Zero Free

Pack every sector on your disks for efficient storage.

by Mike Stortz

Over the years, I've collected quite a few binary load programs, distributed over thirty-two disk sides. I've always tried to fit these programs together on a disk as economically as possible, spending hours over a hot calculator, but there were usually ten or twenty free sectors left over.

I recently obtained a "boot menu," a program that eliminated the need to have DOS and an AUTORUN.SYS file on each disk. I had to redistribute my programs to take advantage of this extra space anyway, so I decided that a utility program to replace my calculator—and callused digits—was in order. **Zero Free** was the result.

Since some amount of speed was obviously going to be needed, I naturally chose to write in Action! I take second place to few people in my admiration of this excellent language. Machine language is fine where speed or small size are necessities rather than conveniences, but it's a terror to debug (at least, it's a terror for me to debug). I knew that I wanted to combine file lengths to add up to 707 sectors, but I had no idea how to go about it. I wrote several "intelligent" (and unsuccessful) algorithms before deciding to do the job with brute computing power.

Zero Free will read in the directories of your disks (ignoring any .SYS extenders) and fiercely recombine them randomly, until it meets with a favorable arrangement. You may be suprised at how few programs are required for **Zero Free** to come up with a completely full disk. It then prints out the appropriate filenames to a disk file, the screen or your printer.

Using *Zero Free*.

Insert your Action! cartridge, and type and save Listing 1. Please use **D:Check in Action!** from issue 44 to check your typing.

When run, **Zero Free** will provide you with initial instructions. Give the number of free sectors you have per disk, the maximum number of files you wish to have on a disk side (I did this because my labeling program will only fit seven filespecs on a label), and D, S or P, depending on whether you wish the program output to go to a file named D:PRINTOUT, the screen or your printer, respectively.

Insert each disk that has files you want to pack and press the SPACE BAR. The directory of each disk will be displayed. If a program on the current disk has the same name as a file previously entered, a number sign (#) appears next to it. If these two files are also of equal length, an equal sign (=) appears. You'll probably want to eliminate duplicate files. You can do this by pressing the letter next to the unwanted program. You can also add an extra file, by pressing the plus sign (+) and giving the filename and its length.

When you've finished entering files, press ESC to quit data entry and begin calculation. **Zero Free** will produce a list of files that will fill as much space as possible on a disk side. If you've selected output for D:PRINTOUT, you will be prompted for a disk to write it on.

When zero waste is no longer possible, **Zero Free** will go for the minimum waste it can find. This process continues until all files have been assigned, or till the user has pressed a key and aborted the program.

Using **Zero Free** and a boot menu program like NoDOS, QuikLoad, or BOOT, I reduced my library from thirty-two to twenty disk sides, a savings of six disks. Now, I can put write-protect tabs and neat labels on all those disks, secure in the knowledge that I have—**Zero Free** sectors.

*Mike Stortz is the P.D. Librarian for G.R.A.S.P., the Richmond, Virginia Atari users' group. Seemingly unable to find employment in the programming field, he's working on about thirty projects at once, including a graphic arcade/adventure game that will make **Ultima III** look like "Hunt the Wumpus."*

