DECUS

PROGRAM LIBRARY

DECUS NO.	8-553
TITLE	BIG BROTHER II
AUTHOR	Gary Garber
COMPANY	Lawrence High School Trenton, New Jersey
DATE	June 30, 1972
SOURCE LANGUAGE	PAL III

BIG BROTHER II

DECUS Program Library Write-up DECUS NO. 8-553

INTRODUCTION

Big Brother II is a superset of PAL III. Although the language of Big Brother is not as versatile, the assembly time to actual running is almost instantaneous. Whereas you need three passes in PAL then loading of the binary tape etc. Big Brother II stores the user program then when the command is given the program is assenbled,checked for errors, put into a corrosponding section of core then automatically run. Upon completion of the program control is then transferred back to Big Brother. You can then either edit,debug or get an ASCII or Binary tape of your program. This is between ten and twenty times faster than the other assemblers.

In order for Big Brother to be run your computer must have at least 8K of core, an ASR-33 teleprinter-reader, and a Memory Quotient Register (MQ).

NOTE: When writing programs in Big Brother II follow the formatting strictly or else the program may destroy itself!

LOADING

I. Introduction

Big Brother does not need the binary loader because it carries its own loader and is self-starting being similar to the Edusystem 20 BASIC.

The first thing to go over is the RIM. This loader must be in, and in the correct section of core. If your computer has a Bootstrap loader like the one I work on, then you have nothing to worry about. Otherwise here are the RIM Loaders for either low or high speed input;

	LOW SPEED	HIGH SPEED
7756/	6032	6014
7757/	6031	6011
7760/	5357	5357
7761/	6036	6016
7762/	7106	7106
7763/	7006	7006
7764/	7510	7510
7765/	5357	5374
7766/	7006	7006
7767/	6031	6011
7770/	5367	5367
7771/	6036	6016
7772/	7420	7420
7773/	3776	3776
7774/	3376	3376
7775/	5356	5357

II. Loading without the Bootstrap loader

Once the RIM is loaded put in the Big Brother tape in one of the readers. Then put 7756 in the switch register, press load address. Ther place 0000 in the switch register, press clear then press continue. The tape will then automatically load through its own binary loader and will automatically stop at the checksum trailer and then print out "READY" on the teletype. If this message isn't printed out then Big Brother is not correctly loaded and should be reloaded.

III. Loading with the Bootstrap loader

Put 0000 in the switch register and press load address, then clear. After that put the SW. down and up. This will load the RIM then Big Brother and print "READY" automatically if correctly loaded. If not reload the tape again.

LANGUAGE OF BIG BROTHER II

The language of Big Brother II is, in some ways, similar to that of PAL III, but there are significant changes. In the following sections certian symbols will be used to denote certian things.

SPACE= _ TAB= ... CR/LF=↓↑

I. TAGS

A. In Big Brother II tags are signified in a different way than in PAL. PAL uses a comma

TAG,___HLT

Big Brother uses an exclamation mark,

TAG!HLT

There are essentially three different types of tags in Big Brother.

1) An instruction after a tag. In this case the instruction is placed directly next to the exclamation point.

TAG!TAD A

2) A numeric value after the tag. In this case when the tag signifies a storage space there is a space left after the exclamation point, then some number.

TAG!_nnnn

a. It is possible to negate a number by putting a minus sign in place of the space.

TAG!-nnnn

3) Another symbol is in the storage space. This type of tag is used as a transfer space;

```
      JMP I TAG
TAG,AREA
AREA,HLT
```

In Big Brother this is coded in the following manner.

```
      JMP I TAG
TAG!'AREA'
AREA!HLT
```

Note that the symbol is enclosed in apostrophes.

You can also do this;

TAG!'AREA'±nnnn (the number is not enclosed in the apostrophes)

In this case you will jump to AREA±nnnn

B. The length of a tag is unlimited but it may not be started with a number and it may not contain a space within it.

C. In order to just express a number without a tag like,

```
200/TAD 201
201/7200
```

You would express 7200 like this;

```
200/TAD 201
201/!_7200
```

No letters in front of the exclamation mark, signify that there is some constant to be put into the program.

