Infotek Systems
1400 N. BAXTER ST. / ANAHEIM, CA 92806 / (714) 956-9300 / TWX 910-591-2711

FALL 1977

INFOTEK UPDATE

A FAMILY OF POWERFUL ENHANCEMENTS FOR THE HP9830A/B

The Hewlett-Packard 9830A/B desktop computer system is one of the most powerful computing tools available to any user in need of small to medium scale capability. As users of this remarkable machine know, there is very little that the system cannot do, independent of the element of time. As major users of this system ourselves, in both scientific and commercial applications, we have been impressed as much by the system's potential, as by the manufacturer provided accomplishments. In fact, we have recognized that the only significant parameter not sufficiently addressed by the manufacturer has to do with basic speed.

After several fruitless attempts at persuading the manufacturer to devote effort toward making a good system better, we resigned ourselves to the fact that enhancements would have to come from elsewhere. The position of the manufacturer is typical and understandable. They believe that return on their available engineering resources can be optimized only through the development of totally new systems which satisfy the need for additional CPU horsepower. Unfortunately for the user, this frequently means disposing of current generation equipment at a fraction of its potential value and converting software and certain salvageable peripheral devices to the next generation system. We felt that it was time for someone to end the kneejerk reaction that occurs each three to five years as the manufacturer deems it appropriate to "announce" the next generation. With this in mind, we set out to make it possible for the user to enhance his current system at negligible cost relative to conversion to the next generation, obtain roughly equivalent computing power, and do so with minimal effort.

We reasoned that the accomplishment of these goals would make it possible to economically skip the next generation of hardware and save the user time to attend to applications rather than system integration. Indeed, the goal of every dedicated data processing manager is to delay the costs of upgrading to new system hardware and software for as long as economically feasible. This gives the user a respite from the treadmill of hardware upgrades, software revisions precipitated by hardware differences, and back for another cycle just when the software is debugged. By skipping over a generation of hardware, applications can be brought to maturity. A period of stability in DP operations results.

Based on these assumptions, a tremendous engineering and software effort was undertaken to gain an understanding of the HP9830. With such knowledge, we could design system enhancements with the same degree of latitude as the manufacturer. However, this was not an easy task, for Hewlett-Packard, more than most computer manufacturers, does not release technical data in the areas that had to be understood. We had to employ a virtual regiment of sophisticated instrumentation, some manufacturered by HP, to pull us through the maze of hardware with no generic markings and interminable labyrinths of unknown CPU instructons. After months of persistence we finally felt that we understood all the major components: memory, I/O, CPU and most importantly, software at the assembly language level. We realized that we could fulfill our every wish in terms of additional capability. It was as though we now had HP asking us what we needed! By February, 1976, our first product, the EM-30 double density memory system was demonstrated at a computer show. Having gained a complete understanding of this machine, we were even more in awe of the design genius embodied in its architecture and completely taken by its inherent but hidden power.

Now that we had the ability to expand the BASIC language instruction set in any way we desired, our only remaining quandry was the selection of new instructions that would have the broadest application, concurrent with significant impact on system performance, particularly speed. Our efforts have resulted in the Fast Basic series of plug-in ROMs, significant improvements in the HP9880B Mass Memory System and a family of unique, high performance peripheral devices, as well as the

original extended memory system. At this point we can count over 1,000 systems using these enhancements.

FAST BASIC I

The first plug-in ROM produced, concentrated in the area of speed enhancement and also significantly strenthened the Mass Memory System instruction set. Since operations performed by ROM instructions are nominally 150 times faster than the same function implemented through a BASIC subroutine, it is possible to achieve astounding increases in throughput. Among the major capabilities in Fast Basic I are SEND, COMPARE, SWAP and SCAN. Each of these statements manipulates entire arrays of data several thousand times faster than previously possible. As we felt that HP had completed all development activity for the HP9830, we optimized our ROMs by calling subroutines in existing HP ROMs to maximum advantage. This enabled us to pack 38 statements, functions and commands into this first ROM alone.

Several Mass Memory statements and functions were incorporated to provide users the ease of moving swiftly through files of data and updating the contents on an item by item basis. Some of the design ideas were patterned after other powerful implementations of BASIC such as the HP3000 series. We placed maximum emphasis on adhering to HP9830 language conventions and the "spirit" of BASIC.

