

8K Sector Copier Instructions	2
Introduction	2
Running ADAM.....	3
Files on the Disk	3
8KCOPIER.ADM	4
8KCopier.BAS	5
8KLEVELC.ADM.....	5
8KSECTAG.ADM	6
BATLOAD.ADM.....	7
BATMAN.ADM.....	7
UTILITY.BAS	8

8K Sector Copier Created by Jason Brooks (aka Argonaut) 1990

Introduction

In 1990 Speedlock created a disk protection system for the Amstrad CPC which ensured that standard hardware could not copy the disk.

The Disk sector parameters were modified such that any program reading the information would be presented with a single sector per track size of 8K, however physically the disk could only hold a maximum of 6k data per track.

You could fool the disk controller to format a sector with 8K by modifying the same parameters, however when formatting and writing the data to the sector, there was an inevitable overrun invalidating the disk and would appear to be corrupt/unformatted.

Conversely, reading 6K sectors was a trivial task since you can set the maximum amount of data to read into a buffer.

Hackers including myself took on the challenge, trying to spin the disk and stop the motor such that the additional data would not overwrite the sector information/overlap the data on the track.

These attempts all failed, it was possible to achieve this by modifying the Amstrad CPC hardware by adding transistor to the write signal but this was impractical and with risk to all by the hardened enthusiast.

I approach the problem in a different way. We'd been quiet adept at breaking various protection systems, transferring tape to disk over the years, and so this problem was simply one of transferring files from a protected disk to one that could be read and copied by any copier on the market.

Intercepting the decryption routines to find the loader, load the game code and then write this to disk. Some games had additional levels, and as such didn't use the 8K Sector format but standard Data/Vendor formats so it was possible to copy the game level sectors using a combination of cracking the main game, and copying the game level data.

The code I have shared with you, worked on the examples of 8K Sector Disks I could find at the time, but it doesn't mean that it's completely fool proof.

My main assembler of choice was ADAM (Assembler/Disassembler and Monitor) also known as DAMS in Europe. It was probably one of the most underestimated tools available to hacking and development, using a combination of loading the application into a 6128 rambank it was possible to speed up development.

Of course I had Maxam, but because of the tokenisation of code and built in monitor tools it was (in my opinion) the best on the market for the humble CPC.

Of course, 28 years later, I'm sure there are better people than I who can make use of this code for education purposes.

Running ADAM

Insert the disk and type

```
RUN"ADAM"
```

This will automatically load into ram address &4000 (16384 Decimal)

If you have a CPC6128 you can issue the following

```
OUT &7F00,196
```

This will load ADAM into RAM 4 at address &4000

Should you crash your code, or reset your Amstrad, usually switching back that rambank and call &4000 will get you back up and running without loading the tool.

All ADAM Source files will have the file extension **.ADM**

```
Ready out &7f00,196
Ready
Ready run"adam"
Loaded at 16384 (4000 hex)
```

```

                                     LRAM  URAM  Dec.  Min.
.COPYRIGHT 1985 MICRO-APPLICATION.
.DAMS.
❖
```

