A quiet revolution in personal printing

Norb Gotner/VCD

On March 12, 1984, Hewlett-Packard introduced in the US and Canada (and soon ICON and Europe) a new technology printer, ThinkJet, the HP 2225 Personal Printer family. They are quiet, fast, high-quality printers that use a new print head technology that will revolutionize the art of putting ink on paper.

Have you ever tried to concentrate on budget matters or draft a sales letter and had your concentration drowned out by a noisy impact printer pounding away nearby? Now HP has brought quiet printing into the office environment. Not only is it quiet (less than 50 decibels sound pressure), but it hums along at 150 characters per second, creating characters with an 11 × 12 dot matrix that hints at near letter quality.

Shift easily into four different print pitches with the HP Roman 8 character set, and boldface or underline without degradation of print speed.

Dot resolution is such that numbers are converted into clear distinct pictures with 96 × 96 dots per inch raster graphics dumps. Graphics resolution can even be increased to 192 × 96 dots per inch for even higher resolution.

The mechanism will use 8½" × 11" standard size or 21.5 × 29.7 cm metric A4-size paper in single sheets or fanfold. The best print quality is obtained on ink-jet paper designed for the ThinkJet family.

What really gives HP the technology edge is a unique disposable ink-jet cartridge that contains the actual print head and enough ink to print 500 pages of type. You have no moving parts, reservoirs to fill, or messy ribbons to change. The disposable ink-jet head used in the ThinkJet printers has a liquid ink supply and twelve microscopic nozzles. Each nozzle can supply a drop of ink on demand from the printer as the print head scans across the paper. A resistor heats the ink to its critical temperature and causes a spontaneous vaporization. The resulting vapor bubble rapidly expands upward and, in turn, pushes a tiny droplet of liquid ink through the nozzle and onto the paper. The ink refills the nozzle automatically by capillary action. As the ink is used up, the thimble-shaped rubber bladder collapses, providing both a constant back pressure and a visual indication of the amount of ink remaining in the system. The print head contains approximately 3cc of usable ink, which is sufficient to print 500 pages of text.

If you think that is amazing, take a look at the small package all of this comes in. Where else can you find a printer with all of these features in a footprint the size of a common sheet of bond paper? The dimensions are: 11.5 inches wide, 8.1 inches deep, and 3.5 inches high. The unit only weighs between 5.5 and 7.4 lbs., depending on the interface. Its other attractive feature is its low price.

The ThinkJet family consists of three separate printers:
HP 2225A — the HP-IB version for HP computers and instruments.
HP 2225B — the HP-IL version for HP portable computers and calculators.
HP 2225C — the Centronics Parallel version for other personal computers, such as IBM, Apple, Compaq, and TI.
Note: The HP 2225B is powered by a rechargeable Nicad battery pack.

Accessories available are:
HP 82199A — Spare battery pack for the 2225B.
HP 92261A — Print head cartridge (1).
HP 92261B — Print head cartridges (pack of 10).
HP 92261M — Ink-jet paper (500 sheets, single sheets).
HP 92261N — Ink-jet paper (2,500 sheets, fanfold).
HP 92261S — Acrylic printer stand.

CONTENTS

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Co-ordinator's Comments

CROSSTALK is not DEAD!! It merely drifted into a state of suspended animation for a period. I have been greatly encouraged lately by the number of people who have been asking me when the next issue will be out. Well, HERE 'TIS! My great concern is that your interest in reading Crosstalk far exceeds your interest in contributing to it. I know you are already up to your eyeballs in work to be done, but it only requires a small sacrifice of time to be made by a number of people to create a thriving and most worthwhile publication. At present nearly everyone is sitting back hoping someone else will write the articles. If your concern is that I may already be swamped with contributions, please call me, I would love to be able to say YES!

Remember, Crosstalk is your User Group publication but it needs YOU in order to survive.

GLENDA PATTERSON
(H.P. Melbourne)

PUZZLE PLACE

OK, I thought it was about time I got you to write a program to solve a puzzle.

MARCH

S	M	T	W	T	F	S
30	31					1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

The date of March 27, 1980, would be written by our American friends as 3/27/1980. This particular date is unusual since it has seven different digits — the first four (0, 1, 2, 3) and the last three (7, 8, 9).

Write computer programs to solve:
(i) How many dates in the 1980 decade also have the sequences of the first four and the last three digits, and what are they?
(ii) In the same decade, how many dates have the sequences of the first three and the last four digits?

See page 10 for the solution to last issue's puzzle.

2

FACTORY FLOOR DATA COLLECTION

Harold Norrie,
Nortronic Instruments
Phone: (02) 938 4994

"I keep six honest serving men,
(They taught me all I knew);
Their names are WHAT and WHY and WHEN
And HOW and WHERE and WHO."

Rudyard Kipling's little verse highlights the essential parameters for an efficient production management system. If a manufacturing process is thought of as a series of transactions, that series both describes the work flow and provides the means of control.

Products and accompanying documentation flow through a structured environment where productivity is described as a series of transactions occurring at specific locations and times. Any deviation from that orderly flow represents something a production manager should know about.

Traditionally, this data collection (in computer based manufacturing systems) has been a batch operation, with hand-written information on job cards, works orders and other documents being keyed into a system at some time after the event. Information available to production management therefore, was out of date and subject to human error in writing, reading and keying.

The fundamental move to get this data entry job done in real time holds the key to management knowing:-

	WHAT job
is at	WHICH location
having	WHAT done to it
by	WHO
at	WHAT time

One way of moving to a real time operation is to give every worker on the line (or each group of staff doing similar jobs) a terminal. Real time entry is now possible, but the user has to be able to type, data entry is slow and terminals are expensive.

When each document is produced with a machine readable symbol such as a bar code symbol, and a network of bar code readers is installed throughout the work area, real time data is gathered quickly, accurately and at relatively low cost. Perhaps the most important benefit is accuracy as automatic identification overcomes the human frailties of carelessness forgetfulness and laziness.

The document involved is a vital part of the process and may take many forms. Typically, we see a complete traveller produced by a dot-matrix printer, involving normal print and graphics to draw form lines, large characters and, of course, the bar code symbol. As the job or lot number points to all information on the traveller, this need be the only data encoded. In many industrial environments, some protection of the code is required and this may have an effect in the scanning process. Generally, a symbol can be read quite effectively, through a single clean plastic layer.

The choice of code is also dependent on the application, however Code 39 is used in most factory floor systems. This is because the code is alpha-numeric, is supported by nearly all reader manufacturers (both on-line and portable), is easily printed on dot-matrix printers and has very good first pass read and substitution error performance.

To provide automatic identification of the work station is considered necessary in an industrial environment as it reduces by half, the number of scans required to input data. That is, the operator scans a job number symbol, with the reader transmitting a message consisting of job number and location to the computer. This facility also improves operator acceptance of the system, ensuring its continuing use as a means of data capture.

The purpose of this discussion was to show how bar code scanning may be applied to the task of capturing real time production control data. The need to automate this process was made clear with a survey by a U.S. aircraft manufacturer who found that 60% of a line supervisor's time was spent gathering, checking and correcting information, instead of providing the support he should be giving his staff.

