Increase the speed of your BASIC programs by 10 to 45 percent with

FASTER 64

By Anthony Wood

Faster 64 is a utility for the Commodore 64 that analyzes the use of variables in a BASIC program while it runs. It tells you how often each variable in the program is accessed. By defining your most often used variables first, BASIC does not have to search as far for them, thus speeding up your program. For some programs, this speed increase can be considerable.

Faster 64 is a machine language program. Since it is slow and inconvenient to use a BASIC program to POKE in a machine language program, we have created a special ML listing of *Faster 64.* To enter *Faster 64,* refer to the *Flankspeed* instructions on page 94. Once entered and SAVEd you need only enter 'LOAD "FASTER 64",8,1' (tape users, 'LOAD "FASTER 64",1,1') when you wish to load the program. To initialize it, you also have to type 'SYS 49152'. This should be done after you first LOAD it or after you press RUN STOP/RESTORE.

After you key in the enclosed program, *Faster 64,* you should save it before you run it. Once you have saved it, LOAD it and type SYS 49152, and you should see the message "*Faster 64* working."

USING *FASTER 64*

Load *Faster 64* and initialize it. When you see the message "*Faster 64* working," enter the following line:

```
Q=0:A=0:A(1)=A(2)+A(3):Z$="FRED"
<RETURN>
```

The following should appear:

```
A() 3 ,Q 1, A 1,Z$ 1
```

This means the array A was referenced 3 times and the variables Q, A, and Z$ were each referenced 1 time. The variables referenced the most are listed first.

Key in this short program to test *Faster 64* some more:

```
NEW
10 DIM A(20)
20 FOR Y=1 TO 20
30 A(Y)=A(Y)+1
40 NEXT
```

RUN it. You should get the message:

```
Y 41, A() 40
```

This means that the array A( ) was referenced 40 times, and the variable Y 41 times. Notice that a FOR-NEXT loop only references its index once. This is because the FOR-NEXT loop stores the address of its index variable. It does not have to keep looking it up. It will, however, look up the index in every loop if you enter 'NEXT Y' instead of just 'NEXT.'

You should be aware that the variable TI will not work with *Faster 64*; it causes a syntax error.

To use *Faster 64* on one of your programs, load *Faster 64* and initialize it. Note that you might have to enter "NEW" after you load *Faster 64,* to prevent the "out of memory" error. This is a bug in the Commodore BASIC ROM. Now load your program. Run your program all the way through. After your program is finished, its variables will be listed in numerical order. Suppose that you run your program, and you get the following display from *Faster 64:*

```
X 2131, Z 511, P() 200, F()154, X$ 100,
D 2
```

To initialize the variables in the correct order, you would enter a line at the beginning of your program like this:

```
1 DIM P(100),F(100)
2 X=0:Z=0:X$="":D=0
```

This puts your variables in the most efficient order. Notice that the arrays are on a separate line.

You should look out for certain exceptions. For example, suppose you find out that the variable A$ is referenced 4000 times. It might not be best to define it first, if it is not at a place in your program where speed is important. For example, suppose A$ appears in this line:

```
1000 GET A$:IFA$<>CHR$(13)THEN1000
```

You can see that A$ is in a loop waiting for a return. Since it is used in a wait loop, you can define A$ last because speed is not important—defining it first would just slow down the search for more critical variables. □

SEE PROGRAM LISTING ON PAGE 106

To enter Faster 64 and BASIC Trace...