FAST BASIC II

During our evaluation of the HP9830, we discovered a large area of unused firmware space submerged in the memory system architecture. Given this, the limitation of eight ROM positions in a machine and the realization that before long many users would run out of places to install new ROMs, we determined to put this space to good purpose. From this, Fast Basic II evolved. This ROM was designed to reside in the available space rather than consume one of the eight precious ROM positions. Furthermore, we implemented the important features of the HP Advanced Programming I ROM not duplicated elsewhere in a more flexible manner. Now the API ROM could be removed from those machines that used it, thus actually freeing a ROM slot for Fast Basic I.

Fast Basic II has several powerful cassette oriented statements which, when coupled with other cassette capabilities in Fast Basic I provides the user with practically unlimited tape handling abilities. These become even more synergistic when the FD30 floppy disk is used in lieu of the HP9830 cassette system. Of equal interest are two types of cross-reference generators and a power on/auto start capability that works with either the 9880 disk or the 9830 cassette.

FAST BASIC III

The latest and most powerful ROM in the Fast Basic Series provides high speed I/O at well over 10,000 bytes per second data transfer rate, keyboard interrupt, program suspension to cassette or floppy, and the unlimited ability to dimension arrays of any size. Each of these features is a major breakthrough independently. Taken together in one package, it is a potent combination.

The system I/O speed was limited to about 900 bytes per second, even with a binary program utility. The MREAD/MWRITE statements increase this speed by an order magnitude. Simultaneously, application flexibilities previously available at a maximum transfer rate of 150 bytes per second are also provided at no reduction in speed.

A long awaited feature is the ability to contact a running program without halting the machine. The KEY function provides this capability along with a choice of two modes of operation. Now control can be transferred, flags set and variables altered - even while the program is running. This is significant considering that the system had no user controllable interrupt structure.

A problem in many applications has been the dilemma of what to do when the operator wants to go home or run a priority program but the machine is still busy with a previous task. With MDUMP the user can now swiftly suspend the program, all variables and flags, and even mass memory operations to a cassette or floppy. This is accomplished simply by pressing two keys on the keyboard without stopping program execution. As many users have discovered, the STOP key destroys certain subroutine returns and FOR-NEXT loop nests when activated. MDUMP eliminates this risk entirely. A suspended program can be resumed quickly with a single statement and the selection of two keys. The program never knows it was suspended.

Perhaps the most powerful feature of all is the ability to deal with arrays of data and vectors of any size that will fit into machine memory. Previously, applications had to battle the frustration of an artificial limit of 256 elements per side of an array. Thus, the sorting of 1,000 items presented a complex problem. Now such tasks

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

are accomplished by a single statement, eliminating over a thousand words of basic program and gaining an incredible speed increase. Programmers can design software more closely representative of the applicaton at hand rather than machine idiosyncracies. Usually, programs will run much faster when dealing with appropriate array dimensions, and in some cases it is possible to use the HP9830 efficiently where previously it would have been avoided for such tasks.

MASS MEMORY II

Recently our attention was focused on the needs of the HP9880 Mass Memory System user who unquestionably has the largest investment in an HP9830 System. Coincidentally, such a user has the most potential for gain because when properly configured, and with good software, the system can outperform many mini-computer systems of other manufacture.

The Mass Memory ROM was modified so as to keep track of current cache memory contents. At the same time, certain frequently referenced system bootstrap functons were made memory resident in conjunction with the EM30 double density memory system. Mass Memory II gives much more than it takes in that NO user available memory is reserved for this purpose. This was possible due to the HP9830 memory architecture in combination with the EM30 system. The system is now much faster and the majority of the noise produced by the head carriage assembly during bootstrap seeks has been eliminated.

Additional powerful features such as the ability to update a program on disk from a new version in memory with a single command has been incorporated. Also, binary utility programs such as Platter-Duplicate can be saved and retrieved from the disk. Space management has been automated so that contiguous areas are routinely combined whenever a file is killed. The system can be requested to create files of maximum possible size by a new option to the OPEN statement. Also, the inconveniences associated with REPACK have been eliminated in that it now resides in the system bootstrap area as a programmable statement. Contrary to previous methods, none of these operations cause memory to be cleared.