By encoding production documents and subsequently scanning these symbols, production management information may be captured in real time, accurately and quickly. The range of bar code readers manufactured in Australia by Nortronic Instruments are designed with this specific application in mind.

The HP 7550 — Quality Color Graphics in Quantity!

Need quality, color graphics in quantities for reports and presentations? Now you can have Hewlett-Packard quality **plus quantity,** quickly, with the HP 7550 graphics plotter including built-in automatic sheet feed for cut paper sheets.

Because the HP 7550 is designed to serve the graphics needs of either departments or individual users, you can make multiple copies of color graphics quickly and efficiently.

Take a look at these features — then choose the HP 7550 graphics plotter for **all** your plotting needs.

MORE RESULTS — LESS EFFORT
Increase efficiency with the HP 7550. Rather than load paper by hand, use the HP 7550's automatic sheet feeder to load each sheet for you! The large tray capacity lets you complete up to 150 pages of graphs before adding more paper.

Because the HP 7550 is designed to provide superior plotting performance, each chart is completed quickly. With 6 g's pen acceleration and 80 cm/s (31.5 in./s) maximum velocity, your charts are ready when you need them. For comparison, a plotter with 2 g's acceleration completes a typical area fill chart in 4½ minutes; but with the HP 7550's 6 g's acceleration, the same chart is completed in 2½ minutes.

Place the easy-to-load 8-pen carousel in the plotter and a built-in microprocessor senses the pen type — paper, transparency, roller ball, or liquid-ink drafting. The HP 7550 then automatically adjusts pen force and speed based on your pen selection. When returned to the carousel, the pen is automatically capped so inks stay fresh and pens last longer.

With a liquid crystal display built into its front panel, the HP 7550 almost lets you "talk" with your plotter. The display, used together with function keys, lets you either respond to program messages or choose plotter states and functions when communicating with your computer.

QUALITY GRAPHICS EVERY TIME
You can be sure your graphics will be the highest quality possible when they're made on a Hewlett-Packard 7550. With an addressable resolution of 0.025 mm (0.001 in.) and a mechanical resolution of 0.006 mm (0.00025 in.), curved lines are smooth, not jagged, and straight lines are consistently straight. The exceptional repeatability of 0.1 mm (0.004 in.) assures you that edges align in bar or pie charts, that inks on area fills are applied evenly and smoothly, and that characters are smooth and completely formed.

Unlike other plotters in its price range, the HP 7550 keeps pen velocity and acceleration constant for both diagonal and axial pen movements. Thus, lines are drawn with the same visual quality and speed — regardless of direction.

TWO CHARACTER FONTS
Each of the HP 7550's 20 character sets comes in two fonts: stick and arc-based, as shown in the example below. The stick font provides simple high resolution characters with equal spacing — fast and efficient for working drawings. The arc-based characters with proportional spacing provide superior resolution and readability, similar to LEROY* lettering.

COLOR GRAPHICS IN QUANTITY
With the HP 7550 it's now practical to make multiple copies of color graphics for your reports and presentations. Using the built-in automatic sheet feed for cut paper sheets, you can quickly produce up to 150 pages of graphs before refilling the paper tray. And you don't need to spend your time operating the plotter — it's automatic! Use the replot feature to get up to 99 copies of each graph — without re-running the program.

ADVANCED INTELLIGENCE
The HP 7550 features some of the most advanced intelligence available in a graphics plotter today. Built-in intelligence takes the burden off your computer or controller. With its powerful MC68000 16-bit microprocessor, the HP 7550 can plot in any of 20 character sets including ISO European and Katakana, read the carousel type in the plotter and automatically set pen speed and force, rotate the chart 90 degrees to fit either horizontal or vertical formats, sense media size and set correct plot parameters, and digitize point co-ordinates.

SOFTWARE COMPATIBLE
Because the Hewlett-Packard Graphics Language (HP-GL) internal instruction set is incorporated into the HP 7550, your software investment is protected. The HP-GL programs you are currently using can also be used for the HP 7550 with little or no modification. A large base of HP-GL graphics software is widely available.

Programming is easy with HP-GL — you use simple instructions to perform a variety of operations like drawing arcs and circles, selecting pen velocity, and labeling. For example, with just three HP-GL instructions you can define and color an area of your chart.

COMPLETE MEDIA SUPPORT
Hewlett-Packard pens and writing media meet the same strict quality standards as HP's electronics products. Paper and pens act as a system to give you the highest quality results possible. With the HP 7550 you can choose fiber tip for either paper or transparencies, roller ball, or liquid-ink drafting pens. And your choices of media include plain paper, transparency film, polyester film, and vellum.

DUAL INTERFACE
The HP 7550 has one IEEE 488-1978 (HP-IB) connector and two RS-232-C/CCITT V.24 connectors on the back panel. The two RS-232-C connectors provide flexibility to adapt to your needs. Using only one computer port, you can connect the HP 7550 either directly (hardwire or direct) to your computer, or in series (eavesdrop) between your computer and terminal. The HP-IB connector lets you use the HP 7550 with other computer systems offering this interface. Because the HP 7550 has these built-in interfaces, it's easy to share among many workstations and to adapt to your changing system requirements.

* LEROY is a registered trademark of Keuffel & Esser Company.

Each of the HP 7550's 20 character sets comes in two fonts:

```
For faster labeling, use the 'STICK'
font with fixed spacing between characters

For highest quality lettering, use the 'ARC'
font with proportional spacing between characters
```

HP 2622 Block/Forms Mode Terminal Emulation for Series 200

The new HP 98790A Series 200 Block/Forms Mode Terminal Emulator emulates the HP 2622A Terminal's block and forms mode features, as well as its line and character mode. It also provides bidirectional file transfer, screen dump to a printer, and Print All On. The emulator does not support HP 2622P operations. The Block/Forms Mode Terminal Emulator has been fully tested with the HP 3000, particularly with HPMail, HPSlate, and VIEW/3000 applications.

Announcing the HP 9000 Series 200 HP-UX products

Three new computers have been added to the HP 9000 family. A new, high-performance CPU board option is available for the Series 200 Models 200, 236, and 236C. It accommodates the HP-UX operating system which will further enhance our line of 16/32 bit, MC68000-based workstations.

HARDWARE
It's simple! Take the CPU board out of current Model 220 (HP 9920) or Model 236 (HP 9836A/C), replace it with a new CPU board, and you have a Series 200 product capable of running HP-UX.

The new CPU board includes a 12.5 MHz 68000 processor (previously, only a 8 MHz 68000 was available). In addition, the board contains 16Kb of cache memory and the memory management hardware necessary for multi-user applications. Note that there is no user RAM present on the new CPU board.

The base level hardware order numbers are HP 9920U, 9836U, and 9836CU. They designate the new-CPU versions of the HP 9920A, 9836A, and 9836C respectively. As such, they include no software or extra RAM.

The Series 200 HP-UX products are more expensive than the corresponding Series 200 products. This is primarily due to the increased cost of the state-of-the-art, higher performance parts on the CPU board (for example, 45-nsec static RAM parts are used for cache).