you must use our Flankspeed program on page 94. (The BASIC loader for BASIC Trace should be entered in the usual manner, prescribed on pages 93 and 94.)

```
IM ·50 RETURN
PI ·51 GOSUB48
   ·52 IFERTHENPRINTTAB(8)ER;ER$;T;S:PRINT"[
    RVSON][RED][s P][s L][s E][s A][s S][s E
    ] [s P][s O][s W][s E][s R] [s D][s O][s
     W][s N][SS]& [s S][s T][s A][s R][s T]
    [s O][s V][s E][s R]!";:GOTO52
JN ·53 CLOSE15:OPEN15,DV,15
KN ·54 PRINT#15,"NO:"+DN$+","+ID$
AB ·55 CLOSE15
AB ·56 GOSUB48
PI ·57 IFERTHEN52
DG ·58 CLOSE2:OPEN2,DV,1,"0:[SS]THE MAGIC BO
    OT,P,W":REM SHIFTED SPACE BEFORE TITLE
JO ·59 PRINT#2,CHR$(S%-INT(S%/256)*256);
NI ·60 PRINT#2,CHR$(S%/256);
BF ·61 FORI=0TOCS%-1
AD ·62 PRINT#2,CHR$(PEEK(S%+I));
OH ·63 NEXT
IA ·64 CLOSE2
NC ·65 RETURN
IM ·66 PRINT"[CLEAR]";CHR$(14):POKE36879,25
KO ·67 PRINTTAB(7)"[3"[DOWN]"][BLUE][s V][s
    I][s C] [s B][s O][s O][s T][s E][s R][D
    OWN]"
LN ·68 PRINT" ([s C]) 1984 [s G]EORGE [s J]O
    NES"
OI ·69 FORT=1TO2000:NEXT
OB ·70 RETURN
IM ·71 FORC=1TOLEN(PN$)
BK ·72 POKESR%,(PEEK(FI%))
NL ·73 SR%=SR%+1:FI%=FI%+1
IB ·74 NEXT
IA ·75 POKELM%,LEN(PN$):REM SET LENGTH OF NE
    W PROGRAM NAME
IK ·76 RETURN
IM ·77 REM*******************************
KF ·78 REM  MX=65535 MAXIMUM ADDRESS
IB ·79 REM  BS%=692 START OF NUBOOT
EN ·80 REM  FI%=512 SYSTEM INPUT BUFFER
OG ·81 REM  CS%=80 NO OF BYTES TO SAVE
BK ·82 REM  SR%=735 LOC OF FILE NAME
AI ·83 REM  RN%=733 INTERPRETER LOOP &
GC ·84 REM  SYS ADDRESS FOR ML PROG
BI ·85 REM  LM%=702 STORE LEN(PN$)
IL ·86 REM  IM%=770 BASIC WARM START VECTOR
PC ·87 REM  PN$= PROGRAM NAME
BJ ·88 REM  DN$= DISK NAME
MM ·89 REM  ID$= DISK ID
CC ·90 REM*******************************
GH ·91 DATA40,67,40,49,57,56,52,71,74,79,78,
    69,83,169,1,162
ML ·92 DATA8,160,1,32,186,255,169,15,162,223
    ,160,2,32,189,255,169
CO ·93 DATA0,162,255,160,255,32,213,255,134,
    45,132,46,32,239,2,32
AO ·94 DATA91,228,32,89,198,76,174,199,160,1
    60,160,160,160,160,160,160,160
GI ·95 DATA160,160,160,160,160,160,160,160,1
    62,0,169,160,157,223,2,232
OL ·96 DATA224,16,208,246,141,189,2,96,0,139
    ,227,180,2
JB ·97 REM 'VIC BOOTER' COPYRIGHT 1984 GEORG
    E JONES * FOR VIC 20
CD
```

