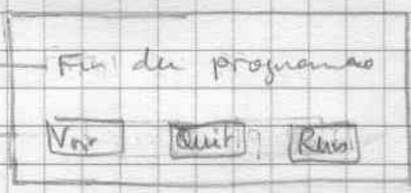
```
~~XBEND~~: LEA  BINSTR,~~A0~~
           TAS  (A0)+        inhibe (définitif B_INSTR)
~~XBEND~~: TAS  (A0)
           BMI  GEB45        → inactif
           LEA  GEB44,A5        "B_END"
           BSR  WSPBAS
GEB45:     ~~RTS~~
```

```
GEB95: IF * FLAGPY                      si run only
       BSR   CURSACTG
       BSET  #0,PASS1ST                 ] teste le 1er passage
X      BNE   PD91                       → non
                                        ↓oui : créer la source

np
GAL31  MOVE.L  TDHAUT,A1
       MOVE.L  TPILE,A0
       MOVE.L  (A0)+,D0
       BSR     RYCODE                   traduit
       ST      TNEWLB
       BSR     XCONVR                   remet les pointeurs
       MOVE.L  TPILE,SP               } remet limite max
       BSR     YLIMTM                 }
X      BRA     PD94

PD91 : LEA     PD92,A0
       LEA     PD93,A2
       MOVEQ   #3,D1      ——→ /!\
       BSR     GEV81      (1) 1376
```

PD92 ←
PD93 ←




```
       BNE     PD95                     kayget
voir   BSR     VDKEY                    vide buffer clavier
       BSR     XKEY
       BRA     PD91

PD95:  SUBQ    #2,D0
       BEQ     WQUIT                    → quitter (après WDSEKA)
PD94:  BRA     WRUNR   (barré)
       LEA     MD94 ,A6
       BRA     WRUNR
```