SYSTEMS
A bundled T system is generally better for your first system. A T system consists of an HP 9XXXU computer, 1 Mb RAM, and the complete HP-UX software product.

For multiple Series 200 HP-UX installations, the second and successive systems should be ordered as U products along with RAM and the "additional copy" HP-UX software product. This will result in savings, compared with the price of the corresponding T product.

UPGRADES
A CPU upgrade kit is available to allow installed Series 200 products to run HP-UX. The HP 98143U provides the higher-performance CPU board. A special bundled upgrade kit, P/N 98143T, offers an economical way to upgrade an S system to a T system. The 98143T includes a high performance CPU board, 0.5Mb RAM, and a DMA board.

Announcing Context MBA™ on the HP 150

Context MBA, the integrated software package offering spreadsheet, graphics, word processing, data management and telecommunications, is now available and shipping on the new HP 150. MBA is the highest performance spreadsheet available on the HP 150. It provides the advanced user a sophisticated tool to solve complex problems and communicate the results.

HP MBA version 2.3 on the HP 150 is even improved over the highly successful version that you may be familiar with on the HP Series 200. The 150 implementation makes use of HP Touch. MBA now allows the user to position the cursor in the worksheet by touching the screen. The cursor can even be moved from one window to another by touching the desired window. In addition, the user now has a selection of softkeys available both on the screen and on the keyboard.

MBA will operate on a standard 150 with 256K bytes of RAM. However, for optimum performance and greatest user satisfaction, we recommend a 512K-byte configuration. Not only does this give the user a substantially larger workspace; it greatly improves the performance of the MBA. This is because more of the MBA code can reside in memory, greatly reducing the amount of code swapping necessary from the disc.

Context MBA is a registered trademark of Context Management Systems.

NEW ROLL-FEED PLOTTER AT LOWER PRICES

On February 1, HP introduced a new roll-feed drafting plotter, the HP 7586B. At the same time we announced exciting enhancements and reduced prices on our current drafting plotter models, the HP 7580B and HP 7585B.

HP 2930 family replaces HP 2631B/G printers

HP 2934A CORRESPONDENCE PRINTING
The new HP 2934A Office Printer offers **correspondence-quality** printing. What is correspondence-quality printing?

The 2934A is a dot matrix printer. Characters are formed using dots. The smaller the dots and the more they are overlapped, the more the characters can be made to look like they were typed with a typewriter.

In order to get more overlap of the dots, the 2934A must slow down from its normal 200 cps. It produces correspondence-quality printing at 67 or 40 cps, the slower speed providing the best print quality.

For many applications, the 2934A's correspondence print quality will be adequate to replace typewriters or daisywheel printers.

Holguin's drafting software gets "personal" with the HP 150

Holguin & Associates, HP's leading third party software supplier, recently announced support of the HP 150 personal computer with their CEADS-CADD Multi-User Drafting package.

The HP 150 will not only help CEADS-CADD users with business management and administrative needs, it will double as a production drafting terminal for CEADS-CADD as well.

USING THE HP 150 AS A PRODUCTION DRAFTING TOOL
Initially, CEADS-CADD will support the HP 150 as an HP 2623A-compatible graphics terminal. Although the use of the HP 150 touch screen is currently limited to the use of its softkeys, near-future enhancements of CEADS-CADD will take full advantage of this outstanding user interface.

THE HP 150 AS A BUSINESS MANAGEMENT TOOL
As a business management tool, the HP 150 can be used as a stand-alone personal computer utilizing a variety of available business application software such as:

- word processing for creating documents
- spreadsheets for analyzing and planning
- data base management to track and manage information
- project management to manage large projects
- finance and accounting to keep the books.

Each drafting department or company can now have their own small business system to support their needs, without the added costs.

The combination of the CEADS-CADD Multi-User Drafting System and the HP 150 is a unique addition to the computer aided drafting market-place.

HP at your service?

I was disappointed recently when talking to a regular customer of H.P., when he commented to me that it was difficult finding the right people at H.P. to get the answers to questions. He compared H.P. to the public service where one section does not know anything about what the other is doing (his comment, not mine!). He also commented that he got different answers depending on whom he spoke to. Well, as I said, I was obviously disappointed and embarrassed about this, and we at H.P. have some improvements to make if we want to be number one for customer service, and we do!

As the representative of H.P. on the Desktop Computer User's Group Committee I consider it my responsibility to be the open door into H.P. for all members of the user's group. Any lack of service, responsiveness and information, etc., from H.P. is my responsibility. I can't guarantee that I will know all the answers; if I don't know I will try to find the answers or put you in contact with the person whom you should be talking to. I also can't guarantee that I'll be available if you 'phone; if you find it impossible to contact me at H.P., my home number is (059) 68 3324.

People often comment to me that H.P. has high quality products, and whilst I feel proud, I cannot accept any credit for this since they are obviously designed and manufactured elsewhere. However, the service which we provide customers is one part of our product which is supplied locally, and it is our responsibility to ensure that that part of our product is of at least equal quality with the range of H.P. hardware and software. To all members of the Desktop User's Group, please contact me if you need help.

PHILIP GREETHAM

RECENT DESKTOP COMPUTER USER GROUP MEETINGS

Graphics and Data Communications supported by equipment demonstrations have featured at the last two meetings of H.P.D.C.U.G.V. This continued our policy of presenting members with a range of topics and speakers spanning areas of interest to desktop computer users.

Computer Aided Drafting was the subject of the meeting held on 9th November. It began with a presentation by Bill Pizzey of E.C.S. who supported his talk with an excellent set of slides and a video tape which showed the tremendous power of the 9836 desktop computer in this application. This was followed by a presentation from John Klein of H.P. on the HP2700 showing its great ability in preparing graphic displays for management and advertising applications. From the extent and number of questions put to both speakers it was obvious that we had found a popular topic.

At the next meeting on 23rd February Mike Canvin of Datacraft broadly outlined the basic problems of data communications and how these are addressed by various types of modems. Mike supported his talk with a number of slides and fielded a wide range of questions which extended well into the refreshment break.

Finally we were treated to a demonstration of the new HP150 Desktop Computer by Stefan Zieman of H.P. A great deal of interest was aroused by this very smart machine and it was with considerable difficulty that we cleared the hall at the end of the night to lock up!

B. T. O'SHANNASSY

A Mailing List for the HP Desktop Computer Users Group of Victoria

IAN G. McWILLIAM,
Swinburne Institute of Technology.

Soon after the Desktop Computer Users Group was established in Melbourne it became evident that an alphabetical list of members was required, and that this should be suitable for mailing purposes. Furthermore, it should be easy to insert or delete data from this list, to make alterations and to change addresses.

Initially the names and addresses were typed on a sheet of paper, cut out and inserted into Hagner hingeless stamp sheets. This provided an alphabetical listing but was inconvenient to update. Nevertheless it could be copied onto Avery self-adhesive labels, using a standard plain paper copying machine. These labels, on an A4 size sheet, can be readily transferred to envelopes and are available in various sizes. I chose a sheet containing 24 labels, each label being 35 × 70 mm. The size of the label was determined by the amount of space required for the name and address, together with a reasonable clearance between each entry.