Having been there ourselves, we have attempted to consider the user in an application environment. As additional ideas are generated from within and without, new products will be created to fulfill meaningful requirements during the mature years of this computer system.

THE BEST NEWS YET FOR ALL 9830 USERS

Hang on to your keyboards '30 users, Infotek's done it again! The 256 row-column array dimension limit, the only frustrating aspect of '30 BASIC is gone...banished by Infotek.

Now you can dimension arrays any size you like. Just think of the endless opportunities for things like sort, searches and large but manageable data arrays. A SORT of 4000, 8 character items contained in a 4000 by 4 integer array or a sort of 1000 rows in a 1000 by 4 full precision array now executes substantially faster. How about a 16,000 word vector? You bet. If you have Infotek's FAST BASIC III ROM you can dimension an array any size whatever, provided only that the product of the subscripts for the precision used does not exceed available memory.

And that's just one of the many FAST BASIC III's new tricks. Additional capabilities of this ROM include I/O rates of over 10k bytes per second, ability to suspend a running program to tape or floppy without halting the program, interrogate and/or ammend variables also without halting the program, and allowing a peripheral to demand service. Then, for good measure, FAST BASIC III allows you to address an array location by element number as well as subscripted location, convert a string to its upper case equivalent, shift an integer number by 1 to 16 bits left or right, and extract the decimal ASCII value of a string character.

The 9880B Mass Memory owner gets several new goodies too. First, he can MAT PRINT or MAT READ the new big arrays without confusing the Mass Memory. Second, the string to ASCII number converter allows a four-fold increase in the amount of data that can be crammed on a disk record when using integer precision arrays containing transferred strings. And the data can be conveniently recovered!

With the RXREF binary tape provided with FAST BASIC III or Infotek's new MASS MEMORY II ROM you get a cross reference listing of ROMs used in program lines or a selective program listing by ROM type. RXREF will produce a listing of each ROM used in the program, followed by every program line that uses that ROM. What if an ERROR 1 occurs? No problem. RXREF will identify the missing ROM. The selective list allows you to list only those program lines which use the specified ROM. For

example, RXREF #15,8 will provide a listing on printer select code 15 of all lines containing Mass Memory instructions. RXREF #2,3 will do the same on printer select code 2 for the MATRIX ROM and so on. All HP and Infotek ROMs are covered, and provisions are made for any ROMs that might be added by either HP or Infotek in the future.

THE 9880 MASS MEMORY IS NOT FORGOTTEN

All of us working with the 9880 certainly have the right to consider ourselves among the most serious '30 users...after all, we've spent an average of 15 kilobucks extra for the honor. So, if you're one of the 15-kilobuck clubmembers, read on. This is for you! The rest can go on to the next article where you'll find news of a more general interest.

Did you ever ask yourself why that little FD-30 floppy of Infotek's is almost five times faster than the 9880? It's almost unfair! After all, there's a ten-grand spread between the two gizmos, so what's the explanation? Better yet, can the speed gap be closed? Ask no further. The problem was...and we mean WAS, that the 9880 spent much of its time reading the disk when the data is already in cache memory. Also it read bootstraps each time MAT PRINT and MAT READ were encountered to see what they are. Finally, the directory was read repeatedly even though the file assignment has not changed, only the unit designation. So you see, it's a simple matter of lost time on disk motion, waiting for the head actuator to stop thrashing about.

Now here is what we've done to fix this problem. First, we put the most frequently used bootstraps into the 9830's memory, so right off the bat there's no more crashing from data to bootstraps and back every time you do a MAT PRINT, MAT READ, FILES or IF END. Just look in memory...it's much faster and there's no thrashing about. But that chews up memory you say? No sir! You know that 1,024 word hunk of memory you can't use because it shares addresses with the 9880's cache memory? You guessed it. The first time a FILES statement is executed, the code is transferred from disk to that unusable memory. After that, it's all gravy.

This effort forced us to decipher how the bootstraps worked. Having gained that knowledge, we went on to fix some of the other peccadillo's of the 9880. Next to fall before the sharp pencil was READ. It seems that every time an item is read, the specified record is transferred from disk to cache memory. The fact that the desired information and perhaps many following items are already in cache memory is ignored. We test to see if the desired item is in cache memory and if not, we go to the disk, but most of the time we just get it from cache memory. Voila! So much for readng the disk 64 times to get 64 items!

