VOLUME 4 NUMBER 8 AUGUST, 1987

THE SORCER'S APPRENTICE: Or programming on the Apple ///

by Wayne A. Schotten

If you simply sit at your Apple ///, pushing buttons on the keyboard, running somebody else's program, never desiring to instruct the machine to do a trick or two of your own, then you're cheating yourself out of the most rewarding sensation that a computer user can have. You may as well go watch TV!

So what if they're not making ///'s anymore? You have a several of the most popular languages available. Once you learn a language, you can use it on any computer that uses the same language. On the /// you won't be learning a cockney or hillbilly dialect written for toy computers, you'll be learning a high level version.

This column could be a regular newsletter feature, with programs, algorithms, funstuff, Q&A's, if you'll help by contributing. Send any of the above to Wayne Schotten, % Precision Audio, Pier 26, San Francisco, CA 94105, or call me at 415-541-0960. I've got a modem; if you want to try that, call me!

Whatever you do, try some programming yourself. Here's a list of our tongues.

- WPL: You get this with AppleWriter ///. It's simple to use, compact, and very, very useful.
- Business BASIC: Apple ///'s own, and a good one. BASIC is versatile. You can type just one line and do something useful, or you can write an enormous program occupying many disks of artful and intricate problem solving.
- Pascal: Language of choice for many multilingual programmers. Excellent string handling, and its ability to define files, arrays, records, and data types is surpassed only by ADA. Not everything in Pascal is easy, but it probably is possible.
- COBOL: The language of data processing. Probably somewhere near 80 to 90% of all business computing is performed with COBOL. Some call it structured, I call it strict. If you enjoy exactness, this one's for you. Maybe I'm a little strange, but I like it.
- FORTH: Oh Boy! This is the language that for me comes closest to the ideal I had of the computer, before I actually learned to use one. FORTH starts very small and simple with rules that enable you to define new words, thereby causing it to grow in the direction most useful to you. Plus, it's fast enough for real-time, control applications.
- FORTRAN: Where would science, mathematics and engineering be without FORTRAN? With a little luck, and a lot of work, we may have a FORTRAN surprise in the A.T.U.N.C. library.
- Microsoft BASIC: Supplied with the Softcard containing the Z80 microprocessor and the CP/M operating system. Microsoft is used by a lot of other computer manufacturers, and so you can run programs written for them with little or no modifications. Has a few words or functions not found in Business BASIC.
- Applesoft BASIC: You get this Apple][version in Emulation Mode. Probably the most widely understood dialect of BASIC in the world, since it is the resident language in the][.
- Integer BASIC: Older than Applesoft, you also get this Apple][version in Emulation Mode. Integer BASIC has no decimal point arithmetic. I've had problems in this implementation. Maybe someone can help me.
- 6502 Assembly Language: Very important, but you won't need to tackle this for awhile. The microprocessor itself at the heart of your computer operates on Assembly Language, in fact all other languages must be interpreted or compiled into assembly language, before the processor can use

it, and the translator itself is an Assembly Language program.

- C: Resembles assembly language, conferring a lot of the privileges of Assembly in a higher level, friendlier format.
- Modula-2: A sort of extension of Pascal, clearing up some difficultiies it has. More portable and more flexible.
- ADA: Don't I wish! Probably NEVER have it on the ///, but it's likely to be the main language of the future. It was created to be the common language of the entire military system, cabable of everything from accounting to digital weapons control. It probably won't be anybody's favorite language, but it is versatile.