

APPLE THREE USERS OF NORTHERN CALIFORNIA
VOLUME 5 NUMBERS 3, March 1988
Sorcerer's Apprentice, Part 5,
by Wayne Schotten

This column and its codefiles can be ordered from the ATUNC library. This will SAVE YOU the TEDIUM and TIME of typing the programs printed here, WHICH YOU CAN USE FOR DEVELOPING PROGRAMS OF YOUR OWN TO CONTRIBUTE TO THIS COLUMN.

Programming is the most fun you'll ever have with your computer. So get busy and try some. If you're already programming, send some samples of your own for inclusion here. The Unanimous Group in San Jose meets regularly to learn Pascal together. I'm sure you'd be welcome to join them. Call John Cowman 408-259-9006 for info. (Hey You Nanimer's, how about something for this column in Pascal?) All Apple /// languages are fair game. Send contribution, questions or suggestions to me at Precision Audio, Pier 26, San Francisco 94105 or 415-541-0960.

Track Ball

We've been talking about the Track Ball by Dale Sykora, now available through On Three. Dale has written patches to the .console driver, so now not only will it work in place of a mouse, but it will work with any program that uses the standard keyboard cursor keys. The Catalyst .console driver requires a different patch, but that too is now available. If you already have a ball, but do not have the .console patches, get in touch with me.

Dale is developing other projects that you're going to love. His Optimizer program, being user tested now, facilitates building and editing your SOS.Driver files, including getting full use of the allotted memory space.

Hardware projects by Dale are well along, but a different approach may be required if YOU are to benefit. However, if we ourselves don't get involved, then the /// you have now is all the /// you'll ever have. Enthusiastic discussions at our ATUNC meetings point to the possibility of producing printed circuit boards for Dale's projects, so they could be assembled as kits. These may include a color graphics board with double the resolution of a IIGS and millions times more colors, a massive expansion of RAM, a slot extender for using more plug-ins, and an any-program color text adapter.

ASSEMBLY LANGUAGE MODULES:

The Sorcerer's Apprentice is beginning a look at assembly language programs. In Applesoft BASIC you can poke assembly language routines into memory and this facilitates a little experimenting and play. Business BASIC does not have this function, and not without good reason. Memory management is handled by SOS, and it places and moves things around in memory in a difficult to predict fashion.

Our BASIC assumes that your assembly language routines will be written and assembled with the Pascal system, and then INVOKED by your BASIC program. There are some invokable modules already in existence which enable you to do things faster, or better, or do things you can't do any other way. So far as I understand, both BASIC and Pascal can use the same modules, the only difference being that Pascal calls them as external functions.

I want our library to collect as many of these as may already exist, and if you have few of your own, get them into the library ASAP! I have at this time the following modules:

(From the Business BASIC Language Disk:)

BGRAF.INV - using the .GRAPHIX driver directly is no task for the faint hearted. BGRAF (called PGRAF in Pascal) gives you flexible and

logical means to produce graphics on the console screen.

VOLUMES.INV - returns status information of devices.
TIMESSET.INV - reads or sets the /// clock and calender.
RENUMBER.INV - moves and renumbers lines in BASIC.
READCRT.INV - returns character displayed anywhere in viewport.
DOWNLOAD.INV - changes the system font.
REQUEST.INV - enables direct access to block and character devices, for status checks, and such.

(From the Silentype Driver Disk:)

PSCREEN - provides a variety of printing formats of graphics screens to Silentype.

(From the EXERSOS Disk:)

SCREEN.CODE - additional .CONSOLE functions.
TOOL.CODE - provides SOS calls directly from BASIC.

(From BASIC EXTENSION:)

MATRW - read and write matrixes to disk as binary files, up to 10 times faster, normal files may use 20 to 30% more space.
BLOCK - read or write directly to any part of disk, very high speed access outside of normal SOS structure.
FTYPE - read or change type of file on disk, use with MATRW for access to any kind of file.
MOVE - high speed manipulation of arrays, insert or delete elements, move sections, transfer between arrays.
SEARARRY - similar to BASIC's INSTR, but with extra features including ignoring case.
CON - an extended GET for the whole /// keyboard, plus set length of the type ahead buffer.
REBOOT - clears memory and displays reboot message.
RESET - disables and enables CONTROL -RESET.
BIT - provides bitwise OR and bitwise AND functions.
UPSHIFT - converts a string to all upper case.

(From the PROGRAMMER'S POWER TOOLS:)

PPT.INV - includes case conversion, RESET enable/disable, disk formatting.
SORT.INV - sorts string arrays.
NUMSORT.INV - sorts array of real, integer or long integer numbers.
SEARCH.INV - fast searches through arrays of strings.
NUMSEARCH.INV - fast searches through arrays of numbers.
GENINPUT.INV - similiar to INPUT, but controls and traps out many incorrect entries, options for fast out, like ESCAPE in /// EZ Pieces (W.S).