II. Address Initialization

In PAL III, when you want to designate a new section of your program is going into a new section of core,. you use,

*nnnn

In Big Brother you use this format,

.....*↓↑
.....nnnn↓↑

Only in this part must you have a tab function in front of the instruction.(explained later)

III. TAB Function

In PAL the TAB is signified by Ctrl/Tab,. which gives eight spaces.. In Big Brother I use Ctrl/Z. As stated above you need this function only in front of the address initialization, otherwise it optional. But it is advised that it be used for the sake of neatness.

IV. SEMI-COLON

In Big Brother if you care to write comments or key words that tell you about the operation of your program you just substitute semi-colons in the place of the spaces.

THIS;IS;THE;STARTING;ADDRESS;↓↑
*↓↑
200↓↑
.
.
.

The semi-colon just tells Big Brother to ignore the previous word typed in.

V. End of the Program

This is the same as in PAL II. You just put a dollar sign ($) after the last line. NOTE: There must be a dollar sign or else there might be a fatal error when either editing or assembling

THE MODES OF BIG BROTHER II

I. Neutral Mode

This mode is the waiting mode. Whenever it is entered READY is printed. This signifys that any one of the five other modes are now ready to be entered. The modes are:

1. INPUT
2. RUN
3. EDIT
4. DEBUG
5. TAPE

II. Input Mode

In order to enter it from the neutral mode press Ctrl/Tab (Ctrl/I). The computer will then go into a holding loop so that you can get the switch register set to the reader options. They are:

BIT#	0	1
KEYBOARD	0	0
H.S.R.	1	0
L.S.R.	0	1

After the Switch register has been set, press "G" on the keyboard and you are now in the input mode. If reading from one of the readers, after "G" has been pressed your tape will be read in.

NOTE: If typing a tape off line,be sure, if the tab function is used(eight spaces) to put Ctrl/Z in front of the spaces. Because when Big Brother II reads a Ctrl/Z it ignores successive spaces and just stores the Ctrl/Z to save core space.

If a mistake was made in typing in the input mode just press the RUBOUT key, once for each letter rubbed out. A rubout appears as a bracket (⌋). If a CR/LF needs to be rubbed out, only one rubout is needed because it is stored as a zero in the computer.

To exit the input mode, press Ctrl/D.. The computer will then return with a CR/LF then a /=nnnn. This is the total line count in the user program. Then another CR/LF is given and entrance into the Neutral mode and "READY" is printed. Your program has now been stored in core and is ready to be worked on.

III. RUN MODE

This is the biggest advantage of Big Brother over PAL III. To enter this mode press Ctrl/Form(Ctrl/L). No CR/LF is given. In this section Big Brother takes the symbolic program, checks it for errors,assembles it, and runs it.

When the user program is being run, it is run in the locations specified by the user, but, it is run in the upper 4K of core, so the Extended Memory Address light will be on. When the user's program halts, transfer is

then brought back to Big Brother II. At that point the acumulator and link are printed out and you are put in the debugging mode, to examine core,etc.

There are three types of errors that can be encountered;

1. DUPLICATE TAG
2. UNDEFINED ADDRESS
3. ILLEGAL REFFERENCE

A. Duplicate Tag

This takes the form of;

DUPLICATE TAG XXXX AT NNNN WITH XXXX AT JJJJ

Where NNNN and JJJJ are the locations of the duplicate tags.

The numbers XXXX are the numeric value of the tags. In Big Brother II. Thwn a tag or any name is encountered the letters are summed up and then that value along with its location are stored in the user symbol table. Therefore,

204/TAG!ISZ NAME

would cause Big Brother to sum up the letters in TAG which equals 1134 then 204 which is the address at which the tag occured would be stored next to it. (When summing up letters the ASCII code is used)

NOTE: A duplicate tag can occur with two tags that look totally different, like

204/HOLD!-7

205/SANE!+260

would cause a duplicate tag message of

DUPLICATE TAG 1447 AT 204 WITH 1447 AT 205

becaues HOLD and SANE both add up to 1447.

B. Undefined address

This takes the form of,

UNDEFINED ADDRESS XXXX AT NNNN

Where XXXX is the sum of the symbolic tag refered to and NNNN is its address.

When Big Brother II hits one of these, it puts a 5000 in the space of the instruction. If there are NNNN references to one undefined address, then NNNN messages will be printed out because Big Brother checks each line for its validity.