Another aggravation is the way DAVTP works. If you add enough to an existing program so it won't fit in its original location, then it will be stored in the first available space that is big enough. However, before you can save the enlarged version you must first kill the original. That puts a hole in the availability table which may very well be adjacent to another hole, and the sum of the two is sufficient for the new file. But alas, you'll never know it! To combine the contiguous areas of the availability table you have to do a DAVTP first. This is another way of saying you must do a SCRATCHALL first! Of course you can't do that because you've got a program in memory that must be saved. So store the program on tape? Of course not! You don't really have a choice. You simply kill the old file, save the new one, and hope that someday you'll remember to clean up the availability table by doing a DAVTP. In the meantime the disk system space utilization efficiency keeps decreasing as the availability table continues to look less and less like the actual space on the disk.

To solve this problem we implemented an auto DAVTP. Each time you do a KILL a DAVTP is automatically executed. But what about the SCRATCHALL? No problem, we fixed that too. As long as there are 770 words of unused memory remaining in the machine, the auto DAVTP will be performed and your memory will remain unchanged. You'll find that this feature really holds your entries together.

Of course this whole business of what happens when you modify a program, kill the original and save the new, leads in very nicely to the next feature. This is the RESAVE capability. An optional parameter has been added to the KILL statement which, when used, kills the specified program, does a DAVTP and then saves the program in memory...all in one fell-swoop. To do this, the KILL statement is as follows:

```
KILLS,"file name"
```

Want more? We've got it, and this one's a dandy! How do you like the idea of being able to save and get binary programs with the 9880? Well we did that too. Now, when you want to do a repack, platter dup or anything else requiring a binary utility, just bring it in from the disk. No more staring at the

cassette as it winds to file 80...well once more - so you can save it on disk!

Another foible to go is DCOPY's aggravating and usually erroneous conclusion that the first logical end of file mark in the source file is the end of all data in that file. DCOPY is now much more attentive to detail and it even takes care of the user. First it will copy the specified file in its entirety, END markers and all, but nevertheless, intelligently. For instance, if a destination file has fewer records than the source file but the data does not overflow the destination, all is well. If overflow occurs, then and only then, an error is issued. The cherry on top is DCOPY's new willingless to accept variables for unit designatons.

And yet another powerful capability that comes along with Mass Memory II is the ability to do a selective catalog of the disk. Now when we say selective we really mean selective. First, you can get your catalog alphabetically. Then, you can get it alphabetically but beginning at a specific letter so that you only get a partial catalog. On top of that, you can have a catalog that is ordered according to origin, so that the files roll out in the same order that they appear on the disk in terms of track and record number. And if that's not enough, you have the choice of listing according to origin with any starting entry point. Now that's really great for the fellow that has 600 names on a disk and is interested only in the last 10. With the HP system there's nothing to do but wait for the thermal printer to roll out ten feet of paper (in which you are patently disinterested) so that you can see what is in the last ten slots. With the selective catalog, just type in 591 for an entry point and it will ignore the first 590 entries giving you a catalog of the last ten. What's more, the selective catalog will operate on any specified select code for a printing device. You're not tied to select code 15, like it or not.

Finally, the icing on the cake. The selective catalog gives you a place to put in the time and date that the catalog was made, automatically prints out the unit number being cataloged and, best of all, at the completion of the catalog it gives you all of the information pertaining to the availability table. It tells you how many and what size files are empty, the total number of files on the unit, the total number of records used, the total number of records available and the size of the largest file that may be opened. It even prints out the number of defective tracks, if any. How's that for a catalog!

For good measure, PLATTER-DUPLICATE now VERIFIES the copy, a function previously performed by assumption, defective destination disk tracks notwithstanding. You should have heard the uproar following a disk crash when we discovered our belief that we had a backup platter was only an assumption. No more...Now you get a VERIFIED duplicate or an ERROR 90. FAST BASIC I allows you to determine if the ERROR 90 is a source read fail or destination verify fail.

Finally, the CAT listing now identifies the UNIT number and the bootstrap as Infotek modified.

