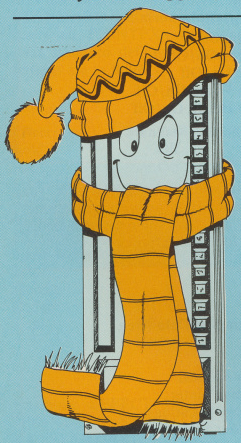
Anti-Freeze

Protect yourself against pirates with this ingenious new program

By Neil Higgins



In past issues of *Your Commodore*, there have been many articles on the subject of program protection – indeed, I've even written one myself. One thing they all had in common was that they only described the simple ways of protection, such as disabling the run stop key, detecting a hardware reset or even using a secret password to prevent access to a program. These methods worked well, but how can we protect ourselves against the freezer systems now used on the latest state-of-the-art back-up cartridges? Well, in this article I will explain one way in which machine code programmers can defeat such systems.

The most common cartridges for the Commodore 64 at present have to be 'Action Replay', 'The Expert', 'Final Cartridge' and 'Freeze Frame', all of which have a built-in freeze button which is supposed to be able to stop any program while it is running and make a back-up. If you have any of these, then stay tuned! I was only able to test my anti-freeze routine on two versions of these cartridges – namely the Final Cartridge 2 and Action Replay MK 3, and in both cases it proved highly successful.

Eureka!

Before I explain how to use the anti-freeze, I will tell you how I first came across it. It all started when I was sitting around one day wondering what my next project would be. After an hour spent staring blankly at my computer equipment, I noticed my 'Action Replay' cartridge, and I had a brainwave – why not try and detect whenever the freeze button had been pressed! But where on earth would I start?

I decided not to try using non-maskable interrupts or other equally precarious methods. I thought there would have to be at least one location somewhere in memory that changes whenever that button is pressed. Some obvious memory locations to test were around the CIA chips from $DC00 to $DD0F, and the stack from $0100 to $01FF, so after a length of time around the CIA s without getting anywhere, I concentrated on the stack area.

Since 'Action Replay' is supposed to have a system that leaves the stack contents intact, I decided to fill it with a random byte (I filled $0100 to $01FF with $00) and then run a routine which would set the stack pointer. After countless attempts at pressing the freezer and checking the stack, I finally set the stack pointer to 25, then again cleared the stack area and pressed the freeze button, having entered the built in monitor. I then noticed that location $0103 had changed, and thus my anti-freeze program was born.

Trying it out

The routine is given as a source code listing which was written using the Micro Assembler, but it should be compatible with most other assemblers (you may just need to change the TXT pseudo ops to BYT). A basic loader is also given for those without assemblers. To test it on your cartridge, load it into memory and start it with SYS 49152. A message will appear asking you to press the freeze button, and after doing so you must return to it by using the 'RUN' option in your cartridge. If the test was successful, the message 'I gotcha! You have just used the freeze button ' will bé displayed, at which point you can re-test to satisfy yourself that it works. If it crashes on return, then you could also call it successful, as it means a working back-up cannot be made.

Machine coders can of course add their own routines onto this, such as filling the whole of memory with random bytes, or even printing a silly message to try on your mates. If you do edit anti-freeze for your own programs, it's important to note that since the stack is reduced to 25, you are severely limited as to the number of nested subroutines or ROM routines you can call. Also, check your usage of the instruction PHA, in other words keep an eye on the stack pointer.

The routine also needs to use location $0103, but this is also used by certain rom routines – such as the one at $BDDD which converts the contents of floating point accumulator 1 to a string starting at $0100 – so if you intend to use the ROM; make sure you know what memory addresses are used, or it could be disastrous.

As there are new versions of cartridges coming onto the market every few months, I can't tell you if anti-freeze would successfully work with them. I hope you will try it out yourselves, and if anyone out there has already got all the latest ones, then why not send in the results to *Your Commodore* and demand they be put in the Mailbag for us all to see. Anyway, I hope anti-freeze helps develop your own programs to stamp out the pirates. YC