C. Illegal Reference

This takes the form of,

ILLEGAL REFERENCE LLLL AT NNNN

Where LLLL is the addressed location and NNNN is the location where the instruction is.

When this in encountered Big Brother II still tries to code the instruction.

In all cases of an error, the program is not run, but when all errors have been found, control is returned to Big Brother II and the neutral mode is entered so the user can go and repair his mistakes immediatly by going to the EDIT mode.(explained later)

When assembling a HLT Big Brother replaces the 7402(numeric coding of a HLT), with a 5000. Because this is a JMP 0000. In locations 0-2 in upper-4K

is a transfer area so we can get between upper and lower sections of core.

IV. EDIT MODE

In order to enter the Editor from neutral mode, press the ALT MODE key. The bell will ring and a CR/LF will be given, you are then in the Edit mode. There are quite a few options in this section which allow the user to edit his program.. They are;

1) List
2) Delete
3) Change
4) Search
5) Insert
6) Apend
7) Punch

NOTE: All numbers refered to in this section and all subsequent sections are in base eight.(Octal)

A. List

This takes the form of

L_NNNN↓↑

Where NNNN need not be four digits long. It is very inportant that there be a space after the L. When the number has been input,press return and the line will then be printed out. If you wish that the whole program is to be printed out, just put

LA↓↑

and the editor will automatically print out the whole program. In you wish to list the next line or the previous line use these two instructions,

\+ this will list the next sequential line.

\- this will list the previous line.

If you wish to know what line you are on at any point or to know the last line in the program, two more instructions are used. They are,

\# this will list the numeric value of the last line in the user program(octal).

& this will list the numeric value of the current line that the editor is on(octal).

B. Delete

This takes the form of,

DNNNN↓↑

Where NNNN is an octal number of any length.

This instruction will delete the line NNNN from the program and after it has been executed a CR/LF will be given to signify its completion.

C. Change

This instruction takes the form,

CNNNN↓↑

Where NNNN is an octal number of any length. When return is pressed another CR/LF is given. You are now ready to type in the corrected line. When you are finished, press return and the new line will be substituted in the place of the old line.

D. Search

This takes the form of,

SNNNN↵

Where NNNN is an octal number of any length. Once the CR/LF is given the editor waits for the search character. When it is given, the line will be printed out until and including the character. You then can add on or rub out all previous characters.

To continue the search for the same character press Ctrl/S. Then the line will be printed out until the character or until the line is finished.

To change the search character, press Ctrl/C. You now can press a new search character and repeat the above process.

NOTE: If you want to delete a part of a line up to a certian point, search for the end of that clause then rubout the characters until the desired letter.

Once the desired corrections have been made, in

order to leave the rest of the line, don't press CR/LF because that will be accepted as part of the corrected line. To exit and leave the rest of the line you press Ctrl/S. This will continue the search and eventually will hit the end of the line and when this occurs you will automatically end the search and return to the main editor.

E. Insert

This instruction takes the form of,

INNNN↓↑

Where NNNN is an octal number of any length. The lines inserted will be before NNNN. Any number of lines can be inserted before NNNN and when you wish to exit the insert routine, press Ctrl/D and you will return to the main editor..

F. Apend

This instruction takes the form of,

A

Note that no CR/LF is needed.. This allows you to add on any number of lines on to the end of your program. After the "A" has been pressed, just start typing in the new line. When enough lines have been added press Ctrl/D to exit the Apend routine to the main editor.

G. Punch

This is really a combination of two subroutines.

1) Tape

2) Punch

First you press T for tape. The computer will then punch out a section of blank leader and then stop. Turn the punch off. Then press P for punch. The computer will then wait for you to turn the punch on. Then press G for go. This tells the computer to print out the users program from core, then halt after a $ has been printed. when the program has been printed out repeat the first step to get a section of trailer.

To exit from the editor mode press Ctrl/D. This will put you into the neutral mode and "READY" will be printed.

V. DEBUGGING MODE

To enter this mode you press Ctrl/BELL (Ctrl/G). The bell will ring and a CR/LF will be given.

This mode allows you to examine any section of core or change any section of core or run your program under the control of Big Brother II.

