

My name is Stan Shell and I work in the Marketing organization of the General Systems Division. I would like to briefly go through an announcement of a new subsystem on the 3000 Series II: an APL compiler. Announced last week at an APL users convention in Canada, it took approximately 15 man years to develop this subsystem. APL was developed at our HP Advanced R & D facilities in Palo Alto, and it will be marketed by General Systems Division. It runs under MPE, and is currently being tested at Yale University. By the way, the reason why it runs only on the Series II, and not Series I, is APL involves such features as four word floating point, and special instructions that are only available on the Series II.

Now, what is the significance of the APL Compiler? We feel that it is quite a significant advancement in that it is the first example of a full scale, full capability, APL on a relatively "small" computer.

The next slide mentions a couple of the features that we are quite proud of. With APL there is a feature called work space, in which all programs, data, variables, etc., fit into an APL "environment". Typically on other APL offerings this is limited to 80Kbytes and has been significant resource restriction. With APL/3000 we have implemented a relatively unlimited work space. (It is limited only by the size of disc storage on the system.) As a comparison between the typical 80Kbytes working space, APL/3000 starts each user with a work space of 800Kbytes. There is special hardware that assists the implementation of this dramatically larger work space. It consists of several chips which fit onto the extended instruction set board. These chips implement 11 new instructions which implement the virtual work space.

Another unique feature of HP's APL is a coexistent editor. Typically, APL Editors are primitive, however, HP's is a sophisticated one similar in many respects to the Text Editor. While it doesn't have all the capabilities that you can find in the Editor/3000 advanced features, it is a little cleaner for some of the more simpler commands. For instance, it has an undo command; if you don't like what you have

modified with the APL Editor, you can say "Undo it", and the text will return to its original state.

APL/3000 coexists with our MPE file system. It also has a feature that makes it the very significant APL announcement. That feature is called a Dynamic Incremental Compiler facility. APL is typically an interpretive operation, whereby each line of code is interpreted a line at a time. Machine instructions are generated, then executed and then the next line of source is interpreted by the interpreter, machine instructions generated, executed and so it goes a line at a time.

With APL/3000, as a line is interpreted, the corresponding instructions generated are saved. During the next pass through that part of the code, an evaluation is made as to whether the object code has changed such that a reinterpretation of the source code would be required. Of course, if need be, the source code is reinterpreted and new instructions are generated and executed. However, if it so happens that the line need not be reinterpreted, then reinterpretation does not occur. It simply executes the previously interpreted instructions. So with APL/3000, the first pass through the program might be relatively slow as is typical with all interpreters. But, the second pass through the program and the third and fourth, etc., will probably be much more rapid since APL didn't have to interpret all the instructions. This feature is a Dynamic Incremental Compiler.

In short, we see this new subsystem of the Series II as being a significant advancement for users of APL. The availability of an APL this powerful on a machine the size of the HP 3000 is really a significant achievement, and makes APL even more effective in solving the problems that APL users have today. And, we also believe that if you take a look at APL, it might make your life a little bit easier, too.