Listings

Flow of Ideas

PROGRAM: DIRECTOR

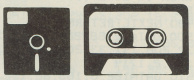
```
BE  100 POKE53280,0:POKE53281,0:
    GOTO440
39  110 GET#8,A$:IFA$=""THENA$=C
    HR$(0)
F6  120 RETURN
55  130 INPUT#15,E,E$,T,S
63  140 IFE>0THENPRINT"[C3J]"E;E$
    " "T","S:END
1B  150 RETURN
9F  160 GOSUB110
07  170 DN$=DN$+A$
3A  180 RETURN
78  190 GET#8,A$:IFA$=""THEN190
A0  200 IFSTTHEN430
AB  210 IFASC(A$)<128THENF$="DEL
    ":GOTO260
20  220 IFASC(A$)AND56THEN190
BD  230 P$=" ":F=(ASC(A$)AND63):
    F=F+2*(F-1):IFASC(A$)AND64TH
    ENP$="[RVSON]<RVSOFF>"
E9  240 F$=MID$("SEQPRGUSRREL",F
    ,3)
C9  250 GOSUB110
DA  260 I$=RIGHT$(" "+STR$(ASC(A
    $)),2)
B2  270 GOSUB110
95  280 S$=RIGHT$(" "+STR$(ASC(A
    $)),2)
67  290 N$=""
53  300 FORA=1TO16
2B  310 GET#8,A$:N$=N$+A$
C3  320 NEXT
BA  330 FORA=1TO9:GET#8,A$:NEXT
6B  340 GOSUB110
17  350 BL=ASC(A$)
54  360 GOSUB110
EF  370 BL=BL+ASC(A$)*256:IFF$<>
    "DEL"THENBT=BT+BL
D5  380 B$=RIGHT$("  "+STR$(BL),
    3)
4C  390 PRINT"[WHITE]";:IFF$="DE
    L"THENPRINT"[C1J";
B2  400 POKE212,128
99  410 PRININ$;:POKE212,0
A9  420 PRINT"[CBJ"P$" [C6J"F$"
    [C3J"B$"[5PC3,C7J"I$"[5PC4J
    "S$
21  430 RETURN
F1  440 PRINT"[CLR,YELLOW,REV N,
    SLJOADING HEADER BLOCK"
BA  450 OPEN15,8,15:PRINT#15,"IO
    ":GOSUB130
CD  460 BT=0:DIMBM%(35)
75  470 OPENB,8,1,"$,S,R":GOSUB1
    30
E2  480 FORA=1TO2:GET#8,A$:NEXT:
    A=0
14  490 FORA=1TO35
C8  500 GOSUB110
22  510 BM=BM+ASC(A$):BM%(A)=RIG
    HT$(" "+STR$(ASC(A$)),3)
15  520 GET#8,A$:GET#8,A$:GET#8,
    A$
96  530 NEXT
F5  540 DN$="[CLR,REV N,YELLOW,S
    DJISK: [RVSONJ"+CHR$(34)
3E  550 FORA=1TO16:GOSUB160:NEXT

F3  560 DN$=DN$+CHR$(34)
FB  570 FORA=1TO15:GOSUB160:NEXT

BF  580 PRINTDN$:B=0
82  590 PRINT"[DOWN,WHITE,SFJILE
    NAMEtSPC9,C6,STJYPE [C3,SSJI
    ZE [C7,STJRACK [SSJECTORtDOW
    NJ"
82  600 GOSUB190:B=B+1:IFF$<>"DE
    L"THENPT=PT+1
8E  610 IFSTTHEN630
EB  620 IFB<20THEN600
78  630 POKE198,0:PRINTTAB(12)"[