The next step was to write a program so that the mailing list could be put onto an HP9825A computer and printed out on an EPSON MX-80F/T dot-matrix printer. The constraints imposed were:

    Computer storage limited to 23228 bytes,
    A maximum of 25 characters per line for names and addresses,
    A maximum of 4 address lines per entry, and
    Data to be stored on a tape cartridge.

Initially the system was based on a file size of 24 names and addresses,

equivalent to one page of labels, with the data for each page being individually transferred to and from the tape cartridge. This allowed for about 1000 members, but even with a small membership list it involved a significant number of file transfers to update the list. At the present time there are 75 members, and the system has now been changed so that it is based on a single file of up to 120 members.

To simplify the alphabetical ordering of names, they are entered as
    Name, I.N. — Mr (where — indicates a space)
or
    NAME OF COMPANY (no comma)
or
    NAME 1, NAME 2 and NAME 3 (containing AND)

When printed out, names containing a comma but no AND are restructured to give
    Mr — I.N. — Name

The remainder are not changed on printout.

The data is stored as an array A$[600,25] which consists of up to 120 names and addresses. This occupies a total of 16,206 bytes, including 1,206 bytes of overhead. To add data, the following procedure is used (see also Figure 1):

(i)   New names and addresses, up to 10 at a time, are entered into array B$ and the names are also placed into array C$.

(ii)   The names in C$ are placed in alphabetical order into D$.

(iii)   At the same time, the array locations of the names in C$ are placed into array D to specify the address positions in B$.

(iv)   A new list is then generated by comparing the last entry in A$ with the last entry in D$. If there are N names in A$ and 3 names in the new list D$, the last name on the combined list will be entered at the A$ array location $5*(N+2)+1$. The appropriate address will then be obtained either from A$ or from B$, depending on whether the name came from A$ or from D$. This procedure is repeated until all of the names in D$ and their addresses have been transferred to A$.

In order to change data in the file, the page number, block number (24 per page) and line number are specified and the appropriate line is shown in the 32-character display. The new line is then entered from the keyboard. This line replaces the old line in the A$ array when the CONTINUE key is pressed. When all changes have been completed, the old file on the tape is replaced by the new file.

To delete data, the appropriate page and block number are selected, and the name is shown in the display. The name and address will then be removed only if the y key is pressed. This removal is effected by successfully replacing these A$ array elements by those further along the array.

The options at present offered by this program are:
1. Print the file
2. Change data in the file
3. Delete data
4. Add data
5. Copy the file
6. Set up an initial file

The program itself occupies about 3500 bytes. With the space used by the arrays in the program, including those for the printer output, the total space occupied is approximately 22,000 bytes.

On the HP9825A, as with some of the more recent HP desktop computers, the entire file could have been stored as a single string. This would provide more efficient packing of the main file. Although string searching would then be required to extract, delete or add information, this is relatively fast and the subsequent operations would be simple to carry out. However, the need to generate sub-strings in manipulating the file will approximately double the computer storage required, and this would offset the space saved by using a single string.
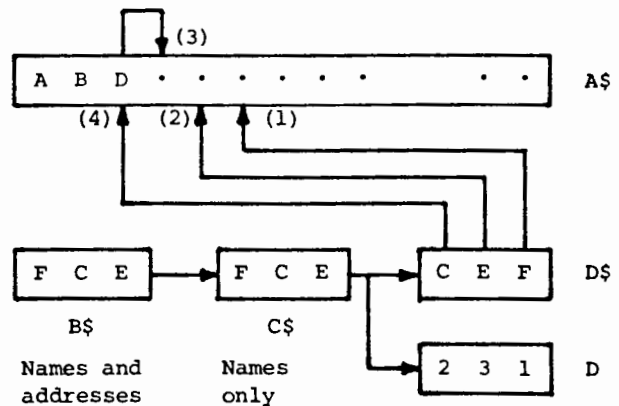

Figure 1: Addition of data to the file A$, showing (in brackets) the sequence of data transfer for names C to F.

HP 150 On-Line Help

One of the features which I found most useful when beginning to use the HP 1000 A series was the easy-to-use and friendly on-line help facility. Likewise, when recently I purchased my HP 150, not being familiar with MS-DOS, I was looking for a similar feature to save me reaching for the manual frequently. To this end I ended up writing a little utility to do just that. I did not have the disk space to make full page manual entry for each command, instead I opted for printing a brief synopsis — a line or two.

This feature is very easily done with MS-DOS, which has the capability of using batch files. With MS-DOS you can put commands into a batch file and simply execute the sequence of commands by typing in the name of the batch file. Hence the name of my simple batch file is "HELP" and on typing 'HELP' I get a list of the commands — one screenful at a time. I had in mind to enhance it to include a summary of "Microsoft"[R] BASIC commands. Typing "HELP MS-DOS" or "HELP BASIC" would pass the appropriate parameter to the command file and display the relevant file of commands. If anyone gets around to doing it before I do, please pass on a copy to me! Anyway, here are the files to use if you wish.

FILE: "HELP.BAT"

```
ECHO OFF                    ! Don't echo cmds to screen
CLS                         ! Clear the screen
TYPE\BIN\MS-DOS.DOC ! MORE  ! List the file of cmds
                            ! one screenful at a time
```

N.B. The comments are not part of the file, but are for this article only.

FILE: "MS-DOS.DOC"

SUMMARY OF MS-DOS COMMANDS

BREAK	Sets CONTROL-C check
CHDIR	Changes directory of the default or designated drive and checks for consistency
CLS	Clears screen
COPY	Copies file(s) specified
CTTY	Changes console TTY
DATE	Displays and sets date
DIR	Lists requested directory entries

DISKCOPY	Copies disks
EXE2BIN	Converts executable files to binary format
EXIT	Exits command and return to lower level
FIND	Searches for a constant string of text
FORMAT	Formats a disk to receive MS-DOS files
MKDIR	Makes a directory (MD)
MORE	Displays current output one screen at a time
PATH	Sets a command search path
PRINT	Background print feature
PROMPT	Designates command prompt
RECOVER	Recovers a bad disk
REM	Displays a comment in a batch file
REN	Renames first file as second file (RENAME)
RMDIR	Removes a directory (RD)
SET	Sets one string value to another
SORT	Sorts data alphabetically, forward or backwards
SYS	Transfer MS-DOS system files to the drive specified
TIME	Displays and sets time
TYPE	Displays the contents of the file specified
VER	Prints MS-DOS version number
VERIFY	Verifies writes to disk
VOL	Prints volume identification

BATCH COMMANDS:

ECHO	Turns batch files echo feature on/off
FOR	Batch command extension
GOTO	Batch command extension
IF	Batch command extension
PAUSE	Pauses for input in a batch file
SHIFT	Increments number of replacable parameters in batch process

The above file of commands was taken straight from the MS-DOS User's Guide.

I would certainly like to hear from anyone else who has any useful little snippets for the HP150.