Files on the Disk

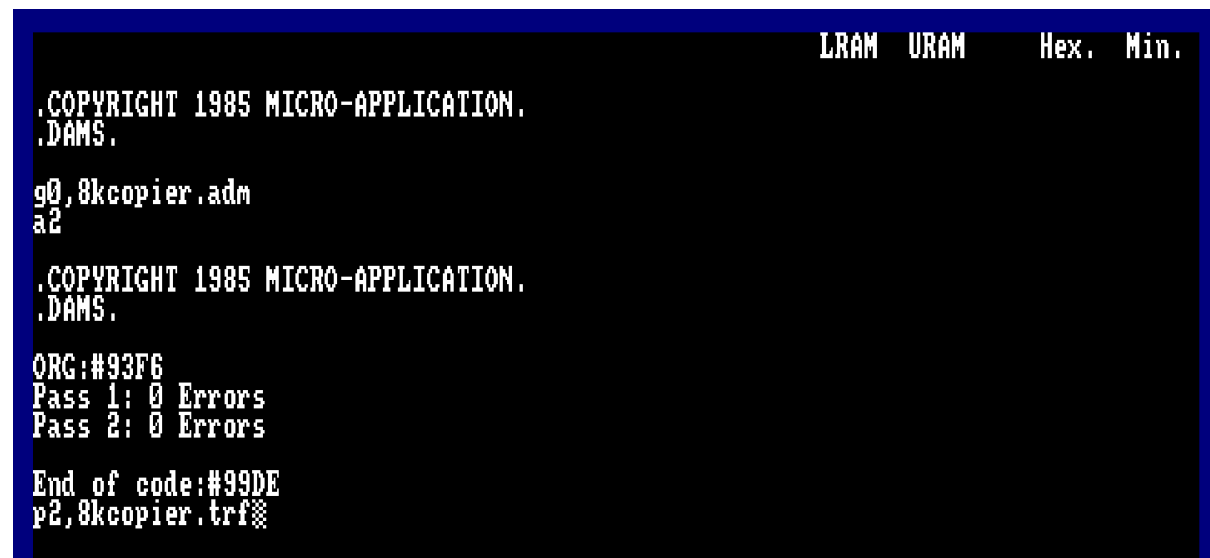
```
Drive B: user 0
8KCOPIER.ADM  7K  8KLEVELC.ADM 11K  8KSECTAG.ADM  4K  BATLOAD .ADM  2K
8KCOPIER.BAS  2K  8KLEVELC.BIN 2K  ADAM .BAS  1K  BATMAN .ADM  7K
8KCOPIER.TRF  2K  8KSECT .BIN  1K  ADAM .BIN 12K  UTILITY .BAS 5K
122K free
Ready
```

8KCOPIER.ADM

To Assemble this code, from within ADAM use the following :-

Press CTRL+B to enter HEX Mode

```
g0,8kcopier.adm
a2
p2,8kcopier.trf
```



The screenshot shows the ADAM assembler interface with a black background and white text. At the top right, there are labels for 'LRAM', 'URAM', 'Hex.', and 'Min.'. The main display area shows the following text:

```
.COPYRIGHT 1985 MICRO-APPLICATION.
.DAMS.
g0,8kcopier.adm
a2
.COPYRIGHT 1985 MICRO-APPLICATION.
.DAMS.
ORG:#93F6
Pass 1: 0 Errors
Pass 2: 0 Errors
End of code:#99DE
p2,8kcopier.trf
```

G0 will Get the file from disk.

A2 will assemble the code at the first available address space after the assembler has loaded. We use this command as I have relocating code and different ORG Addresses.

P2 will Put the object code onto disk.

X1 will display the symbol table if you're interested.

More instructions on the use of ADAM can be found here

https://cpcrulez.fr/download.php?a=VlesnZvWgsLI2M69gsbe29PNgLXk0M7liH_ZzdLNT7rp2Q==

```

xl
start #1000 filetype #104D entry1 #105F wrong #1067
CPM #108C DISK #1093 DISK1 #10A8 D11 #10AB
return #10C2 waster #10D6 detect #10E7 t1 #10F4
t2 #10FD t3 #1106 t4 #110F t5 #1118
t6 #1121 t7 #1186 calcjrpo #1189 calcjpz #11CB
calcjpnz #1201 execute #121D jpnz1 #1226 checkout #1280
calcjp #1293 compare #12B6 compare1 #12BB compare2 #12C5
compare3 #12C9 swichcol #12CB Restore #12ED
refresh #12F2 Restore1 # 16 copyrest #1303 copypres #1323
Preserve #132C Preserve1 # 1D Stackreg #BFF0 naddress #1349
naddressx #134B Decrypt #2000 Border #134D offset #134E
number #1350 name #1352 len # 04 comm #1356
read #1357 adam #135A adam1 #4000 adama #135C
copygame #136D end1 #1370 end2 #1384 end2a #1386
end3 #1391 end3a #13C1 md1 #13C4 mdd #13E2
sc1 #13EB sc2 #1407 sc11 #140E sc2a #141D
sc22 #1448 save #145C save1 #B017 barb #B01C
barb1 #B02B bd37 #B034 savek #B054 savex #B074
saves #B07A save2 #B07C saver #B08E saven #B0A7
slen # 05 load #B0AC set1 #BE9C mode #BEAD
xdest #BEEC barb1 #BEC5 loader #BED4 name1 #BEE7
len1 # 05 inks #BEEC execcode #BEFC
Text:#68F5 End:#81E7 #18F2 Bytes
Hmem:#8D39