    RED,RVSONJ [SPJRESS [SAJNY [
    SKJEY [RVSOFFJ";
F9  640 GET K$:IFK$=""THEN640
EA  650 IFST=0THEN580
2D  660 GOSUB130:CLOSE8:CLOSE15
FC  670 PRINT"[CLR,WHITE,SPC11,S
    VJITAL [SSJTATISTICS"
77  680 PRINT"[SPC11,CT16J"
5F  690 PRINT"[YELLOW,SNJUMBER O
    F PROGRAMS ="PT
FD  700 PRINT"[SBJLOCKS USED ="B
    I
5D  710 BF=BM-VAL(BM$(18))
21  720 PRINT"[SBJLOCKS FREE ="B
    F;
B7  730 A$=""
1B  740 IF664-BT<>BFTHENA$="[C3,
    SBJLOCK COUNT MISMATCH![YELL
    OWJ"
8E  750 PRINTA$
B6  760 PRINT"[SDJIRECTORY BLOCK
    S FREE ="BM%(18)
84  770 A$="[GREEN,S-,SPC36,S-J"

07  780 PRINT"[DOWN,GREEN,CA,S*3
    0,CSJ"
66  790 PRINT"[GREEN,S-,SFJREE T
    RACK SECTORS:[SPC17,S-J"
63  800 PRINT"[GREEN,S-,CT19,SPC
    17,S-J"
F4  810 PRINT"[GREEN,S-,C8,STJRA
    CK:  1  2  3  4  5  6  7  8
     9 10[GREEN,S-J"
DD  820 PRINT"[GREEN,S-,C7,STJOT
    AL:";:FORA=1TO10:PRINT"[C7J"
    ;:IFVAL(BM$(A))=21THENPRINT"
    [C2J";
86  830 PRINTBM$(A);:NEXT:PRINT"
    [GREEN,S-J":PRINIA$
66  840 PRINT"[GREEN,S-,C8,STJRA
    CK: 11 12 13 14 15 16 17 18
    19 20[GREEN,S-J"
91  850 PRINT"[GREEN,S-,C7,STJOT
    AL:";:FORA=11TO17:PRINT"[C7J
    ";:IFVAL(BM$(A))=21THENPRINT
    "[C2J";
5E  860 PRINTBM$(A);:NEXT
E0  870 FORA=18TO20:PRINT"[C7J";
    :IFVAL(BM$(A))=19THENPRINT"[
    C2J";
98  880 PRINTBM$(A);:NEXT:PRINT"
    [GREEN,S-J":PRINIA$
04  890 PRINT"[GREEN,S-,C8,STJRA
    CK: 21 22 23 24 25 26 27 28
    29 30[GREEN,S-J"
DB  900 PRINT"[GREEN,S-,C7,STJOT
    AL:";:FORA=21TO24:PRINT"[C7J
    ";:IFVAL(BM$(A))=19THENPRINT
    "[C2J";
14  910 PRINTBM$(A);:NEXT
4E  920 FORA=25TO30:PRINT"[C7J";
    :IFVAL(BM$(A))=18THENPRINT"[
    C2J";
AA  930 PRINTBM$(A);:NEXT:PRINT"
    [GREEN,S-J".PRINIA$
6E  940 PRINT"[GREEN,S-,C8,STJRA
    CK: 31 32 33 34 35[SPC15,GRE
    EN,S-J"
9A  950 PRINT"[GREEN,S-,C7,STJOT
    AL:";:FORA=31TO35:PRINT"[C7J
    ";:IFVAL(BM$(A))=17THENPRINT
    "[C2J";
50  960 PRINTBM$(A);:NEXT:PRINT"
    [5PC15,GREEN,S-J"
BA  970 PRINT"[CZ,S*36,CXJ"
```