Listing a section of core takes the format of,

>NNNN↓↑

Where NNNN is an octal digit of any length. This will list out the octal contents of the location NNNN.

To change the contents of a location you use the instruction with the following format,

<NNNN↓↑

XXXX↓↑

Where NNNN is an octal digit of any length. This is the location to be changed. XXXX is also an octal digit of any length and is the new contents of that location.

If the user wishes to run the program in core, up to a certian point, and then examine core and the registers for debugging a program, a breakpoint facility was included.

To insert a breakpoint use this instruction,

←NNNN↓↑

Where NNNN is an octal digit of any length.

To remove a breakpoint that has been inserted use this instruction,

↑ (shift N)

Note that no CR/LF is needed.

To run your program when a breakpoint is inserted or at any time at all use this instruction,

RNNNN↵

Where NNNN is an octal digit of any length. This will transfer control over to the users program until either a breakpoint or a halt has been hit. When one of these is encountered control is then transfered back to the debugging mode and the contents of the acumulator and link are printed out.

To exit the debugging mode press Ctrl/D. You will then return to the neutral mode and "READY" is printed out.

VI. TAPE MODE

To enter this mode press Ctrl/TAPE (Ctrl/R). No CR/LF will be given. This section consists of two instructions that enable the printing of a binary tape of the users program.

First press L for leader. The computer will wait for you to put the punch on, then press G for go, and a code 200 leader will be printed out. After that is done put the punch off. Then you are ready to punch out as many sections of core as needed. It takes the format of,

PNNNN↓↑

,XXXX↓↑

Where NNNN and XXXX are octal digits of any length. NNNN is the starting address of the block and XXXX is the ending address of the block. Note that there is a comma before XXXX, this is necessary to have. After the return key is pressed after XXXX the computer will again wait for you to put the punch on. Then press G and a binary tape will be punched out for that section of core. when it is finished turn the punch off. If more sections are to be printed out, repeat this previous step. Otherwise repeat the first step to obtain a trailer.

To exit this mode, as in the others, press Ctrl/D and enter the neutral mode.

ILLEGAL PROCEDURES AND RESULTS OF ILLEGAL FORMATTING

I. Once a command has been given such as a list or punch, or

any instruction,you are committed to that instruction. The reason for this is the fact that once the letter has been read in, Big Brother automatically evaluates its validity and jumps to a subroutine to take care of its execution.

II. There are quite a few things that will destroy Big Brother II. They are all in the users program, and cause illegal references to occur when assembling is being carried out.

A. If there is incorrect formatting in the initialization of the address (*↓↑....XXXX↓↑).

B. Incorrect formatting of tags.

C. Misspelled words

D. No dollar sign at the end of the program.

III. These next srrors will not destroy Big Brother II but may destroy the users program, or cause unnecessary print-out.

A. Calling for some operation on a line that is non-existant in the users program.(such as listing line 30 in a 10 line program)

B. Again no dollar sign.

IV. <u>Restarting Big Brother II after an error.</u>

If the program starts to print wildly after an editor command is given, just halt the computer. Put 2200 in the switch register, press load address, clear and then continue. The program will then print out "READY". At this time go to the editor and list out

your program to see that it is still there and unharmed.

If there was an error when trying to assemble your program, reload Big Brother again, because an error like that is fatal to the program..

DISTRIBUTION OF USER SPACE

I. When writing a program in Big Brother II, the stored program starts at 2000 and goes to 7777 in upper 4K. Therefore the reccomended space for the operable ares of the program is from locations 3 to 1777. Spaces 0 to 2 are used as transfer areas between upper and lower 4K of core.

If the program goes up beyond 2000 there will be a conflict and possibly there will be an irretrievealbe error. Also the stored brogram will be destroyed. But if the user is daring enough, I suggest that he get an ASCII tape of the program first,so that if an error occurs, he can read in the tape again.

Starting on the next page is a map of the distribution of core space in Big Brother II and where the subroutines are located.

