



Récupérez vos programmes ou fichiers protégés

Vous avez écrit un programme, puis l'avez enregistré sur une cassette ou une disquette en le protégeant. Après quoi, vous l'avez effacé de la mémoire de votre AMSTRAD. Malheureusement, dans votre précipitation de mettre à l'abri votre œuvre, vous avez oublié de conserver une copie non protégée de votre travail.

Le programme proposé ici vous permettra de recopier vos fichiers de cassettes, ou disquette à cassette, ou disquette en supprimant le flag de protection mis en place par le soft du CPC. Il vous est fourni sous la forme d'un listing Assembleur et d'un listing Basic. Le listing Assembleur permettra à ceux qui le désirent d'améliorer ce programme. (Par exemple en y incluant un traitement des erreurs). Le listing Basic autorise tous les possesseurs de CPC à utiliser le programme et permet de choisir le type de support des fichiers entrée et sortie.

Programme Assembleur

Le programme Assembleur qui vous est présenté aurait pu être écrit par tout possesseur de la précieuse brochure *AMSTRAD COMPLETE FIRMWARE SPECIFICATIONS*.

Il permet, pour tout fichier enregistré en "FILE FORMAT" (c'est le cas des fichiers ou programme créés par la commande SAVE) de charger ce fichier en RAM, puis de la ré-enregistrer sur un support magnétique. Le fichier obtenu en sortie aura des spécifications strictement identiques au fichier fourni en entrée, mais il ne sera plus protégé. Cela signifie que, à l'exception du bit de protection, les différentes parties, utiles du header du fichier entrée (adresse des données, adresse du point d'entrée, longueur du fichier, zone utilisateur) auront les mêmes valeurs dans le header du fichier sortie.

La routine AFFER1 est destinée à permettre le traitement des erreurs. Elle n'a pas été utilisée dans ce programme et se compose uniquement de deux instructions, mais vous pouvez éventuellement y inclure le traitement d'erreurs que vous désirez. Attention toutefois, dans ce cas, à modifier l'adresse de chargement du

programme. Celle qui est donnée (&A4B5) a été calculée pour permettre aux possesseurs d'une unité de disquettes d'utiliser ce logiciel.

Programme Basic

Le programme BASIC permettra à ceux qui ne disposent pas d'un Assembleur-Éditeur de mettre en œuvre ce programme. Il permet en outre à ceux qui disposent d'une unité de disquettes de choisir le support du fichier entrée et celui du fichier sortie. Pour pouvoir utiliser ce programme, les utilisateurs AMSTRAD qui ne possèdent pas d'unité de disquettes ne devront pas saisir les lignes 50, 60, 70, 100, 110, 130 et 140.

Données et routines utilisées en Assembleur

TYP-IN type de support du fichier entrée. sera à 0 pour une cassette, à 1 pour une disquette.

TY-OU type de support du fichier en sortie. sera à 0 pour une cassette, à 1 pour une disquette. Ces deux zones sont renseignées par le programme Basic. Elles permettent de déterminer la longueur maximum du nom de fichier.

— NOM-IN nom du fichier en entrée.
— NOM-OU nom du fichier en sortie.
— LNG-IN longueur du nom du fichier entrée.

— LNG-OU longueur du nom du fichier sortie.

— TYP-PG type du fichier entrée.
— ADR-DE adresse de chargement du fichier.

— ADR-EN adresse du point d'entrée.
— LNG-PG longueur totale du fichier.
— USR-ZO zone utilisateur du header.

Les zones TYP-PG, ADR-DE, ADR-EN, LNG-PG, USR-ZO, sont renseignées par le header du fichier entrée.

Les zones NOM-IN, NOM-OU, LNG-IN, LNG-OU sont renseignées par la demande de nom de fichier à l'écran.

Principales routines utilisées

Ces routines sont appelées par un CALL

à leur adresse. Les adresses d'appel sont les suivantes:

BC77 OPEN d'un fichier input. Cette routine retourne:

+ Dans le registre A le type du fichier,
+ Dans le double registre BC la longueur totale du fichier,

+ Dans le double registre DE l'adresse de chargement des données,

+ Dans le double registre HL l'adresse d'un buffer contenant le header du fichier.