```

8KCopier.BAS

This is the basic file for loading and executing the 8K Copier Code

```

LOAD"8KCOPIER.BAS"
LIST
RUN

```

Follow the instructions

8KLEVELC.ADM

To be able to assemble this, you will need to load ADAM into a lower memory address.

```

RUN"ADAM

```

Immediately after hitting enter, hold down the SPACE bar and you are presented with the option of a loading address.

```

Load address (in decimal) ? 1000

```

Enter 1000 and hit enter

Once loaded enter the following commands :-

CTRL+B to enter HEX Mode

```
G0,8klevelc.adm
A2
P2,8klevelc.bin
```

```
LRAM  URAM  Hex.  Min.

.COPYRIGHT 1985 MICRO-APPLICATION.
.DAMS.

g0,8klevelc.adm
a2

.COPYRIGHT 1985 MICRO-APPLICATION.
.DAMS.

ORG:#7A63
Pass 1: 0 Errors
Pass 2: 0 Errors

End of code:#8120
p2,8klevelc.bin
```

8KSECTAG.ADM

```
G0,8ksectag.adm
A2
P2,8ksect.bin
```

```
LRAM  URAM  Hex.  Min.

.COPYRIGHT 1985 MICRO-APPLICATION.
.DAMS.

g0,8ksectag.adm
a2

.COPYRIGHT 1985 MICRO-APPLICATION.
.DAMS.

ORG:#72EB
Pass 1: 0 Errors
Pass 2: 0 Errors

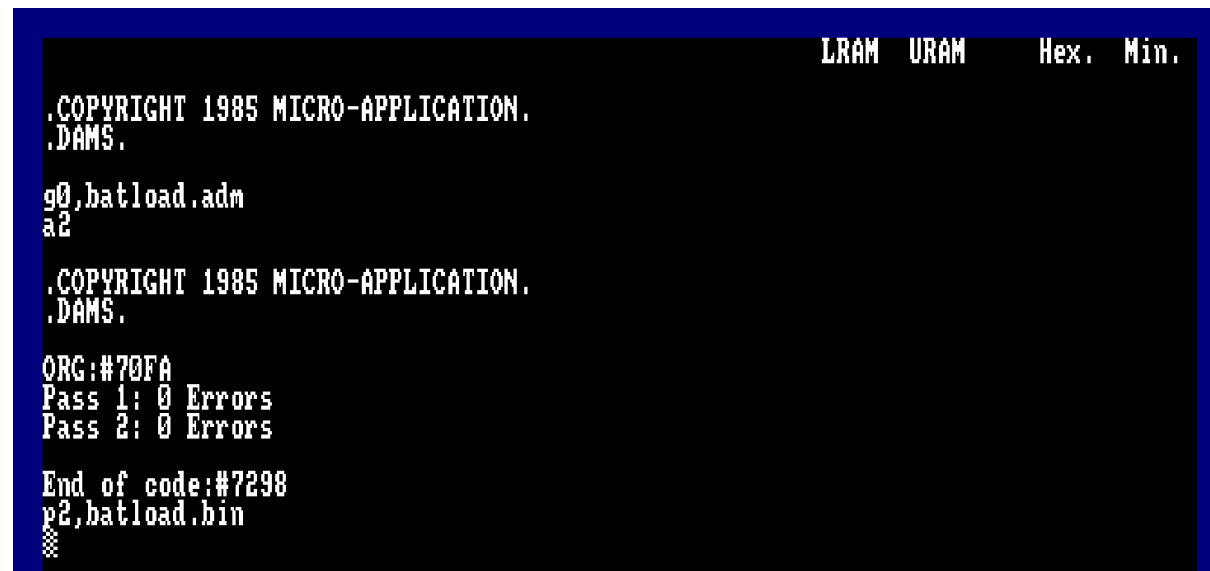
End of code:#7641
p2,8ksect.bin
```