PHIL GREETHAM,
H.P. Melbourne

HP86B BUGS
(Found the hard way)

1. SLASHES IN IMAGE STATEMENTS
Don't use slashes either in IMAGE statements or in the image segment of the OUTPUT statement! The HP86 seizes solid and in some cases secures lines randomly!
There are other ways to achieve linefeeds within an output statement, for example, define C$=CHR$(13)&CHR$(10). (Carriage-return, Linefeed). Then use it wherever needed.
E.G. OUTPUT P USING "K,4/,K";X$,Y$ could be rewritten:
   OUTPUT P; X$;C$;C$;C$;C$;Y$

2. AUTO-POWERUP
After power is re-applied to an HP86B + 9121D disc drive, autostart is NOT achieved. This is apparently because the 9121D takes too long with its self-check. Of course it is OK with the 82901 drives. Apart from changing the disc drive, Hewlett-Packard have come up with NO FIX as yet. I hope they do, as (unwarned), I have installed an unattended system that depends ENTIRELY upon autostart! HELP!

3. RENUMBERING
John Hedger (03) 350 1766, reported at a recent Melbourne desktop user group meeting that:
   Renumbering large programs after deleting sizeable slabs of code leads to 'missing line' and other errors that are quite untrue. The FIX is to delete SMALLER slabs at a time, renumbering in between.

CHRIS SIMPSON,
(03) 859 6643
4/5/84

Focus 1000

HP 1000 User Group News (Vic.)

Welcome to 1984. We didn't expire after the IMAGE/1000 Workshop last year, but your committee has been a little slack over the festive season.

We plan to have regular meetings this year. The idea is to meet at HP Blackburn on the second Wednesday of every odd numbered month, except January. All members and prospective members are welcome.

The Annual General Meeting was held on March 23 at HP Blackburn. All existing committee members were re-elected without any changes. The General Business included a discussion of possible topics for User Group Workshops this year, with RTE-A and various aspects of Communications raising the most interest. There was also interest in arranging user group visits to Philip Morris and C.S.I.R.O. in Parkville and the Australian National Animal Health Laboratories in Geelong. The General Business was followed by a presentation by Joan Dillon, Product Sales Manager of CCS, U.S.A. The meeting concluded with refreshments, and those who stayed the distance went on to a nearby B.Y.O. for a casual meal.

There are currently 62 members of the HP1000 Users Group in Victoria. A list of these members will be distributed to all current members of the group.

For details about the Users Group or its activities, please contact: John Gwyther (President) (03) 859 9487 or Chris Emery (Secretary), (03) 667 2328.

MANAGEMENT VIEWPOINT

CONFERENCE MANAGEMENT ROUNDTABLE

There were more questions asked than time to answer them at the management roundtable session of the recent HP 1000 International Users Group Conference in Fort Worth, Texas.

In addition to general questions about HP's future strategy, support and product developments (which were answered at the conference), many users had more detailed questions about HP products and services. These questions are answered here.

The members of the HP management panel were Dave Carver, support marketing manager/CSD; Marc Hoff, worldwide SEO manager/CSD; Greg Gillen, marketing manager/DSD; Peter Rosenbladt, R&D Manager/DSD; Jan Stambaugh, manager of training development/PSD; Dave Yewell, technical sales manager/BDG; and Steve Zalewski, manager information central support/IND.

Does HP plan to move UNIX to the A Series?
No. In the past we considered several alternatives aimed at moving UNIX to the A Series. The decision not to implement UNIX was based on our concern in two areas. The A Series was designed around the HP 1000 instruction set and RTE. UNIX was designed around a different set of features. Thus, many A Series capabilities would not be utilized by UNIX. An example is the UNIX I/O system design around character processing. The A Series I/O system is an intelligent I/O system that transfers data via DMA in blocks. Other examples involve memory management, process management, and other areas.

The second area of concern was the lack of UNIX real-time capabilities. Since our customers expect a real-time operating system, we knew we would not only have to modify UNIX for A Series hardware but also for real-time—an effort we considered unjustified. Therefore, we will focus our efforts on developing optimal RTE-compatibility features for HP-RT/UX, as well as more sophisticated and useful migration utilities and analysis packages.

Can the HP 1000 A600 be linked with a Tektronix 4112 Graphics CRT?
Connection of a Tektronix 4112 Graphics CRT to an A600 would have to be done by our Specials Engineering Group, and at this time they have received no requests to support the 4112. However, we do offer special products for support of Tektronix 4010-16 and 4025-27 terminals.

Is there a link from the A Series to the IBM 34?
Our HP 1000 to IBM data communications products are designed to support communication with IBM mainframes. Keeping this in mind, we now have MRJE for A Series, and RJE is under development. It's possible that RJE on the A Series will allow file transfer with RJE on the IBM S/34; however, we do not plan to support this configuration.

Will the 2647A/F be supported on the A Series as a system console?
In the near future we will add system console support to the 12040B multiplexer; at that time we will also add support of the 2647F/2648A.

Is HP planning to build a differential I/O card for the A Series?
A differential I/O card is available from the Specials Engineering Group. Ask your sales representative to send you information and a quote on the 93699G card.

Will HP do an A Series data communications card to IBM 4341 and VAX via Ethernet?
Yes, HP will be introducing an A Series IEEE 802.3/Ethernet data communications interface card. However, the customer must write his own Ethernet software to communicate with a VAX.

We are not aware of IBM support of Ethernet or IEEE 802.3 on the 4341.

Will the MAC-ICD problem on the A Series be resolved?
Yes, MAC support through the 12745D MAC to HP-IB converter will be supported as of B.84 Software Update. Performance will equal that of "H" disc.

Will HP support 7470 and 7475 over RS-232 with DGL Graphics 1000-II DGL?
New versions of Device-independent Graphics Library (DGL) and Advanced Graphics Package (AGP) will be introduced in 1984. These new versions will be added to the GRAPHICS/1000-II family and will be supported on RTE-A, RTE-A with VC + and RTE-6/VM. They will make use of the new hierarchical file system and provide code- and data-separated graphics tools for VC + systems. The results will be a set of graphics tools that fit smoothly with the new RTE systems and in some cases perform better or use less resources via shared code.

The new version of AGP will differ only slightly in calls that reference the file system—JDINT, JSDF, JDFNT, and so forth. There will be no difference in DGL calls. For those customers on support services, the upgrade to the new products will be free. Current graphics customers without support services will be able to upgrade at a discount.

When the new versions of DGL and AGP are introduced, the current versions will be mature, which means they will continue to be available and supported. But new device handlers will not be added to these products. All new handlers will be added to the new versions of AGP and DGL.

Currently, the 7470A plotter is supported via HP-IB. 7475A plotter HP-IB support is planned for the new versions of DGL and AGP. We are also investigating the support of the 7470A, 7475A, and the 722X plotters as standalone RS-232 devices. Such support would also be available with the new versions of graphics.

MANAGEMENT VIEWPOINT

What plans does HP have to provide multidrop communications for the 2250, with the 2250 answering only when spoken to?
HP has no plans to change the current offering of HP-IB or DS. As one of the 802 standards is implemented within HP, we will evaluate it as an alternate communication scheme for many of our products, which would include the 2250.