La zone utilisateur est contenue dans les octets 28 à 63 de ce buffer.

+ BC83 READ d'un fichier input. Cette routine retourne dans le double registre HL l'adresse du point d'entrée.

BC8C OPEN d'un fichier sortie. Cette routine retourne dans le double registre HL l'adresse d'un buffer contenant le header. La zone utilisateur obtenue lors de l'OPEN du fichier entrée sera reconduite dans ce buffer.

BC98 WRITE d'un fichier out put.
BC0E Mise de l'écran dans le mode fourni dans le registre A.

BC32 affectation à l'INK dont le numéro est donné dans le registre A des couleurs dont les numéros sont donnés dans les registres B et C.

BB06 Lecture d'un caractère frappé au clavier. Le caractère est donné dans le registre A.

BB5A Affichage à l'écran du caractère dont la valeur est dans le registre A.

BB78 Retourne les positions horizontales du curseur dans les registres H et L.

BB75 positionne le curseur aux positions données dans les registres H (Numéro de colonne) et L (numéro de ligne).

Utilisation du programme

En ce qui concerne le programme Basic, vous n'avez qu'à taper K pour une cassette ou D pour une disquette lors de la demande dy type de support. Lors de la copie de fichier, le programme vous demande le nom du fichier en entrée puis celui du fichier en sortie. Le nom que vous tapez alors doit obéir aux règles suivantes:

VIDEO CLUB BOBIGNY 2

Centre commercial BOBIGNY 2, 93000 BOBIGNY

du lundi au samedi de 9 h 30 à 20 h

Tél. 831.69.33

Les matériels et logiciels importés d'Angleterre sont disponibles à Rallye Brest...

Nom du fichier en entrée

Il doit contenir au maximum 16 caractères pour une cassette et 12 (8 pour le nom et éventuellement 4 pour le type sous la forme. XXX). Le programme considère le nom, comme fourni lorsque vous avez atteint ce maximum ou appuyé sur ENTER. Ce nom est obligatoirement pour un fichier sur disquette, mais facultatif pour un fichier sur cassette. Si vous ne fournissez pas de nom, appuyez sur ENTER. Le programme chargera alors le premier fichier trouvé sur la cassette. Si il s'agit d'une dis-

quette, vous aurez le message "Bad command" et le programme vous redemandera le nom du fichier.

Nom du fichier en sortie

Il doit obéir aux mêmes règles que le nom du fichier en entrée. Appuyez sur ENTER directement si vous voulez reconstruire le nom du fichier d'entrée sur le fichier de sortie. Toutefois, si vous n'avez pas fourni de nom de fichier d'entrée, le fichier sera écrit sans nom sur la cassette (Unnamed File), et sur dis-

quette vous aurez le message d'erreur "Bad Command" puis à nouveau la demande de nom du fichier en sortie.

Demande d'un autre fichier

Après le traitement complet d'un fichier, le programme vous demandera: AUTRE FICHIER (O/N)?

Si vous avez à traiter un autre fichier obéissant aux mêmes conditions de support, tapez O. Sinon, tapez N. Le programme effectue alors un RESET général (par RST 0).