Anti-Freeze

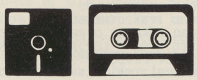


PROGRAM: ANTI-FREEZE

```
A1  100 REM ********************
    *********
3C  110 REM * ANTI-FREEZE BASIC
    LOADER  *
86  120 REM *-------------------
    --------*
96  130 REM * BY N.HIGGINS JANUA
    RY 1988 *
8A  140 REM *-------------------
    --------*
56  150 REM *  DEIECTS FREEZER B
    UTTON.  *
E5  160 REM ********************
    *********
25  170 REM
3B  180 REM
97  190 PRINTCHR$(147):POKE53281
    ,0:POKE646,7
5C  200 PRINT"[DOWN2,RIGHTJPLEAS
    E WAIT - POKEING IN MACHINE
    CODE"
58  210 GOSUB460:POKE53280,0
FF  220 PRINT"[CLR,DOWN2,RIGHT2J
    ANTI-FREEZE M/C NOW IN MEMOR
    Y"
6C  230 PRINT"[DOWN2,RIGHT2JSELE
    CT OPTION (1-3)"
13  240 PRINT"[DOWN2,RIGHT2J(1)
    SAVE CODE TO DISK"
CC  250 PRINT"[DOWN,RIGHT2J(2) S
    AVE CODE TO TAPE"
FF  260 PRINT"[DOWN,RIGHT2J(3) E
    XIT"
7F  270 GETA$:IFA$=""THEN270
EB  280 A=VAL(A$):IFA<1ORA>3THEN
    270
9C  290 IFA=3THEN410
FF  300 A$="PREPARE CASSETTE FOR
```

```
   SAVING":D=1
5E 310 IFA=1THENA$="PLACE REQUI
   RED DISK IN DRIVE":D=8
CO 320 PRINT"{CLR,RIGHT}";A$:PR
   INT"{DOWN3,RIGHT3}PRESS ANY
   KEY WHEN READY"
BE 330 POKE198,0:WAIT198,1:POKE
   198,0
5A 340 REM ------ SAVE MACHINE
   CODE -------
B2 350 SYS57812"ANTI-FREEZE M/C
   ",D,1:REM * FILENAME *
E7 360 POKE193,0:POKE194,192:RE
   M * START ADDRESS TO SAVE ($
   C000) *
71 370 POKE174,104:POKE175,193:
   REM * END ADDRESS TO SAVE-1
   ($C168) *
F2 380 REM
C5 390 SYS62957:REM * PERFORM S
   AVE *
1E 400 REM
A4 410 PRINT"{CLR,DOWN,RIGHT}OK
   AY - SYS49152 TO TEST CARTRI
   DGE"
25 420 END
20 430 REM
36 440 REM
4C 450 REM
F1 460 FORL=0TO22:CX=0:FORD=0TO
   15:READA:CX=CX+A:POKE49152+L
   *16+D,A:NEXTD
F6 470 POKE53280,(PEEK(53280)AN
   D15)+1
DB 480 READA:IFA<>CXTHENPRINT"E
   RROR IN LINE";500+(L*10):STO
   P
1D 490 NEXTL:RETURN
9D 500 DATA32,68,229,169,0,141,
   32,208,141,33,208,169,7,141,
   134,2,1714
0B 510 DATA162,0,189,82,192,240
   ,6,32,210,255,232,208,245,16
   2,25,154,2394
FF 520 DATA169,200,141,3,1,173,
   3,1,201,200,240,249,162,255,
   154,32,2184
8A 530 DATA68,229,169,10,141,13
   4,2,162,0,189,253,192,240,6,
   32,210,2037
4D 540 DATA255,232,208,245,169,
   0,133,198,32,228,255,201,32,
   208,249,76,2721
A3 550 DATA0,192,13,13,32,32,32
   ,32,32,32,32,32,32,65,78,84,
   733
3A 560 DATA73,45,70,82,69,69,90
   ,69,32,82,79,85,84,73,78,69,
   1149
6B 570 DATA13,32,32,32,32,32,32
   ,32,32,32,45,45,45,45,45,45,
   571
2A 580 DATA45,45,45,45,45,45,45
   ,45,45,45,45,45,13,13,13,
   624
41 590 DATA32,32,32,80,76,69,65
   ,83,69,32,80,82,69,83,83,32,
   999
32 600 DATA84,72,69,32,70,82,69
   ,69,90,69,32,66,85,84,84,79,
   1136
B8 610 DATA78,32,79,78,13,13,32
   ,32,32,89,79,85,82,32,67,65,
   888
76 620 DATA82,84,82,73,68,71,69
   ,44,84,72,69,78,32,85,83,69,
   1145
2C 630 DATA32,84,72,69,32,39,82
   ,85,78,39,13,13,32,32,32,79,
   813
31 640 DATA80,84,73,79,78,32,79
   ,82,32,83,73,77,73,76,65,82,
   1148
1B 650 DATA32,84,79,32,82,69,84
   ,85,82,78,46,13,0,13,13,32,8
   24
A0 660 DATA32,32,73,32,71,79,84
   ,67,72,65,32,33,32,89,79,85,
   957
C3 670 DATA32,72,65,86,69,32,74
   ,85,83,84,32,85,83,69,68,32,
   1051
75 680 DATA84,72,69,13,13,32,32
   ,32,32,32,32,32,32,32,32,32,
   603
ED 690 DATA70,82,69,69,80,69,32
   ,66,85,84,84,79,78,46,13,13,
   1029
B3 700 DATA13,32,32,32,80,82,69
   ,83,83,32,84,72,69,32,39,83,
   917
13 710 DATA80,65,67,69,66,65,82
   ,39,32,84,79,32,84,82,89,32,
   1047
0E 720 DATA65,71,65,73,78,13,0,
   10,101,8,53,40,21,8,85,74,76
   5
```