FASTER64

FROM PAGE 39

First byte: C000 Last byte: C242

```
C000: 78 A9 00 85 FC A2 A0 86 6E
C008: FD A0 00 A2 00 9D 43 C2 EC
C010: 9D 43 C3 9D 43 C4 9D 43 3B
C018: C5 9D 43 C6 E8 D0 EE B1 DF
C020: FC 91 FC E6 FC D0 02 E6 49
C028: FD A5 FD C9 C0 D0 F0 A5 BB
C030: 01 29 FE 85 01 58 A2 4C 27
C038: 8E 8B B0 A2 79 8E 8C B0 EA
C040: A2 C0 8E 8D B0 A2 CA 8E 6C
C048: 55 A8 A2 C0 8E 56 A8 A2 D9
C050: F0 8E E5 B0 A2 C1 8E E6 40
C058: B0 A2 4C 8E E3 B1 A2 F8 B7
C060: 8E E4 B1 A2 C1 8E E5 B1 10
C068: A0 00 B9 0D C2 C9 00 D0 2D
C070: 01 60 20 D2 FF C8 4C 6A 44
C078: C0 A2 00 8E 07 C2 20 79 CD
C080: 00 20 90 B0 48 98 48 A2 AD
C088: 00 BD 43 C2 F0 25 BD 43 63
C090: C2 C5 45 D0 1A BD 43 C3 0E
C098: C5 46 D0 13 BD 43 C6 CD 1E
C0A0: 07 C2 D0 0B FE 43 C4 D0 1E
C0A8: 1D FE 43 C5 4C C6 C0 E8 8A
C0B0: 4C 89 C0 A5 45 9D 43 C2 D5
C0B8: A5 46 9D 43 C3 FE 43 C4 50
C0C0: AD 07 C2 9D 43 C6 68 A8 F0
C0C8: 68 60 A9 0D 20 D2 FF A2 DD
C0D0: 00 8E 07 C2 A0 00 B9 44 C7
C0D8: C2 F0 7A B9 44 C5 D9 43 E7
C0E0: C5 F0 05 B0 0D 4C 51 C1 B9
C0E8: B9 44 C4 D9 43 C4 90 61 7F
C0F0: F0 5F A9 01 8D 07 C2 B9 FC
C0F8: 43 C2 8D 08 C2 B9 43 C3 18
C100: 8D 09 C2 B9 43 C4 8D 0A B2
C108: C2 B9 43 C5 8D 0B C2 B9 A2
C110: 43 C6 8D 0C C2 B9 44 C2 37
C118: 99 43 C2 B9 44 C3 99 43 56
C120: C3 B9 44 C4 99 43 C4 B9 02
C128: 44 C5 99 43 C5 B9 44 C6 99
C130: 99 43 C6 AD 08 C2 99 44 2A
C138: C2 AD 09 C2 99 44 C3 AD C3
C140: 0A C2 99 44 C4 AD 0B C2 2B
C148: 99 44 C5 AD 0C C2 99 44 46
C150: C6 C8 4C D6 C0 AE 07 C2 3C
```

```
C158: F0 03 4C CF C0 A0 00 8C 56
C160: 06 C2 38 20 F0 FF 98 AC B7
C168: 06 C2 C9 1E 90 05 A9 0D 65
C170: 20 D2 FF B9 43 C2 F0 5A 6E
C178: 29 7F 20 D2 FF B9 43 C3 D4
C180: 29 7F 20 D2 FF B9 43 C3 DC
C188: 29 80 F0 14 B9 43 C2 29 20
C190: 80 F0 08 A9 25 20 D2 FF CB
C198: 4C A0 C1 A9 24 20 D2 FF 08
C1A0: B9 43 C6 F0 0A A9 28 20 51
C1A8: D2 FF A9 29 20 D2 FF A9 EA
C1B0: 20 20 D2 FF B9 43 C4 AA 30
C1B8: B9 43 C5 8C 06 C2 20 CD BE
C1C0: BD AC 06 C2 A9 20 20 D2 B0
C1C8: FF A9 2C 20 D2 FF C8 4C A6
C1D0: 5F C1 A9 14 20 D2 FF A9 4C
C1D8: 00 A2 00 9D 43 C2 9D 43 FF
C1E0: C3 9D 43 C4 9D 43 C5 9D 8E
C1E8: 43 C6 E8 D0 EE 4C 86 E3 52
C1F0: A9 01 8D 07 C2 4C D1 B1 C2
C1F8: AD 07 C2 48 20 B2 B1 68 A5
C200: 8D 07 C2 4C E6 B1 00 00 3C
C208: 00 00 00 00 00 0D 46 41 9C
C210: 53 54 45 52 20 36 34 20 F9
C218: 57 4F 52 4B 49 4E 47 2E 69
C220: 0D 42 59 20 41 4E 54 48 15
C228: 4F 4E 59 20 57 4F 4F 44 79
C230: 0D 48 4F 55 53 54 4F 4E 6F
C238: 2C 20 54 58 20 31 39 38 F3
C240: 34 0D 00 81
```