R.P. Spiegel

```

10 REM *****
20 REM *      Auteur : R.P-SPIEGEL - aout 1985      *
30 REM *****
40 MEMORY 42164: CLEAR:CLS
50 LOCATE 1,1:PRINT "SUPPORT DU FICHIER ENTREE (K ou D) ? "
60 GOSUB 130:LOCATE 38,1:PRINT e$:e1$=e$
70 LOCATE 1,3:PRINT "SUPPORT DU FICHIER SORTIE (K ou D) ? "
80 GOSUB 130:LOCATE 38,3:PRINT e$
90 FOR i=42165 TO 42619:READ a%:POKE i,a%:NEXT
100 IF e1$="K" THEN !TAPE.IN:POKE 42168,0:GOTO 110 ELSE POKE 42168,1
110 IF e$="K" THEN !TAPE.OUT:POKE 42169,1:GOTO 120 ELSE POKE 42169,1
120 CALL 42165
130 e$="":WHILE e$="":e$=INKEY$:WEND:e$=UPPER$(e$):IF e$(">"K" AND e$(">"D" THEN G
OTO 130
140 RETURN
150 DATA &c3,&57,&a5,&00,&00,&4e,&4f,&4d,&20,&44,&55,&20,&46,&49,&43,&48,&49,&45
,&52,&20,&45,&4e,&20,&45,&4e,&54,&52,&45,&45,&20,&3f,&20
160 DATA &4e,&4f,&4d,&20,&44,&55,&20,&46,&49,&43,&48,&49,&45,&52,&20,&45,&4e,&20
,&53,&4f,&52,&54,&49,&45,&20,&3f,&20
170 DATA &41,&55,&54,&52,&45,&20,&46,&49,&43,&48,&49,&45,&52,&20,&28,&4f,&2f,&4e
,&29,&20,&3f,&20
180 DATA &00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00
,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00
190 DATA &00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00
,&00,&00,&00,&00,&00,&00,&00
200 DATA &00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00
,&00,&00
210 DATA &3e,&02,&cd,&0e,&bc,&3e,&00,&06,&00,&0e,&00,&cd,&32,&bc,&3e,&01,&06,&12
,&48,&cd,&32,&bc
220 DATA &cd,&54,&bb,&cd,&72,&a6,&21,&ba,&a4,&cd,&48,&a6,&06,&11,&36,&00,&23,&10,
&fb,&21,&06,&a5,&0e,&00,&3a,&b8,&a4
230 DATA &cd,&55,&a6,&21,&16,&a5,&71,&41,&21,&06,&a5,&11,&72,&01,&cd,&77,&bc,&da
,&a2,&a5,&cd,&46,&a6,&c3,&70,&a5
240 DATA &32,&28,&a5,&ed,&53,&29,&a5,&ed,&43,&2d,&a5,&11,&2f,&a5,&01,&18,&00,&09
,&01,&28,&00,&ed,&b0,&2a,&29,&a5,&cd,&83,&bc
250 DATA &22,&2b,&a5,&cd,&7a,&bc,&cd,&72,&a6,&21,&d5,&a4,&cd,&48,&a6,&11,&17,&a5
,&01,&11,&00,&ed,&b0,&21,&17,&a5,&0e,&00
260 DATA &3a,&b9,&a4,&cd,&55,&a6,&79,&fe,&00,&ca,&eb,&a5,&21,&27,&a5,&71,&21,&27
,&a5,&46,&21,&17,&a5,&11,&72,&01,&cd,&8c,&bc
270 DATA &da,&01,&a6,&cd,&46,&a6,&c3,&c5,&a5,&01,&18,&00,&09,&eb,&21,&2f,&a5,&01
,&28,&00,&ed,&b0,&ed,&4b,&2b,&a5,&2a,&29,&a5
280 DATA &ed,&5b,&2d,&a5,&3a,&28,&a5,&cb,&87,&cd,&98,&bc,&cd,&8f,&bc,&cd,&72,&a6
,&21,&f0,&a4,&06,&16,&7e,&cd,&5a,&bb,&23
290 DATA &10,&f9,&cd,&06,&bb,&cd,&5a,&bb,&f6,&20,&fe,&6f,&ca,&70,&a5,&fe,&6e,&c2
,&24,&a6,&c7,&37,&c9,&06,&1b,&7e,&cd,&5a,&bb
300 DATA &23,&10,&f9,&21,&06,&a5,&c9,&fe,&00,&ca,&5f,&a6,&06,&0c,&c3,&61,&a6,&06
,&10,&cd,&06,&bb,&fe,&0d,&ca,&71,&a6,&0c
310 DATA &cd,&5a,&bb,&77,&23,&10,&f0,&c9,&cd,&78,&bb,&26,&01,&2c,&cd,&75,&bb,&c9