Listing 1.
Action! listing.

```
;       CHECKSUM DATA
;[9F CA 03 3A 1A 80 5A 02
; 43 F0 3F 9D 21 E0 1F F6
; 95 99 75 78 70 06 4C 37
; 46 73 A7 3D 60 39 F1 BC
; 99 AF 06 8C 79 5B B9 7D
; D1 68 EC 7E EC E7 4C 9F ]

BYTE btemp,spaces,len,checkflag,
     maxfiles,devc,num,quit,
     lmargin=82,shflok=702,ch=764,
     atract=77,crsinh=752,errno=73,
     brkkey=17

CARD idx,which,ctemp,sum,max,spare,
     free,a,b,loss,leastloss,waste,
     addlen,TmpErr,dl=560,sc=88

INT  ii

BYTE ARRAY names(6000),name(20),
     extender(5),hold(324),
     string1(14),string2(14)

CARD ARRAY length(500),hlen(27),
     programs(500),pr(500)


CARD FUNC Min(CARD aa,bb)

  IF aa<bb THEN RETURN(aa)
  ELSE RETURN(bb)
  FI

PROC ClearOut()

  Position(2,17)
  FOR a=1 to 10 DO
    Put(156)
  OD
  Position(2,18)

RETURN


PROC MyError()

  ClearOut()
  IF brkkey=0 THEN
    Error=TmpErr
    Break()
  ELSEIF errno<>159 THEN
    Print("Disk Error #")
    PrintBE(errno)
    PutE()
    Print("Check the drive and ")
    PrintE("press a key.")
    ii=GetD(2)
  ELSE
    PrintE("Unexpected error.")
    Print("Check things and press ")
    PrintE("a key.")
    ii=GetD(2)
  FI
RETURN


PROC Title()

  lmargin=0
  Graphics(0)
  FOR btemp=1 to 10 DO
    Put(127)  Put(158)
  OD
  Print("              ")
  Put(159)  Put(125)
  Poke(dl+9,7)
  Poke(dl+10,6)
  Poke(710,194)
  Poke(708,198)
  Poke(712,192)
  crsinh=1

  Print("┌──────────────────────────────────────")
  Print("─────────────┐")
  Print("|   Written in ACTION! by ")
  Print("Mike Stortz  |")
  Print("|      G.R.A.S.P. of ")
  Print("Richmond, Va.     |")
  Print("└──────────────────────────────────────")
  Print("─────────────┘")
  lmargin=2
  Print("    <zero><free>      ")
  PrintE("NO EMPTY SECTORS  ")
  Print("  This program reads in ")
  PrintE("the contents")
  Print("of your binary file disks, ")
  PrintE("remembers")
  Print("their lengths, and sorts ")
  PrintE("them to")
  Print("occupy the least number ")
  PrintE("of diskettes.")
  Print("ZEROFREE will hold about ")
  PrintE("500 ")
  PrintE("programs & their lengths.")
  PutE()
  Print("  A disk has 707 free ")
  PrintE("sectors if you")
  Print("use a boot menu like ")
  PrintE("QuikLoad, or 668")
  Print("sectors minus the length ")
  PrintE("of your menu")
  PrintE("if using DOS.")
  PutE()
  Print("  A '#' will appear ")
  PrintE("before a filename")
  Print("if it is a duplicate, ")
  PrintE("or a '=' will")
  Print("appear if it is of ")
  PrintE("equal length.")
  PutE()
  Print("      ** Please press ")
  PrintE("a key ** ")
  ii=GetD(2)

  crsinh=0

  free=0
  DO
    ClearOut()
    Print("How many free sectors ")
    Print("available? ")
    free=InputC()
  UNTIL free>0 OD

  maxfiles=0
  DO
    ClearOut()
    Print("Maximum files per disk? ")
    maxfiles=InputB()
  UNTIL maxfiles>0 OD

  devc=0
  DO
    ClearOut()
    PrintE("Output to D:PRINTOUT,")
    PrintE("   screen,")
    PrintE("or printer")
    Print("  (D/S/P)? ")
```

```
    devc=GetD(2)
  UNTIL devc='D OR
        devc='P OR
        devc='S OD

  Graphics(0)
  Poke(710,194)
  crsinh=1

RETURN


PROC GetDir()

  Put(125)
  ClearOut()
  Print("▶   Now up to ")
  PrintC(max)
  PrintE(" programs.")

  num=0
  Close(1)
  Open(1,"D:*.*",6,0)
  DO
    InputMD(1,name,18)
    MoveBlock(extender+1,name+11,3)
    extender(0)=3
    ii=SCompare(extender,"SYS")
    IF name(0)>16 AND ii#0 THEN
      num==+1
      MoveBlock(hold+num*12+1,name+3,
                11)
      hold(num*12)=11
      hlen(num)=ValC(name+14)
      IF num=26 THEN EXIT FI
    FI
  UNTIL EOF(1) OD
  Close(1)

RETURN


PROC PrintDir()
  BYTE dup

  Put(125)

  IF num>0 THEN
    Print(" ")
    FOR btemp=1 TO num DO
      IF max>0 THEN
        FOR ctemp=1 TO max DO
          MoveBlock(string1+1,
                    hold+12*btemp+1,11)
          string1(0)=11
          MoveBlock(string2+1,
                    names+12*ctemp+1,11)
          string2(0)=11
          ii=SCompare(string1,string2)
          IF ii=0 AND
            hlen(btemp)=length(ctemp)
              THEN
              ii=10
          FI
          IF
            ii=0 OR ii=10 THEN EXIT
          FI
        OD
      FI

      IF ii=0 THEN
        dup='#
      ELSEIF ii=10 THEN
        dup='=
      ELSE
        dup=32
      FI
      PrintF("%C -%C%S%C ",192+btemp,