In operation, MASS MEMORY II in conjunction with FAST BASIC I benchmarks to a better-than-two-fold improvement in work throughput for typical applications. In addition to this gain in speed, there is a significant reduction in the requirement for disk system adjustments because the mechanical work of the head actuator assembly has been radically reduced. Repetitive MAT READS on one unit followed by MAT PRINTS on another unit, which used to sound like an old washing machine on rinse cycle, is now totally silent under the firm hand of FAST BASIC I and MASS MEMORY II. With the silence has been gained the time not devoted to wear and tear on the head actuator assembly and servo system.

Before Infotek went to work on the MASS MEMORY ROM and the bootstraps, the 9880B system had a lot going for it. Now, with FAST BASIC I, MASS MEMORY II and the improved bootstraps, the 9880B is unbeatable in the world of small system multimegabyte disk drives.

BUFFERED PRINTING IS CHANGED

The BPRINT or buffered printing capability of FAST BASIC III which appeared in the first issue of the preliminary data sheet is now very conspicuous by its absence in current literature. Do not despair, however, it has not been dropped from the Infotek products list.

Several factors persuaded us to implement this capability differently. Taken together, these factors were compelling. First, our lab came in with a breakthrough for circumventing the 256 array dimension limit. Clearly this new capability has a much broader application than buffered printing. The second factor was that BPRINT was not extraordinarily effective except when used in conjunction with slow printers such as the 9861A FACIT, the HP9871A CHARACTER PRINTER, a TELETYPE, or CENTRONICS 101A. In other words, at 10 to 60

characters per second, BPRINT is a bonanza. With printers like the HP9866A thermal or the "Tallywacker" 200 lines per minute which operate in 300 to 400 character per second range, BPRINT is only an enhancement rather than a necessity.

Finally, the third strike against BPRINT in FAST BASIC III was the fact that the amount of code space required to implement the arbitrary array dimensions was just about what BPRINT occupied. So...out with BPRINT and in with big array dimensions.

As with most things, everything has worked out for the best. BPRINT used 9830 memory to whatever extent it was allocated by the user. The slower the printer the more memory was likely required to avoid getting printer-bound. Since the print buffer was in the 9830 and had to be managed, it did take some of the time saved in not waiting on the printer, so the speed gained was not complete. Everything considered, it made good sense to implement the BPRINT as an integral part of our new CP-30 and LP-30 printers. This gives the user all the benefits of BPRINT plus the very powerful and perhaps even more universally valuable benefit of extending the 30's memory by the size of the buffer.

By implementing the buffer in the printers it gave us the opportunity to provide several modes of buffer operation.

First of course, is the mode where data zooms from the '30 to the buffer. The '30 can immediately continue computations while the buffer services the printing mechanism without any involvement of the '30. Next, via software, the buffer can be made to emulate a very fast paper tape punch or paper tape punch reader. In this mode the buffer stores data input from the '30 but does not print it. The data can be read back from the buffer to an array in the '30 just like reading paper tape, only much faster. Also a program can be listed to the buffer and subsequently brought back into program memory via PTAPE. This of course immediately brings to mind program text editor utilities that work directly on the BASIC code and here, the sky's the limit. You can do virtually anything you like.

The third mode of buffer operation implemented in the CP-30 and the LP-30 allows a program statement to address specific locations in the buffer. In this mode, the specified location can be loaded or stored much in the manner of a vector. The practical result is that the 30's memory can be for many purposes, extended to 46K bytes by the CP-30 and 64K bytes by the LP-30.

The buffer, however, should not be considered as a one-for-one replacement for internal memory. After all, your matrix, strings or any other ROM can't access the buffer directly. This means that the selective mode will be slow as it is supportable only by basic code which must be interpreted. ROM speed enhancement for this aspect is not feasible. Those who have already ordered the FAST BASIC III ROM and the BP-30 buffered interface should not despair. A new interface is being completed. This interface offers all of the features of the CP-30 and LP-30 buffers. It will function with any printer and best of all, it also works beautifully with the plotter, paper tape punch, and many other peripherals presently interfaced to the '30. The new stand-alone buffer is presently interfaced to the '30. The new stand-alone buffer is designated our Model EB-30. Specifications and availability will be announced in the next few weeks.

THE NEW RT-30 INTERRUPT TIME CLOCK

We are completing development of the RT-30, a real time clock designed to take advantage of the FAST BASIC III ROM interrupt provisions. The RT-30 accumulates time directly in days, hours, minutes, seconds, tenths and hundredths of seconds, and outputs the data as a 12 byte field including a line feed terminating character.