BATLOAD.ADM

This will load the Batman game and offer a cheat mode.

```
G0,batload.adm
A2
P2,batload.bin
B

Run"batload.bin",&8000
```



```
LRAM  URAM  Hex.  Min.

.COPYRIGHT 1985 MICRO-APPLICATION.
.DAMS.

g0,batload.adm
a2

.COPYRIGHT 1985 MICRO-APPLICATION.
.DAMS.

ORG:#70FA
Pass 1: 0 Errors
Pass 2: 0 Errors

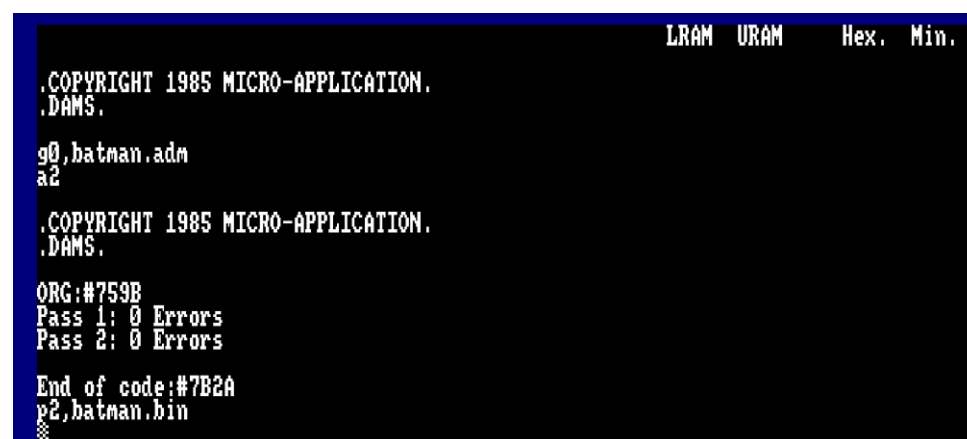
End of code:#7298
p2,batload.bin
█
```

BATMAN.ADM

The Speedlock decoder for Batman the caped crusader

```
G0,batman.adm
A2
P2,batman.bin
B

RUN"BATMAN.BIN",&3000
```



```
LRAM  URAM  Hex.  Min.

.COPYRIGHT 1985 MICRO-APPLICATION.
.DAMS.

g0,batman.adm
a2

.COPYRIGHT 1985 MICRO-APPLICATION.
.DAMS.

ORG:#759B
Pass 1: 0 Errors
Pass 2: 0 Errors

End of code:#7B2A
p2,batman.bin
█
```

UTILITY.BAS

This utility will attempt to copy game level sectors for you on the cracked 8K loader

```
RUN"UTILITY.BAS"
```

```
*****  
*                                     *  
*      THE ARGONAUT'S SPEEDLOCK GAME LEVEL DISC COPIER.      *  
*                                     *  
*                                     *  
*      1) Format Disk                                         *  
*                                     *  
*      2) Copy Game Level Data                               *  
*                                     *  
*      3) Catalogue A Disk                                    *  
*                                     *  
*          >>> Please Select An Option <<<  
*                                     *  
*                                     *  
*                                     *  
*                                     *  
*                                     *  
*                                     *  
*                                     *  
*                                     *  
*                                     *  
*                                     *  
*                                     *  
*                                     *  
*                                     *  
*                                     *  
*                                     *  
*                                     *  
* Source Drive : A    [D] To Toggle Destination Drive.   Destination Drive : A  
*                                     *  
*****
```