Multi-Coloured List



PROGRAM: MULTI-COLOURED LIST

```
A1 100 REM ********************
   *********
B6 110 REM *    MULTI-COLOURED
   LIST     *
86 120 REM *-------------------
   --------*
94 130 REM *  BY N.HIGGINS DECE
   MBER 88  *
8A 140 REM *-------------------
   --------*
E6 150 REM * SYS53076,LINE NO,C
   OL(0-15)*
E5 160 REM ********************
   *********
25 170 REM
3B 180 REM
97 190 PRINTCHR$(147):POKE53281
   ,0:POKE646,7
5C 200 PRINT"{DOWN2,RIGHT}PLEAS
   E WAIT - POKEING IN MACHINE
   CODE"
7C 210 GOSUB550:POKE53280,0
CO 220 PRINT"{CLR,DOWN2,RIGHT2}
   MULTI-COLOURED LIST M/C NOW
   IN MEMORY"
6C 230 PRINT"{DOWN2,RIGHT2}SELE
   CT OPTION (1-3)"
13 240 PRINT"{DOWN,RIGHT2}(1)
   SAVE CODE TO DISK"
CC 250 PRINT"{DOWN,RIGHT2}(2) S
   AVE CODE TO TAPE"
FF 260 PRINT"{DOWN,RIGHT2}(3) E
   XIT"
7F 270 GETA$:IFA$=""THEN270
E8 280 A=VAL(A$):IFA<1ORA>3THEN
   270
9C 290 IFA=3THEN410
FF 300 A$="PREPARE CASSETTE FOR
   SAVING":D=1
5E 310 IFA=1THENA$="PLACE REQUI
   RED DISK IN DRIVE":D=8
CO 320 PRINT"{CLR,RIGHT}";A$:PR
   INT"{DOWN3,RIGHT3}PRESS ANY
   KEY WHEN READY"
BE 330 POKE198,0:WAIT198,1:POKE
   198,0
5A 340 REM ------ SAVE MACHINE
   CODE -------
63 350 SYS57812"M.COL.LIST M/C"
   ,D,1:REM * FILENAME *
4D 360 POKE193,32:POKE194,207:R
   EM * START ADDRESS TO SAVE (
   $CF20) *
1C 370 POKE174,245:POKE175,207:
   REM * END ADDRESS TO SAVE-1
   ($CFF4) *
F2 380 REM
C5 390 SYS62957:REM * PERFORM S
   AVE *
1E 400 REM
49 410 PRINT"{CLR,DOWN,RIGHT}OK
   AY - SYS53076,LINE NO,COLOUR
   (0-15)"
25 420 END
20 430 REM
91 540 REM
53 550 FORL=0TO13:CX=0:FORD=0TO
   15:READA:CX=CX+A:POKE53024+L
   *16+D,A:NEXTD
CC 555 POKE53280,(PEEK(53280)AN
   D15)+1
E7 560 READA:IFA<>CXTHENPRINT"E
   RROR IN LINE";1160+(L*10):ST
   OP
0A 570 NEXTL
C8 580 RETURN
DF 590 REM
D5 600 REM
E1 620 REM
DA 630 DATA 144,5,28,159,156,30
   ,31,158,129,149,150,151,152,
   153,154,155,1904
6C 640 DATA 82,69,77,34,141,5,0
   ,13,32,76,73,78,69,32,69,88,
   938
AF 650 DATA 73,83,84,83,46,67,7
   9,78,84,73,78,85,69,63,32,89
   ,1166
DC 660 DATA 47,78,13,0,32,253,1
   74,32,107,169,165,20,133,251
   ,165,21,1660
16 670 DATA 133,252,32,155,183,
   224,16,144,3,76,8,175,134,25
   3,32,19,1839
6C 680 DATA 166,144,27,162,0,18
   9,55,207,240,6,32,210,255,23
   2,208,245,2378
DD 690 DATA 32,228,255,240,251,
   201,89,240,5,201,78,208,243,
   96,166,253,2786
73 700 DATA 160,5,189,32,207,15
   3,48,207,166,251,165,252,134
   ,99,133,98,2299
2F 710 DATA 162,144,56,32,73,18
   8,32,223,189,32,135,180,32,1
   66,182,162,1988
D9 720 DATA 0,189,0,1,157,0,2,2
   40,3,232,208,245,160,0,185,4
   8,1670
03 730 DATA 207,157,0,2,240,4,2
   32,200,208,244,173,2,3,141,2
   32,207,2252
76 740 DATA 173,3,3,141,237,207
   ,169,231,141,2,3,169,207,141
   ,3,3,1833
69 750 DATA 162,255,160,1,76,13
   4,164,169,131,141,2,3,169,16
   4,141,3,1875
02 760 DATA 3,76,116,164,64,93,
   2,93,66,29,2,21,74,21,64,95,
   983
```

Printfx Loader