DISTRIBUTION OF CORE

LOCATION	SUBROUTINE
0-177	Utility transfer spaces, storage and print routine.
200-577	Main section of the assembler
600-1177	Formulating of operate and I/O instructions and transfer areas for the basic instructions
1200-1377	Routine to formulate symbol table and to search through it for tags
1400-1417	Continuation of search routine for symbol table
1420-1577	Section two of transfer ares to formulate instructions
1600-1743	Binary punch
2000-2025	Formulation of combined instructions
2026-2111	Formulation of symbolic tags
2200-2266	Routine for neutral mode
2300-2343	Subroutine for Jump + or - in assembler
2345-2357	Prints out /=NNNN after input mode
2400-2443, 2476-2534	List routine
2444-2475	Routines to get a character from upper 4K and number input routine
2600-2711	Insert routine
2712-2756	Routine to move characters in users program, back NNNN times.
3000-3157	Change routine

3200-3347	Search routine
3400-3577	Debugging routine
3600-3772	Input mode routine
4000-4164	Editor main routine and assorted subroutines
4200-4377	Routine to check for errors and print them out
4400-4534	Error messages
4600-4650	Delete routine
5000-5046	Apend routine
5100-5177	Buffer area for editor
5200-7700	Buffer area for symbol table of user
7702-7757	Self starter
7757-7777	RIM loader

UPPER 4K

0-2	Transfer routine between upper and lower 4K of core
3-1777	Reccomended area opened* for running of user program*
2000-7777	Area to store the user ASCII program*

* Open to modification if necessary.

SUMMARY OF THE INSTRUCTIONS IN BIG BROTHER II

I. Loading

A. Put RIM in core, and tape in reader

B. SW up or load 7756 and start rim.

C. Tape will automatically load and start itself.

II. Modes

A. Debugging mode; Ctrl/BELL (Ctrl/G)

B. Input mode; Ctrl/TAB (Ctrl/I)

C. Run mode; Ctrl/FORM (Ctrl/L)

D. Binary Mode; Ctrl/TAPE (Ctrl/R)

E. Editing Mode; ATL MODE

F. Neutral Mode; Ctrl/D

III. Debugging Mode

>NNNN↓↑	Print contents of location NNNN
<NNNN↓↑	Change core space NNNN
←NNNN	Insert a breakpoint at NNNN (shift O)
↑	Remove breakpoint (shift N)
RNNNN↓↑	Recycle program from NNNN to breakpoint or halt

IV. Input Mode

S.R. OPTIONS	BIT#	0	1
Keyboard		0	0
High Speed Reader		1	0
Low Speed Reader		0	1

G	Allows you to enter the program from either of the readers or keyboard
Ctrl/Z	Tab function for eight spaces

V. Run Mode

This will execute the user program totally, from assembly, checking for errors, running it to passing control back to Big Brother II.

NOTE; When control is transferred back to Big Brother the contents of the AC. and Link are printed out and you are put in the Debugging mode.

VI. Tape Mode

L	Give leader after G is printed
PNNNN↓↑ ,XXXX↓↑	Punch out a section of core as a binary tape after G is printed
G	Command that gives the user time to put to put the punch on.

VII. Editor Mode

L_NNNN↓↑	List line NNNN
LA	List the whole program
DNNNN↓↑	Delete line NNNN
INNNN↓↑	Insert any amount of lines before NNNN until Ctrl/D is pressed
CNNNN↓↑	Change line NNNN
SNNNN↓↑	Search line NNNN for search character input.
Ctrl/C	Change search character
Ctrl/S	Continue search for character
A	Apend, add on to the end of your program

T	Punch a section of leader or trailer
P	Punch out program in ASCII after G is printed.
#	List numeric value of last line
&	List numeric value of current line
+	List next sequential line
-	List previous line

VIII. Neutral Mode

Ctrl/D	Exits any mode except run mode and puts you into this mode which is a holding pattern from where you go to any other mode.
"READY"	Tells the user that the neutral mode is ready to accept instructions.

NOTE: All numbers refered to in these sections are in octal.

BIG BROTHER II SYMBOL TABLE

Basic Instructions

AND

TAD

ISZ

DCA

JMS

JMP

Group 1 Micrainstructions

NOOP

CLA

CLL

CMA

CML

RAR

RAL

RTR

RTL

INC

BSW

Group 2 Microinstructions

SMA

SZA

SPA

SNA

SNL

SZL

SKIP

OSR

HLT

CLA

Combined Microinstructions

CIA

LAS

Any other double pair of instructions

Input-Output Instructions

KSRF=KSF

READ=KRB

TSS=TSF

TLS