```



Pass 1 errors: 00

```

A4B5      10      org  #a4b5
A4B5      20      ent  $
A4B5      C357A5  30      jp   debut
A4B8      00      40  typ_in: defb 0
A4B9      00      50  typ_ou: defb 0
A4BA      4E4F4D20 60  dem_in: defm "NOM DU FICHER EN ENTREE ? "
A4D5      4E4F4D20 70  dem_ou: defm "NOM DU FICHER EN SORTIE ? "
A4F0      41555452 80  dem_au: defm "AUTRE FICHER (O/N) ? "
A506      90      nom_in: defs 16      ; nom du fichier en entre
e
A516      00      100  lng_in: defb 0      ; longueur nom du fichier
entree
A517      110     nom_ou: defs 16      ; nom du fichier sortie
A527      00      120  lng_ou: defb 0      ; longueur nom du fichier
sortie
A528      00      130  typ_P9: defb 0      ; type du fichier entree
A529      0000    140  adr_de: defw 00      ; adresse debut fichier e
n RAM
A52B      0000    150  adr_en: defw 00      ; adresse Point d'entree
A52D      0000    160  lng_P9: defw 00      ; longueur totale du fich
ier
A52F      170     usr_zo: defs 40      ; zone utilisateur du lab
el
A557      3E02    180  debut: ld   a,2
A559      CD0EBC  190      call #bc0e      ; mise de l'ecran en mode
80 colonnes
A55C      3E00    200      ld   a,0
A55E      0600    210      ld   b,0
A560      0E00    220      ld   c,0
A562      CD32BC  230      call #bc32      ; ink 0 noire
A565      3E01    240      ld   a,1
A567      0612    250      ld   b,#12
A569      48      260      ld   c,b
A56A      CD32BC  270      call #bc32      ; ink 1 verte
A56D      CD54BB  280      call #bb54
A570      CD72A6  290  deminP: call Plccur
A573      21BAA4  300      ld   hl,dem_in
A576      CD48A6  310      call demnom
A579      0611    320      ld   b,#11
A57B      3600    330  razinP: ld   (hl),#0      ; mise a zero du nom du f
ichier
A57D      23      340      inc  hl
A57E      10FB    350      djnz razinP
A580      2106A5  360      ld   hl,nom_in
A583      0E00    370      ld   c,#0
A585      3AB8A4  380      ld   a,(typ_in)
A588      CD55A6  390      call sainom
A58B      2116A5  400  ouvinP: ld   hl,lng_in
A58E      71      410      ld   (hl),c
A58F      41      420      ld   b,c
A590      2106A5  430      ld   hl,nom_in
A593      117201  440      ld   de,#172
A596      CD77BC  450      call #bc77      ; OPEN fichier entree

```

TRUCS ET BIDOUILLES

```

A599 DAA2A5      460      JP      c,chrinP
A59C CD46A6      470      call   affer1      ; si erreur a l'OPEN
A59F C370A5      480      JP      deminP
A5A2 3228A5      490      chrinP: ld      (typ_P9),a
A5A5 ED5329A5    500      ld      (adr_de),de
A5A9 ED432DA5    510      ld      (ln9_P9),bc
A5AD 112FA5      520      ld      de,usr_zo
A5B0 011800      530      ld      bc,#18
A5B3 09          540      add    hl,bc
A5B4 012800      550      ld      bc,#28
A5B7 EDB0        560      ldir
A5B9 2A29A5      570      ld      hl,(adr_de)
A5BC CD83BC      580      call   #bc83      ; lecture fichier
A5BF 222BA5      590      ld      (adr_en),hl
A5C2 CD7ABC      600      call   #bc7a      ; CLOSE fichier entree
A5C5 CD72A6      610      demout: call   Plccur
A5C8 21D5A4      620      ld      hl,dem_ou
A5CB CD48A6      630      call   demnom
A5CE 1117A5      640      ld      de,nom_ou
A5D1 011100      650      ld      bc,#11
A5D4 EDB0        660      ldir
A5D6 2117A5      670      ld      hl,nom_ou
A5D9 AE00        680      ld      c,#0
A5DB 3AB9A4      690      ld      a,(typ_ou)
A5DE CD55A6      700      call   sainom
A5E1 79          710      ouvout: ld      a,c
A5E2 FE00        720      cp      0
A5E4 CAEBA5      730      JP      z,memnom
A5E7 2127A5      740      ld      hl,ln9_ou
A5EA 71          750      ld      (hl),c
A5EB 2127A5      760      memnom: ld      hl,ln9_ou
A5EE 46          770      ld      b,(hl)
A5EF 2117A5      780      ld      hl,nom_ou
A5F2 117201      790      ld      de,#172
A5F5 CD8CBC      800      call   #bc8c      ; OPEN fichier sortie
A5F8 DA01A6      810      JP      c,urtout
A5FB CD46A6      820      call   affer1      ; si erreur a l'OPEN
A5FE C3C5A5      830      JP      demout
A601 011800      840      urtout: ld      bc,#18
A604 09          850      add    hl,bc
A605 EB          860      ex     de,hl
A606 212FA5      870      ld      hl,usr_zo
A609 012800      880      ld      bc,#28
A60C EDB0        890      ldir
A60E ED4B2BA5    900      ld      bc,(adr_en)
A612 2A29A5      910      ld      hl,(adr_de)
A615 ED5B2DA5    920      ld      de,(ln9_P9)
A619 3A28A5      930      ld      a,(typ_P9) ; type fichier sortie
A61C CB97        940      res    0,a        ; deProtection fichier
A61E CD98BC      950      call   #bc98      ; ecriture fichier
A621 CD8FBC      960      call   #bc9f      ; CLOSE fichier sortie
A624 CD72A6      970      askaut: call   Plccur      ; demande si autre fichier
A627 21F0A4      980      ld      hl,dem_au
A62A 0616        990      ld      b,#16
A62C 7E        1000     affaut: ld      a,(hl)
A62D CD5ABB     1010     call   #bb5a
A630 23        1020     inc    hl