A typical application such as a process control could be accomplished entirely by a program which sets the time, interrupt time and instructions to be executed on interrupt. Now let's assume that we are running a program which can be suspended temporarily without inconvenience and that somewhere on your tape or disk you have a process control program that must be run at a specific time. Let's further assume that the process control program is file 12 and the currently running program is file 3 on your peripheral cassette or floppy. File 0 is a program dump file.

At the set time for interrupt, the clock will issue the following commands. MDUMP 1, STORE 5, LOAD #5, 12,10,10. This will cause the running program (file 3) to be suspended to a floppy or cassette on select code 5 and then load and run file 12. The process control program, as its last act, simply institutes an MDUMP, LOAD #5 which brings back and resumes execution of the suspended program. This example will initiate the execution of the program on file 12 within a variable period of time related to the period required to suspend the running program and subsequently find, load, initialize and start running the new program on file 2. Since the peripherals involved may be

cassettes, floppys or the 9880 Mass Memory system, the times involved can vary greatly, particulary in the case of cassettes. However, if commencing a program at a precise time is required, the RT-30 can handle that job very nicely also. This can be accomplished simply by loading and initializing the program to be executed in advance and having the program halt on a stop statement just prior to the first line of execution. The RT-30 at the interrupt time will simply issue a CONTINUE command and the program will begin execution exactly at the set time.

Instructing the RT-30 to perform its various tasks is straightforward. Simple WRITE or OUTPUT statements are used to issue the instructions to the RT-30 in literals. For example, to set the date and time simply issue a WRITE(6,*)"SET 125,13,30,00." This will initialize the clock to the 125th day of the year, at 1:30 PM plus 0 seconds. Next, if an interrupt is desired, on the 128th day at 4:00 AM, simply WRITE(6,*)"INT 128,04,00,00." Finally, if the task to be accomplished upon interrupt is to LOAD and RUN file 12 from SELECT CODE 5, you WRITE(6,*)"KEY,LOAD #5,12,10,10."

Since the time required to read the clock is in proportion to the number digits read, the clock can be instructed to limit its output to a specified field. For example, to time an operation that is always less than one minute, the clock can be programmed to output only the seconds and fractions, the last four digits of the 11 digit field. The seven digits of days, hours, and minutes are not read. This is accomplished by the PIC instruction issued in a write statement. WRITE(6,*)" PIC S" preceding an ENTER(6,*) S,F will return the seconds in S and hundredths of seconds in F.

To read everything from days to hundredths of seconds, WRITE(6,*) "PIC D" and follow with an ENTER(6,*),D,H,M,S,F. The variables will contain, in order, the Julian day, hours, minutes, seconds and fractions of a second (hundredths). The specified variables can be any precision and any letter. Special treatments provided by the enter statement are all permissible.

The RT-30 requires slightly more power than a 9830 loaded with three other I/O cards can provide. Therefore, the clock has its own power supply and plugs into utility power independently of the 9830. A small internal battery is provided. It avoids nuisance resetting of time if power line transients occur or even if the clock is unplugged for transport. The battery is rechargeable and has a life of about five years. When fully charged, it will keep the clock running for 24 hours.

An optional digital display is available. The display uses 0.25 in (0.64 cm) high light emitting diode, seven segment numerals, and has overall dimensions of 0.812 inches (2.1 cm) high, 4.0 inches (10.2 cm) wide and 0.562 inches (1.4 cm) thick. It may be attached by supplied pressure sensitive double-sided tape to any flat surface. Connections are via a ribbon cable that may be run, for example, between the 9830 and the 9866, so that the display can be attached above the cassette door or anywhere along the top edge of the 9830 and the cable will not be visible. To extend battery life, the display is not energized during battery operation.

The RT-30 is priced at $1,200 and will be available in October.

A WORD ON WARRANTIES

The 9830 user has a significant investment and is understandably concerned about service and warranties. He wants to be secure in the knowledge that he will be supported. The question of how HP warranties or service contracts are affected by the use of Infotek memory, ROMs or peripherals is of primary concern.

Now that over one thousand 9830's worldwide are using Infotek products, and nearly two years of experience has been gained, here is what we have observed on the subject of warranties.