At present, Pascal requires EMA variables to be dynamic variables in the heap. But when large programs are ported in from other systems, the large arrays are usually static, not dynamic. When will Pascal allow declaration of static variables in EMA?
This is being considered as an addition to the Pascal compiler in late 1984 or early 1985.

Does the FORTRAN 77 compiler add an option for subscript-bound checking and subroutine call checking (type and number of parameters) like Pascal does?
While these enhancements would be nice, we have no plans at this time to add these features to the compiler.

Does FTN4X compiled code reference any subroutines or entry points such that it would not load with RTE-M and its FTN4 library? (We would like to upgrade to RTE-6/VM, yet still do program development on our primary system).
HP has not tested the above compatibility, and we are concerned that future revisions may not support this capability even if it existed today.

Any hope in reducing IOP chip costs and increasing speed for the independent designer?
As a special product from our Specials Engineering group, HP offers the IOP chip and carrier. Documentation and artwork for a breadboard is available as well. Contact your sales representative for a quote on 93579P.

When can we look forward to increased PROM storage card capacity to match the RAM memory cards?
The 12008A PROM storage card has 64 Kb capacity. There are no plans to modify this card.

How about microprogramming parts of the O/S, like EXEC calls, memory protect interrupts, table searches, and so forth, and the Pascal runtime library; and also reducing the size of the O/S?
This has already been done for many low-level O/S functions. However, firmware space is at a premium on the A Series processors, which makes it hard to add more firmware routines without increasing the cost of the processors. This suggestion has been considered and will continue to be as microprogram space becomes available. For now, we provide user microprogramming on the A700 and A900 to allow you to speed up an application in the critical areas.

A900 Pascal w/CDS—when? We need code- and data-separated runtime libraries for the FORTRAN formatter file management system and Pascal runtime libraries. These would be loaded into a reserved partition and executed as shared code, thus eliminating the 5 to 10K words of the program that now clutter up the data portion. We would prefer to do this with a structured language like Pascal or C.
The Pascal compiler will emit CDS code and provide CDS libraries at the A.84 PCO cycle. The new file system is currently supplied as both CDS and non-CDS relocatables. The CDS versions are in the library called $CDS.
Since the FORTRAN formatter routines are small and in assembly language, there are currently no plans to supply them as CDS.

How does one change the size of the system cartridge if there is only one CS-80 disc on an RTE-A.1 system? The method

described in the manual (page 3-9) does not seem feasible (our SE cannot see how it works either).
The system LU (LU 16) size cannot be changed without a second disc. This is because there is only one bootable surface on CS-80 drives. The manual will be reviewed.

Is a faster replacement for MLLDR in the works?
Not on RTE-6. However, LINK is available on RTE-A.

In a multi-CPU environment does HP recommend some utilities to install systems (on CS-80 discs) without having to run SWTCH on each host system?
PSAVE is the best solution.

When will HP support block-mode transfers across the DS/1000-DS/3000 link?
We already provide V/3000-compatibility from a terminal connected to an E/F Series through a multiplexer card, across the DS link to an HP 3000. This is not full generic block-mode support, but does support V/3000 applications. (Note: HPMAIL and HPDRAW are not V/3000 applications and are not supported.) V/3000 support for terminals connected to E/F Series through a BACI card is under development; however, there are no plans to support V/3000 applications on the multipoint interface.
For the A Series, V/3000 support is under development for terminals connected via a multiplexer card. Follow-up projects will include support on the asynchronous serial interface card.

When I/O mapping a printer to or from a remote node, the first listing is limited to 80 characters and the first character is stripped.
There is no limitation in I/O mapping that confines a transfer to 80 characters. I/O mapping works via DEXEC calls and supports transfers of up to 512 words in length. Incorrect configuration of the map or the printer could cause truncation of the first

9

MANAGEMENT VIEWPOINT

record. The printer also treats the first character as carriage control. The user application should include a carriage control character in column 1, or the printer should be sent a control function to suppress carriage control. This function is printer-dependent.

Is anything being done to alleviate the problems caused by a three-page X.25 driver on MEF systems?
There are no plans at this time to rewrite the X.25 driver for the E/F Series. Users who require DS access from programs larger than 27 pages could use class I/O to communicate with another program that performs the DS calls.

Why is DS slowing down? It used to run at 2M baud (12889), then 650K baud

(12771), and now 230K baud. Why can't the 12889 be used in 1000-1000 DS?
While it is true that the line speed of DS is slowing down, the actual performance is better. The 12889A HSI card did provide up to 60 Kb/sec of throughput between the 1000 MEF and the 3000 Series III, but at the cost of 50 percent overhead on the 1000 and 100 percent overhead on the 3000. The 12771 WASP card did offer higher line speed, but the actual throughput between user programs was less (18 Kb/sec versus 20 Kb/sec for the PSI).

When will the MUX support XON, XOFF in both directions?
XON/XOFF for one direction (to receive it from the device) has been implemented on RTE-6 and RTE-A. We are currently

looking into the feasibility of implementing XON/XOFF for both directions on A Series.

When will the new file be available on RTE-6?
It has already been submitted for the current PCO process.

Will HP-IB tape drives be supported with RTE-6?
It is not in our current plan to develop this.

Sure would be nice if you could incorporate a system log file into RTE-6.
We agree. We will look into the feasibility of doing this.

## HP Technical Computer User Group (N.S.W.)

Our next general meeting will be held on Wednesday the 6th of June at 4.00 pm.

Following is a list of the User Group Committee members:

President: David Davis, (02) 43 5150
Secretary: John O'Brien, (02) 359 3217
Treasurer: Neil Crosby, (02) 498 4566
Committee: Eddie Hanhan (02) 359 3650
Dr. Warrick Summers, (049) 68 2311
John Quigley, (02) 887 1611

PUZZLE SOLUTION

Below is the solution to the puzzle published in the October/November issue of Crosstalk.

"Oh no, I just accidentally purged The Meaning Of Life."
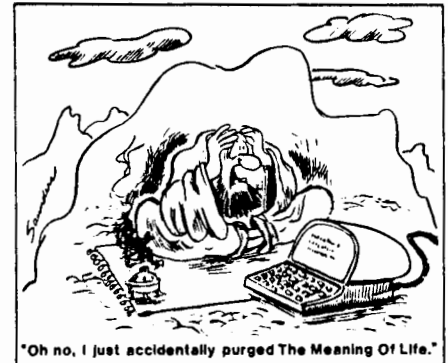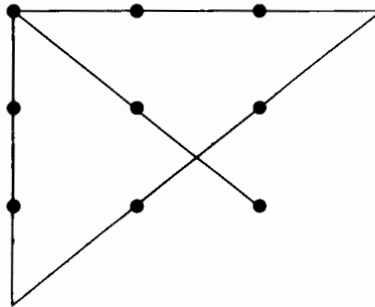
SPECIFICATIONS FOR SUBMISSION OF ARTICLES AND ADVERTISEMENTS

All material for Crosstalk should be sent to one of the addresses listed at right, from where it will be forwarded to the co-ordinator for publication. Publication dates are subject to receipt of sufficient material. For specific details contact Glenda Patterson on (03) 895 2576.
ARTICLES: Articles should be typed with any diagrams and program listings in camera-ready form (Author's name, address and phone number should be included).
ADVERTISEMENTS: Display ads. should be in camera-ready artwork form. The printer may be instructed to layout ordinary typeface ads.
CURRENT ADVERTISING RATES:
Full page — $250
Half page — $125
Column/cm — $4

There is a 20% discount on these rates for regular advertisers. Classified ads. are free for user group members, and $10 each for non-members.