                dup,hold+12*btemp,127)
      OD
    FI
    PutE()

RETURN


PROC CopyDir()

  MoveBlock(names+12+max*12,hold+12,
            num*12)
  MoveBlock(length+2+max*2,hlen+2,
            num*2)
  max==+num

RETURN


PROC Add()

  ClearOut()
  SetBlock(string1,14,32)
  PrintE("Enter filename to add")
  PrintE("(No '.', please)")
  InputMD(0,string1,11)
  IF string1(0)=0 THEN RETURN FI
  string1(string1(0)+1)=32
  string1(0)=11
  ClearOut()
  Print("Enter length of ")
  PrintE(string1)
  addlen=InputC()
  IF addlen=0 OR addlen>400 THEN
    RETURN
  FI

  num==+1
  MoveBlock(hold+num*12,string1,12)
  hlen(num)=addlen
RETURN


PROC Delete()

  btemp==-64
  IF btemp#num THEN
    MoveBlock(hold+btemp*12,
              hold+(btemp+1)*12,
              (num-btemp)*12)
    MoveBlock(hlen+btemp*2,
              hlen+(btemp+1)*2,
              (num-btemp)*2)
  FI
  IF num>0 THEN
    num==-1
  FI

RETURN


PROC GetLibrary()

  DO
    IF idx>480 THEN EXIT FI

    PrintDir()
    ClearOut()
    Print("Insert next disk to ")
    PrintE("be cataloged")
    PrintE("and press SPACE,")
    PrintE("  LETTER to delete,")
    PrintE("  +  to add, or")
    PrintE("  ESC  to quit & print")

    btemp=GetD(2)

    IF btemp=32 THEN
```

```
      CopyDir()
      GetDir()
    ELSEIF btemp>64 and btemp<65+num
      THEN
      Delete()
    ELSEIF btemp='+ THEN
      Add()
    ELSEIF btemp='% THEN
      CopyDir()
      RETURN
    FI
  OD
RETURN

PROC PrintName()

  PrintD(1,names+which*12)
  PrintD(1," ")
  spaces==+1
  IF spaces=4 OR
    (spaces=3 AND devc='S) THEN
    spaces=0
    PutDE(1)
  FI

RETURN

PROC KeyCheck()

  IF ch<255 THEN
    ch=255
    PutE()
    FOR idx=1 TO max DO
      PrintD(1,names+programs(idx)*12)
      PrintD(1," ")
      PrintCDE(1,
              length(programs(idx)))
    OD
    PrintE("Press RETURN.")
    ii=GetD(1)
    quit=1
  FI

RETURN

PROC PrintMess()

  Put(125)
  PutE()
  PrintF("%S%U%E",
         "Programs left   - ",max)
  PrintF("%S%U%E",
         "Sectors wasted  - ",spare)
  PrintF("%S%U%E",
         "Allowable waste - ",waste)
  PutE()
  PrintE("Press any key to abort")
  PutE()
  Print("Thinking about ")
  PrintE("combinations...")
  PrintE("This many free sectors :")
RETURN

PROC Switch()

  idx=Rand(max)+1
  which=Rand(max)+1
  ctemp=programs(idx)
  programs(idx)=programs(which)
  programs(which)=ctemp

RETURN

PROC PrintOut()

  spaces=0
  FOR idx=1 TO len DO
    which=programs(idx)
    PrintName()
  OD

  IF spaces#0 THEN
    PutDE(1)
  FI

  PrintCD(1,free-sum)
  PrintDE(1," FREE")
  spare==+free-sum

  IF devc='S THEN
    PutE()
    Print("Press any key ")
    PrintE("to continue")
    btemp=GetD(2)
  FI

RETURN

PROC Remove()

  FOR idx=len+1 TO max DO
    programs(idx-len)=programs(idx)
  OD
  max==-len
  IF max=0 THEN
    Close(1)
    Close(2)
    ClearOut()
    PrintE("All done...")
    Break()
  FI

RETURN

PROC Check()

  sum=0
  b=Min(max,maxfiles)

  FOR idx=1 TO b DO
    len=idx
    ctemp=sum
    which=programs(idx)
    sum==+length(which)

    IF sum)free THEN
      sum=ctemp
      len==-1
      EXIT
    FI
  loss=free-sum
  OD
RETURN

PROC PrintLibrary()

  FOR idx=1 to max DO
    programs(idx)=idx
  OD

  PrintMess()

  DO
    atract=0
    leastloss=10000
    FOR a=1 to 10000 DO
      IF quit=1 THEN EXIT FI
      Keycheck()
      Switch()
      Check()
      IF loss(leastloss THEN
        leastloss=loss
        PrintCE(loss)
        a=1
        IF loss=0 THEN EXIT FI
      FI
    OD

    waste=leastloss

    FOR a=1 TO 10000 DO
      IF quit=1 THEN EXIT FI
      Keycheck()
      Switch()
      Check()
      IF loss(=leastloss THEN
        a=1
        PrintOut()
        Remove()
        PrintMess()
      FI
    OD
    IF quit=1 THEN EXIT FI
  OD

PROC Main()

  Close(2)
  Open(2,"K:",4,0)
  TmpErr=Error
  Error=MyError
  idx=0  spare=0  waste=0   quit=0
  max=0  num=0    shflok=64
  ZERO(hold,240)

  Title()
  GetLibrary()

  Close(1)
  IF devc='D THEN
    ClearOut()
    Print("Insert disk to ")
    PrintE("hold D:PRINTOUT")
    PrintE("  and press any key")
    btemp=GetD(2)
    Open(1,"D:PRINTOUT",8,0)
  ELSEIF devc='S THEN
    Open(1,"E:",12,0)
    Poke(710,194)
    Poke(708,198)
  ELSE
    Open(1,"P:",8,0)
  FI

  crsinh=1
  PrintLibrary()

  Close(1)
  Close(2)
  Error=TmpErr

RETURN
```

●