Based on many reports from Infotek's customers, it appears that machines containing Infotek Memories and/or ROMs, or machines connected to the floppy have been serviced by HP under service agreements. Furthermore, there was no hesitation on the part of HP personnel to support the HP part of the system. Since these reports have originated from various localities in the United States and several other countries, we believe that there must be an established policy in this regard.

Because of the consistency of the reports, it appears that HP services the HP portion of a machine containing Infotek memory or ROMs provided that the problem is not related to the Infotek portion. If the serviceman determines that the HP parts are not at fault, he advises the machine owner and bills the standard rate for a service call whether or not a warranty or service agreement is in effect. On the other hand, if the replacement of any HP part solves the problem, then the warranty or service agreement is honored. If the

warranty has expired and a service agreement is not in effect, the repair is billed at the standard rate.

The apparent policy conforms to common industry practice, particularly in the computer industry where many manufacturers are frequently involved in a single installation. Should anyone experience anything inconsistent with the reports we have received, please advise us, particularly if it's a problem. We may be able to assist you.

THE USER'S CORNER

In future editions, this page of the INFOTEK UPDATE will be reserved for you, the 9830A/B user.

If you have a suggestion that you feel will be of benefit to other '30 users, send us a note with a brief description of the information you would like to share. We will print as many of these as space permits.

SORTING LARGE DATA BASES

Here's an application for Fast Basic III that will be of particular interest to people currently going through the tedium of contiguous sorting of several thousand items.

The customary procedure for sorting thousands of items that contain more than one element per item is to use small arrays such that two or more contain data to be sorted and one additional array is used as a carrier. The number of arrays that can be processed between trips to the cassette is strictly a matter of memory size. Therefore, an EM-30 16,000 word memory all by itself can greatly speed up a sort, but for the most part, even with a big memory, sorting is very slow and the program itself takes a lot of memory.

A program to sort 6000 items of two elements optimized for a 16K word machine takes just under two hours to run. On an 8K word machine, this jumps to over three hours. The programs take from 1600 to 2800 words, depending on technique and memory size.

The following program does this job in about 2 minutes and uses a negligible amount of program memory:

```
10 DIM A1(2,120),B1(50,240)
20 For J=1 to 50
30 Load Data J
40 Send A(1,1) to B(J,1)
50 Next J
60 Store Data 51,B
70 VDIM B(6000,2)
80 VSORT B,C,2
90 Store Data 52,B
100 END
```

For the sake of simplicity, the program assumes the 12,000 words of data to be in 50 arrays of 120 words each. Lines 20 through 40 merely load all 12,000 words into a single array and are a one-time procedure. Thereafter, array B contains all the data which is stored in a single 12,000 word file.

Lines 70 and 80 do the entire job. Line 70 gives the array a VIRTUAL dimension of 6,000 rows by two columns, and line 80 does a 6,000-row sort according to column 2. The gain in execution speed is truly phenomenal.

INFOTEK IN EUROPE

In October, our European Service Center will be opened in Frankfurt, Germany. The center will stock the complete line of Infotek products to provide much faster delivery for our distributors in the Common Market.

The primary function of the center will be the training of distributor personnel in the support of Infotek products. Knowledge is strength and the more information we impart to the distributor, the stronger will be his ability to give direct support to you, his customer and 9830 user. The center will also provide comprehensive training for distributor service technicians so that field service can be rendered promptly and efficiently.

The center will include a fully equipped service facility for handling any difficult problems and will also maintain the European spare parts inventory.

Our Technical Director for Frankfurt is Mr. Theo Stevens who joined us in May after a four year association with Hewlett-Packard in Germany. He is totally proficient in the 9830 and the 9880 systems' hardware and software. For the past year he has been an applications engineer for the Calculator Products Group.

Mr. Stevens will spend three months with us at the factory. With his extensive knowledge of the 9830, he is quickly becoming proficient in all Infotek products, and upon his return to Germany, will be in a position to function effectively in imparting this knowledge to distributor personnel. We are very pleased to have Mr. Stevens with us and are certain that our European distributors, particularly our German distributor, Ernst Fey, GMBH, will profit greatly from his assistance.

The text for THE INFOTEK UPDATE was prepared, edited and printed on the Infotek CP-30, using our Manuscript Processor Software Package. The pages were typed on 11 in. by 17 in. paper. The text was then photo reduced for 8 1/2 in. by 11 in. litho plates.