Advertisers will be billed upon receipt of ad. The user groups reserve the right to change rates, limit space availability and reject advertising which is deemed inappropriate.

ADDRESSES FOR SUBMISSION OF ARTICLES AND ADVERTISEMENTS:
The Editor,
HP Technical Computer Users Group, N.S.W.
Box 3060 GPO,
Sydney, 2001.
N.S.W.

The Editor,
HP1000 Users Group (Vic.)
P.O. Box 132,
Mt. Waverley, 3149
Vic.

H.P.D.C.U.G.V.
Mr Bernie O'Shannessy,
Arlec,
30-32 Lexton Road,
Box Hill, 3128, Vic.

HP Desktop Users Group (N.S.W.)
Dr. R. W. Harris,
C/o C.S.I.R.O. Division of Mineral Physics,
PMB 7, Sutherland, N.S.W. 2232.

THE HP 1000 GURU

by Bill Hassell

How can I keep the terminal screen from burning?

One useful technique to prevent burning (resulting from long periods with the same image on the CRT) is to clear the screen when logging out. Since MTM and session monitor in RTE-IV and RTE-6 do not have a GOODBYE file (similar to the HELLO file), a simple procedure can accomplish this goal.

First, it is necessary to create a transfer file, possibly named EX. The file would look like:

```
:RU,CLSCR
:EX,RP
```

This assumes that the program CLSCR is RP'd or SYSGEN'd into the operating system. CLSCR is a simple program that demonstrates the use of the DTACH call. You want to clear the screen after the logout message is finished, yet any program run from session will be removed at logout. So use DTACH to remove the program from session.

Another call is XLUEX. This is basically an EXEC call that can reach every LU in the system. A normal EXEC call is limited to LUS 0 through 63; thus, the ACCTS system must map system LUs down to the range 0-63. XLUEX uses a 32-bit control word instead of the normal 16 bits. The program must wait while FMGR terminates and the accounting information is displayed, which usually takes about two to four seconds. This completes the program requirements. Now it is necessary to ensure the screen can be cleared, which is accomplished with the array CRTCLR (illustrated in Figure 1).

This program can be run from anywhere in the system and the LU number (to be cleared) can be supplied in the runstring. Note that graphics memory is also cleared and the softkey labels (such as on the 262x or 2647 terminals) are turned off.

Is there a good way to use flags and switches in FORTRAN?

In FORTRAN, a lot of code is used to test flagged conditions, and using integers to set and test flags is particularly cumbersome. Most programmers know LOGICAL declarations, but many fail to use them.

Figure 1: Using Array CRTCLR to Ensure Screen Can be Cleared

```
FTN4,L
      PROGRAM CLSCR(3,99),Clear screen after log-off is done
C
      INTEGER*2 CRTCLR(8)
      INTEGER*2 LUXTND(2)
      INTEGER*2 TEMP   (5)
      INTEGER*2 LUTRUE
      LOGICAL*2 INSESN
C
      DATA CRTCLR/15532B,    ! ESC Z (dispfunct. off)
     +            15555B,    ! ESC m (memlock off)
     +            15446B,    ! ESC &
     +            65100B,    ! j      (turn off skeys)
     +            15452B,    ! ESC *
     +            2HdA,      ! dA     (clear graphics)
     +            15510B     ! ESC H (home cursor)
     +            15512B/    ! ESC J (clear memory)
      DATA LUXTND/0,0/
C
C Check if an LU was supplied   (RU,CLSCR, <LU> )
C
      CALL RMPAR(TEMP)
      LUTRUE = TEMP(2)
      IF(LUTRUE.LE.O) CALL LOGLU(LUTRUE)
      INSESN = LUTRUE.LT.O
      LUTRUE = IABS(LUTRUE)
C
C Construct the XLUEX LU word
C
      LUXTND(1) = LUTRUE
      IF(INSESN) CALL DTACH(TEMP)
C
C After detaching, wait
C
      CALL EXEC(12,0,2,0,-4) ! 4 second wait
      IF(INSESN) THEN
         CALL XLUEX(2,LUXTND,CRTCLR,8)
      ELSE
         CALL EXEC(2,LUTRUE,CRTCLR,8)
      ENDIF
      CALL EXEC(6)
      END
```

Although a LOGICAL item and an INTEGER variable occupy the same amount of space, the logical code is shorter and more readable. For example:

```
JFLAG = 0
IF(J.GT.K) JFLAG = 1
    :
    :
IF(JFLAG.NE.1) ....task..
```

In this case, JFLAG has only two states and, therefore, performs a LOGICAL function. Now look at the same code using LOGICALS:

```
LOGICAL FLAG
    :
    :
FLAG = J.GT.K
    :
    :
IF(FLAG) ...task...
```

THE HP 1000 GURU

Figure 2: Example of More Readable Code When Multiple Flags Used and Tested

```
LOGICAL OVRTMP,LOWTMP,HIPRES,LOPRES,ALARM
  :
  :
OVRTMP = TEMP.GT.42.784
LOWTMP = TEMP.LT.-3.90577
HIPRES = PRESS.GT.200.0
LOPRES = PRESS.LT.40.0
ALARM  = HITEMP.OR.HIPRES
  :
  :
IF(OVRTMP.AND.HIPRES) ...task...
IF(.NOT.OVRTMP.AND..NOT.LOPRES) ...task...
IF(ALARM.AND..NOT.HIPRESS) ...task...
```

Figure 3: Up-to-date List of the $OPSY Values

```
  1  =  DOS  (or any positive value = DOS)
 -1  =  RTE3
 -3  =  RTE2
 -5  =  RTEM3
 -7  =  RTEM1
 -9  =  RTE4
-15  =  RTEM2
-17  =  RTE-6
-29  =  RTE-XL
-31  =  RTE-L
-45  =  RTE.A.1
-37  =  RTE.A
```

In fact, code using LOGICAL is even more readable when multiple flags are used and tested (see Figure 2).

Certainly seems logical, doesn't it?

Is there a way to determine operating system revisions programmatically?

HP software continually improves and revises RTE, and in consequence, user software must know which revision it's running on. There are three entry points to determine this. The first is $OPSY and contains the operating system ID. Figure 3 is an up-to-date list of the $OPSY values.

Now that we know the opsystem, we can determine the opsystem's revision. Several years ago the entry point $DATC was put into RTE, and its value was the actual revision date (HP format or 2013, for example). However, in the early 1980s, $DATC got stuck at 2013 for several revisions. (The workaround is easy—just use the RP command in the SYSGEN, that is, $DATC,RP,2326.)