BASIC Trace

FROM PAGE 57

BASIC LOADER

```
·15 IFPEEK(49152)=169THENSYS49152:END                    FJ
·20 PRINT"[CLEAR][WHITE][RVSON] BASIC TRA
 CE [RVSOFF]":PRINT"[DOWN]TO BEGIN TRACIN
 G YOUR BASIC PROGRAM,"                                  LO
·25 PRINT"TYPE:[3" "]TRACE <RETURN>"                     KO
·30 PRINT"THE COMPUTER WILL PRINT [RVSON]
  ON [RVSOFF], AND THE"                                  CE
·35 PRINT"TRACE WINDOW WILL APPEAR.  NOW
 RUN YOUR"                                               IG
·40 PRINT"BASIC PROGRAM.  TO SPEED UP THE
  TRACING"                                               JM
·45 PRINT"HOLD DOWN THE [RVSON] SHIFT [RV
 SOFF] KEY."                                             MI
·50 PRINT"WHEN YOU NO LONGER WANT TRACE,
 TYPE:"                                                  CI
·55 PRINT"TRACE <RETURN>. THE COMPUTER WI
 LL PRINT"                                               MC
·60 PRINT"[RVSON] OFF [RVSOFF], MEANING T
 RACE IS NOW OFF."                                       IF
·65 PRINT"TO RESTART THE TRACE, TYPE:                    AD
·70 PRINT"TRACE <RETURN> AGAIN."                         CO
·101 PRINT"[3"[DOWN]"]ARE YOU USING TAPE
 OR DISK[3" "](T/D)?"                                    GL
·105 GETK$:ON -(K$="")GOTO105:D=1:IFK$="D
 "THEND=8                                                JC
·110 PRINT"[DOWN][DOWN]PLEASE WAIT WHILE
 I LOAD THE ML.":LOAD"BTML",D,1                          EM
```

