

Program Allows Disk Timesharing to Read Non-Timesharing Diskettes

By: Gale Schonfeld
MITS

Many of you are now sharing our excitement over the new Altair Timesharing BASIC. Those of you who have the disk version may be perturbed about a problem with loading 4.0 or 4.1 Disk BASIC program files under Timesharing. However, with only a few minutes of your time and the computer's, the problem can be solved.

In the disk version of Timesharing BASIC, an optional password may be specified during SAVEing of a program. In regular Disk BASIC, the password facility is not provided. Therefore, the problem may occur when a LOAD or RUN command is issued in Timesharing for a program on a regular BASIC disk. Timesharing may respond to the command with PASSWORD FOR FILE "XXX. . .?", and the user will not know with what password to answer.

This problem is due to the format of the directory track on the diskettes. To review, each sector of the directory track is comprised of eight file name slots. Each slot contains 16 bytes--eight bytes for the file name, one byte for the track pointer, one byte for the sector pointer, one byte indicating whether the file is random or sequential and in regular Disk BASIC, and five unused bytes normally set to nulls. In Timesharing Disk BASIC, these extra five bytes are used for passwords. Occasionally, "garbage" can get into these extra bytes on the normal BASIC diskettes. When Timesharing tries to access these files, it "sees" a password which the user is unaware. If all five bytes are null, Timesharing realizes that a password is not required.

The following program, when executed in 4.0 or 4.1 Disk BASIC, will correct the directory track of a 4.0 or 4.1 diskette. The functions of PASSCHEK are to set the last five bytes of the file name slots to nulls and recalculate the checksum of the sector so it can be read by Timesharing. The program PASSCHEK contains detailed comments regarding its execution. The

remark statements can be left out when entering the program in order to utilize a minimum amount of memory.

To use PASSCHEK, enter it into memory using 4.0 or 4.1 Disk BASIC. (It will **not** run in Timesharing.) Place the diskette you need to correct in **Disk Drive** and **MOUNT** it. Now type **RUN**. PASSCHEK will run for approximately two to three minutes, printing "DONE - CHECK USING PIP DAT COMMAND" when it's finished. If you wish to check using P10, the format of the floppy disk is described in Appendix H of the Altair BASIC Manual.

For those of you who have old 3.4 Disk BASIC program files that you want to run under Timesharing Disk BASIC, a few extra steps are needed before running PASSCHEK on the 3.4 diskette. Since Timesharing will read only 4.0 or 4.1 formatted files, you must convert your 3.4 files to the 4.0 format. This is easily done by first **LOADing** and then **re-SAVEing** all 3.4 program files in ASCII (e.g. **SAVE "XXX", O, A**), using 3.4 Disk BASIC, and then using the 4.0 PIP **CNV** command on the diskette to convert the files to the 4.0/4.1 format. After this, you can run PASSCHEK.

Program

```

10 CLEAR 500
20 '
      LINES 30-80 POSITION DISK HEAD TO TRACK 70

30 DT=70          ' DESIRED TRACK IS 70
40 IF (INP(8) AND 64)<>0 THEN WAIT 8, 2, 2: OUT 9, 2:
   GOTO 40
50              ' TEST FOR TRACK 0, IF NOT AT 0 STEP HEAD OUT ONE
              ' TRACK AND TEST AGAIN
60 IF DT<0 OR DT>76 THEN PRINT "ERROR": STOP
70 FOR K=1 TO DT: WAIT 8, 2, 2: OUT 9, 1: NEXT K
80              ' STEP DISK HEAD IN DT TRACKS, TO TRACK 70
90 '

      LINES 100-160 GET EACH SECTOR OF TRACK 70 AND REPLACE
      5 BYTES OF FILE SLOT WITH NULLS

100 FOR SC=0 TO 31 ' GET EACH SECTOR OF TRACK 70
110 AS=DSKIS(SC)  ' READ CURRENT SECTOR
120 FOR SL=0 TO 7 ' GET EACH FILE NAME SLOT (8 SLOTS/SECTOR)
130 YS=STRINGS(5,0)
140 MIDS(AS, 19+(SL*16), 5)=YS
150              ' REPLACE LAST 5 BYTES OF EACH FILE NAME
              ' SLOT WITH NULLS
160 NEXT SL      ' GET NEXT SLOT
170 '

      LINES 190-290 CORRECT CHECKSUM BYTE OF EACH SECTOR AND
      PUT MODIFIED SECTOR BACK ON DISK

180 CK=0        ' SET CHECKSUM COUNTER TO ZERO
190 FOR I=6 TO 135 ' ADD UP BYTES 6 THROUGH 135
200 CK=CK+ASC(MIDS(AS, I, 1))
210 NEXT I
220 FOR J=3 TO 4 ' ADD BYTES 3 AND 4 TO THE SUM OF 6-135
230 CK=CK+ASC(MIDS(AS, J, 1))
240 NEXT J
250 CK=CK AND 255 ' MASK OUT HIGH ORDER 8 BITS SO THAT CHECK-
              ' SUM IS ONLY ONE BYTE
260 MIDS(AS, 5, 1)=CHR$(CK) ' REPLACE BYTE 5 OF THE SECTOR WITH
              ' NEW CHECKSUM BYTE
270 DSKOS AS, SC ' PUT MODIFIED SECTOR BACK ON DISK
280 NEXT SC      ' GET NEXT SECTOR
290 PRINT "DONE - CHECK USING PIP DAT COMMAND"
300 END
OK

```