```



```

A631 10F9 1030      dinz affaut
A633 CD06BB 1040      call #bb06
A636 CD5ABB 1050      call #bb5a
A639 F620 1060      or #20
A63B FE6F 1070      cp #6f
A63D CA70A5 1080      jp z.deminP
A640 FE6E 1090      cp #6e
A642 C224A6 1100      jp nz.askaut
A645 C7 1110      ret 0
A646 37 1120      affer1: scf
A647 C9 1130      ret
A648 061B 1140      demnom: ld b,#1b ; demande nom du fichier
A64A 7E 1150      asknom: ld a,(hl)
A64B CD5ABB 1160      call #bb5a
A64E 23 1170      inc hl
A64F 10F9 1180      dinz asknom
A651 2106A5 1190      ld hl,nom_lin
A654 C9 1200      ret
A655 FE00 1210      sainom: cp 0 ; saisie nom du fichier
A657 CA5FA6 1220      jp z.casfic
A65A 060C 1230      ld b,#c ; longueur maxi 12 si fic
hier diskette
A65C C361A6 1240      jp linnom
A65F 0610 1250      casfic: ld b,#10 ; longueur maxi 16 si fic
hier cassette
A661 CD06BB 1260      linnom: call #bb06
A664 FE00 1270      cp 13
A666 CA71A6 1280      jp z.retsai
A669 0C 1290      inc c
A66A CD5ABB 1300      call #bb5a
A66D 77 1310      ld (hl),a
A66E 23 1320      inc hl
A66F 10F0 1330      dinz linnom
A671 C9 1340      retsai: ret
A672 CD78BB 1350      Plccur: call #bb78 ; remise du curseur col 1
      ligne suivante
A675 2601 1360      ld h,l
A677 2C 1370      inc l
A678 CD75BB 1380      call #bb75
A67B C9 1390      ret

```

Pass 2 errors: 00

```

adr_de A529  adr_en A52B  affaut A62C
affer1 A646  askaut A624  asknom A64A
casfic A65F  chrinp A5A2  debut A557
dem_au A4F0  dem_lin A4BA  dem_ou A4D5
deminP A570  demnom A648  demout A5C5
linnom A661  lng_lin A516  lng_ou A527
lng_P9 A52D  memnom A5EE  nom_lin A506
nom_ou A517  ouvinP A58B  ouvout A5E1
Plccur A672  razinP A57B  retsai A671
sainom A655  typ_lin A4B9  typ_ou A4B9
typ_P9 A528  usr_zo A52F  wrtout A601

```

Table used: 441 from 500
Executes: 42165