ML LISTING

First byte: C000 Last byte: C1F8 SYS to start: RUN

```
C000: A9 4C 85 7C A9 80 85 7D 25
C008: A9 C1 85 7E 60 60 60 60 F8
C010: 8E FE C1 8C FC C1 A9 00 54
C018: 2A 8D FB C1 60 00 00 00 ED
C020: A2 23 A0 06 8C 42 C1 A0 BD
C028: 05 A9 01 9D 00 D8 E8 88 BF
C030: D0 F9 8A 18 69 23 AA CE A3
C038: 42 C1 D0 EB F0 07 FD 43 32
C040: C1 CA 18 90 43 A2 02 A5 03
C048: 39 8D 3E C1 CD 3F C1 D0 AE
C050: 01 CA A4 3A C0 FF D0 02 8E
C058: A0 00 8C 40 C1 CC 41 C1 57
C060: D0 03 CA F0 78 8D 3F C1 F6
C068: 8C 41 C1 A2 00 A0 00 8C C7
C070: 42 C1 AD 3E C1 38 FD 43 9B
C078: C1 A8 E8 AD 40 C1 30 BE 6A
C080: FD 43 C1 CA 0A B0 0C 6A 7F
C088: 8D 40 C1 8C 3E C1 EE 42 D5
C090: C1 90 DF 8A 4A AA AD 42 32
C098: C1 9D 4D C1 8A 0A AA E8 2F
C0A0: E8 E0 0A D0 C8 A9 00 8D 45
C0A8: 3D C1 AD 8D 02 C9 01 F0 A0
C0B0: 0F A9 05 8D 3D C1 A2 00 9D
C0B8: A0 00 C8 D0 FD E8 D0 F8 A3
C0C0: A2 00 BD 67 C1 9D 62 C1 0C
C0C8: E8 E0 14 D0 F5 A2 00 BD CD
C0D0: 4D C1 18 69 B0 9D 76 C1 E7
C0D8: E8 E0 05 D0 F2 A9 00 8D A2
C0E0: 5C C1 AE 3D C1 8E 5D C1 5A
C0E8: AE 5D C1 BD 52 C1 AE 5C 93
C0F0: C1 9D 23 04 EE 5C C1 EE 73
C0F8: 5D C1 AD 5C C1 C9 05 D0 83
C100: E7 A9 05 8D 3E C1 A2 23 E9
C108: 8E 40 C1 A2 00 8E 42 C1 CD
C110: A0 05 AE 42 C1 BD 62 C1 4A
C118: AE 40 C1 9D 28 04 EE 40 C1
C120: C1 EE 42 C1 88 D0 EB AD C7
C128: 40 C1 18 69 23 8D 40 C1 5E
C130: CE 3E C1 D0 DB 60 60 60 CC
C138: 00 00 00 20 C0 05 00 F5 14
C140: EB 00 19 10 27 E8 03 64 CC
C148: 00 0A 00 01 00 00 00 02 55
C150: 04 05 BC 92 95 8E BE 90 1C
C158: 81 95 93 85 05 0A 00 00 97
C160: 00 00 B1 B0 B0 B1 B8 B1 8F
C168: B0 B0 B1 B9 B1 B0 B0 B2 FA
C170: B0 B1 B0 B0 B2 B1 B0 B0 F9
C178: B2 B4 B5 00 00 00 00 00 95
C180: 8D FF C1 20 10 C0 AD FD 6C
C188: C1 F0 03 20 20 C0 A2 00 E1
C190: BD 00 02 F0 08 DD EA C1 D3
C198: D0 07 E8 D0 F3 E0 00 D0 CF
C1A0: 0E 20 F0 C1 AD FF C1 C9 BA
```

COMMODORE USERS

48355
K

Ahoy!

$2.50/CAN.$3.00 APRIL 1985

Hop around 84
TRACE 57

...GEAR UP TO MACHINE LANGUAGE!..

NEW COMMODORE COMPUTERS
128PC: EXPANDS TO 512K!
LCD: 80 COLUMNS TO GO!

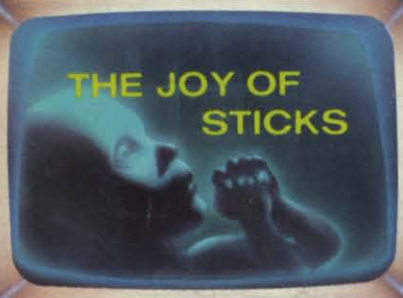
ASSEMBLY CLASS IS IN SESSION!
COMMODORE ROOTS
NEW MACHINE LANGUAGE COLUMN

ELECHECK
C-64 MONEY MANAGEMENT
BOOTER
FOR THE VIC AND 64
SPACE HUNT
A C-64 STAR SEARCH

GET ON THE FAST TRACK!
1541 IMPROVEMENTS
- INDUS, COMMANDER DRIVES
- ON TRACK INDICATOR
- 1541 PHYSICAL

GETTING INTO THE KERNAL
ACCESSING ML ROUTINES FROM BASIC

PROTECTO ENTERPRIZES
COMMODORE MAIL ORDER SECTION
PAGES 64-81



ML PROGRAMS
FASTER 64
BOOST BASIC BY 45%!
BASIC TRACE
DEBUGGING DEMYSTIFIED