PORT FE
PORT FE
PORT FE

Sorcerer User's Group Newsletter
762 Gerrard St. E., Toronto, Ont
vol. 1 no. 2
April 1980

EDITOR'S TURN

The first order of business this month is an apology from the president for the screw-up at the last meeting (no, not the weather - the change in meeting time and place). The Board of Education has screwed us up again. It now appears that our meetings will be held regularly on the 2nd THURSDAY of each month at the same school but in the cafeteria instead of in the staff lounge. The cafeteria is located in the basement and there is parking at the northwest corner of the building.

Secondly, remember that if you have not paid your membership dues, this will be your last newsletter unless you do so by April 15, 1980. The board is not going to chase after you for your money, it's up to you to get it to us. A membership agreement form is attached to this letter which must be signed by all members. It can be mailed or given to one of the board members at the next meeting.

Lastly, I'm a little disappointed since I received no offers of help for the newsletter, no articles, no reviews and very little feedback. I sincerely hope that this will change in the future. If you have any programming techniques, hardware or software reviews or complaints that you'd like to share with us, please send them to me:

John Roy
200 Balsam Ave.,
Toronto, Ont
M4E 3C3

or give them to me (if you can track me down) or call me at 691-2810.

SOFTWARE

I had the occassion recently to do some programming with strings using the BASIC ROMPAC and found a couple of interesting points regarding the Basic Control Area and the way that Basic handles strings. Although it is not complete, I hope that it is of some use to you.

String variables are named in a variable table which follows the programme data and string arrays are kept in a table which follows the variables. However, as you probably already know, these tables only hold pointers to where the strings are located and not the actual strings. Basic takes all ASCII data it receives, whether from the keyboard, from programme computation or peripherals, and slushes it at the top of memory (below the MWA). Current string values are pointed to in the tables by their sixteen bit address pointers. When new strings are assigned or programmed, the ASCII keeps piling up until the buffer area that you specified in your programme is filled. Needless to say that memory is quickly contaminated by this junk. Basic then employs a 'garbage collector' facility which picks up all current strings and compresses them to the top of memory and starts over again. Eventually, if you have not specified enough string space, the buffer will be filled with valid data and an OS error will occur. However, you can control all the string handling parameters including the 'garbage collector' from the BCA.

> Location 145H (325D) contains a 2 byte address which specifies the end of the allotted string space. (Notice the 50 byte default.)
>
> Location 192H (402D) contains a 2 byte address which specifies where the string area starts.
>
> Location 1A6H (422D) contains the 2 byte pointer to where the next string is to be put.

On initialization, location 1A6H=192H. As strings are entered or computed, 1A6H changes until it reaches the address specified in 145H. When it does, the 'garbage collector' is automatically invoked and compresses all valid data up in memory. By changing these addresses via POKE statements, you can place a string anywhere in memory that you want it stored, move the string slush block somewhere else or invoke the 'garbage collector' at any time. As an interesting experiment, try setting the string slush block in video RAM and watch how BASIC handles it's strings.

If anybody has any more information on the BCA, please share it with us. Perhaps we can fill in all the information EXIDY and MICROSOFT feel that we don't need to know.

HARDWARE

If you are running CP/M on a Micropolis Disk System then you can also run all your old Basic programmes that were developed with the ROMPAC from your disk with NO software modifications and only a minor hardware modification. By changing the Micropolis controller card in your S-100 buss, you can have your disk bootstrap at BC00 instead of DC00 and keep your ROMPAC (whether it be BASIC, WORD PROCESSING or the DEVELOPMENT PAC) in the sorcerer. When you load your programmes from the disk, simply make a warm start to the ROM. The hardware modification shown in Figure 1 involves removing the green jumper that you put on the board initially to bootstrap at DC00, soldering a jumper on the back of the board from point A to point B and cutting the trace between points C and D. It's that simple.

The most simple case is with Basic. Boot CP/M from the disk and re-enter the Exidy monitor. Now make a warm start to the ROMPAC by typing PP<cr>. Load the Basic programme from the cassette using CLOAD. When loaded type BYE to re-enter the monitor. Find out where in memory the programme ends (this can be done by adding 100H to the programme size.) Now make a warm start to CP/M by typing GO 0. When CP/M comes back on, type A>SAVE X FILENAME.COM where x is the number of pages to be saved and filename is any name you want. Since location 100H in the RCA contains a warm start to the ROMPAC, when CP/M auto executes at that address when loading a file, the jump to the ROMPAC will be automatic.

If saving from the WORD PROCESSOR or DEVELOPMENT PAC, then it will be necessary to find the data, move it down to 100H for the disk save and back to it's proper location before a warm start to the ROM. This can be easily accomplished by a simple block move routine.

EDUCATION

As I mentioned last month, I will be giving an in-depth course on 8080 and Z-80 Assembly Language Programming. Due to the poor attendance at the last meeting, there was very little response. I have decided to hold off until after the next meeting so that more people will have time to enroll.

Unless you hear to the contrary, the classes will be held on Wed. evenings from 7:30 pm to 9:30 pm at my home (as listed above). The first class will be held on April 16. The course will be in 2 parts of five lessons each at a cost of $1.00 per lesson. A minimum of $5.00 is charged in advance. A schedule of the classes is attached to this letter. All those who are interested should see me at the meeting or call me in advance.

THAT'S ALL FOLKS

Next meeting - Thurs. Apr. 10/80 at Jesse Ketchum in the cafeteria.

8080 and Z-80
ASSEMBLY LANGUAGE COURSE
SCHEDULE

Apr. 16
- 8080 and Z-80 architecture
- Number Systems HEX-DECIMAL-BINARY-OCTAL
- 4, 8, 16 and 32 bit systems

Apr. 23
- BOOLEAN Algebra
- Concept of memory addressing
- 8-bit Load, Arithmetic and Logic Groups
- begin flag operations

Apr. 30
- Languages - Level and Order
- Variable Concepts
- 16-bit Load, Arithmetic and Logic Groups

May 7
- Stack Operation and Theory
- Jump, Call and Return Groups
- CPU Control Groups
- Exchange, Block and Transfer Groups
- ASCII

May 14
- Rotate and Shift Groups
- Bit Set, Reset and Test Groups
- Input and Output Groups
- Index Registers and Indexing

May 21
- Assemblers, Compilers and Interpreters
- Editors, Linkers, Loaders and Debuggers
- Assembler Directives and Pseudo Operations

May 28
- Vectors
- Stack Management
- Data Management
- Programme Control

June 4
- DATA STRUCTURES
  - -free lists
  - -linked lists
  - -double linked lists
  - -bit maps

June 11
- Practical and Review

June 18
- Practical and Review

FIGURE 1.

MICROPOLIS DISK SYSTEM CONTROLLER CARD
(REAR VIEW)

*Concept Courtesy of TED SEKI