To ascertain FMP changes, the entry point $FREV was created at 2101 and remains there for subsequent revisions. $FREV reflects accurately the revision of the FMP package. The date code 2101 signifies the first time extendable type 1 and 2 files became available. Programs that do type 1 access to other file types (like types 3, 4, and 5) are affected because the extents are provided automatically to the caller without the usual FMP-12 error after every end-of-extent. Other major changes affecting the FMP routines can be tested programmatically with $FREV.

I'm having trouble comparing large integers in FORTRAN. How does it work?

When is a comparison not a valid comparison? With the old FORTRAN compiler (FTN4), any two integer values that differed by more than 32,767 would produce an invalid result. Here's the code produced by the old FTN4 compiler:

```
IF(J.LT.K) TEST = .TRUE.
         LDA   K
         CMA,INA
         ADA   J
         SSA,RSS
         JMP   *+3
         LDA   .TRUE.
         STA   TEST
```

As you can see, K is complemented (change sign) and added to J, and the resulting sign is tested for the relationship. If J happens to be + 18000 and K is − 18000, the relationship will be evaluated incorrectly because the magnitude of the difference (36000) exceeds the maximum value allowed for an integer.

Now FTN4x produces the correct code by using the overflow flag to determine what happened:

```
IF(J.LT.K) TEST = .TRUE.
         CLO
         LDA   K
         CMA
         ADA   J
         SOS
         CMA
         SSA,RSS
         JMP   *+3
         LDA   .TRUE.
         STA   TEST
```

Note that quite a bit of code is needed for this implementation. The new FTN77 compiler is even more effective:

```
IF(J.LT.K) TEST = .TRUE.
         JSB   .CPM
         DEF   K
         DEF   J
         JMP   *+4
         JMP   *+3
         LDA   .TRUE.
         STA   TEST
```

In this case, the routine .CPM is called, a microcoded routine in the RTE-6 opsystem ROMs, that is like .CMW, only faster. Either FTN4x or FTN77 will produce correct code.

How can I be sure if a disc LU is error free?

One problem with any disc system is disc error. Murphy's Law is in effect, and errors seem to occur at the worst possible time (such as during an update to a complex database that garbles links and chains).

A good way to verify an LU is to run the FORMT/FORMC program. A FORMT/C option verifies an LU. In this option, a read-only pass checks that all LU areas can be read without a checksum error. The data content on disc isn't verified, but it does provide a good test of the LU including unoccupied areas.

FORMC has an additional feature to verify the Linus tape cache on CS-80 discs

THE HP 1000 GURU

such as 7908/11/12/14. This is a disc area reserved for the Linus tape where data to and from the tape are written and then transferred; this keeps the streaming properties of the tape fairly intact. However, the cache is not part of any disc LU,

so FORMC is the only program that verifies the cache. Just use a short Linus tape (long ones take a while) and run FORMC to verify the tape; it also verifies the tape cache. You can speed up the process by limiting the blocks to be verified to just one or two.

Note that a Linus tape cache is recommended to improve performance and the tape unit's reliability. The tape cache is specified at generation time.

System Utility Program SYZAP

The utility program SYZAP was developed to minimise the effect of 'number crunchers' or CPU bound programs on the ordinary user. The HP1000 system at BHP Central Research Labs is our only general purpose computer and as such it supports a wide variety of users. Most users use the HP1000 for developing programs or for running interactive or I/O bound programs such as data bases but a few users run number crunchers such as signal processing, finite element, models, etc., on a regular basis. The effect of several of these at once, all running with the default priority of 99, on the interactive programs kept us busy handling complaints (18 terminals minus 3 number crunchers leaves up to 15 angry users).

We adjusted program priorities manually for a time and tried unsuccessfully to get the users to load their number crunchers with a lower priority or to batch their programs to run after hours but we still had problems. It was finally decided that an automatic priority adjuster would be the fairest method of overcoming our problems.

While by no means perfect, the program SYZAP has satisfied most of the complaints. The program is started from the WELCOM transfer file when the computer is booted up and is permanently on the system. The program re-schedules itself at regular intervals and looks down the system lists of active programs. A percent CPU usage figure is calculated for programs and their priority is adjusted up or down using this information.

There are eight parameters which can be adjusted to optimize the operation of SYZAP. These are —

interval — the interval between re-schedules of the program, (seconds).

hi_threshold — this is the %CPU usage above which programs become candidates for having their priorities lowered.

lo_threshold — this is the %CPU usage below which programs become candidates for having their priorities raised.

down_delay — the number of consecutive intervals that a program must exceed the hi_threshold before its priority is lowered

up_delay — the number of consecutive intervals that a program must be below the lo_threshold before its priority is raised.

def_priority — the system default priority.

hi_priority — the highest priority in the working range.

lo_priority — the lowest priority in the working range.

The values used in our system are —

interval = 15 seconds.
hi_threshold = 60%.
lo_threshold = 10%.
down_delay = 2 intervals.
up_delay = 4 intervals.
def_priority = 99.
hi_priority = 101.
lo_priority = 106.

So for example, a number cruncher starting off with the default priority of 99 will have its priority reduced to 101 if it exceeds 60% CPU usage for two consecutive intervals. Its priority will continue to fall every

second interval until it reaches a priority of 106 or its CPU usage falls below 60%. If at this stage its CPU usage falls below 10% then its priority will be increased every fourth interval until it comes back to 99 or until its CPU usage exceeds 10%. SYZAP cannot affect compilers, linkers, etc. with priorities less than 99 and it rarely affects 'normal' programs.

PROGRAM DESCRIPTION

The program has three main sections, the first uses an EXEC 12 to delay re-scheduling of the program by the set interval.

The second section uses the operating systems scheduled and dormant programs lists to find programs with the default priority or with priorities in the working range. This information is used to maintain a linked list of currently running programs. The CPU usage is calculated from words 10 and 11 of the programs session control block and from the interval. If the CPU usage crosses one of the thresholds then a counter is updated to keep track of the number of consecutive intervals that a particular threshold has been crossed.

The last section looks down this list and compares the counter to the up and down delays and when appropriate it uses the MESSS routine from the system relocatable library to change the priority of the offending program.

The program SYZAP should be loaded as a normal background program (i.e. not LB or EB) and should be given a fairly high priority (~ 20).

C. STELLER,
BHP Central Research

```
PROGRAM Syzap;
{ This program monitors all programs with the default priority or
    with priorities in the working range and alters priorities
    with the aim of limiting the effect of number crunchers on
    normal users.
        The program maintains a list of all active programs with
    priorities in the working range. The percentage CPU usage of
    each of these prgrams is calculated and compared to the two
    thresholds. If the usage is greater than the hi_threshold
    for a set number of intervals then the programs priority is
    reduced. If the usage is less than the lo_threshold for a set
    number of intervals then the priority is increased.

    C.Steller 2/83
    BHP Central Research
}

CONST   hi_threshold=60;         { cpu % threshold for zap down }
        lo_threshold=10;         { cpu % threshold for zap up }
        interval=15;             { interval between schedules (seconds) }
        dn_delay=2;              { zap down after this many intervals }
        up_delay=4;              { zap up after this many intervals }
        def_priority=99;         { the system default priority }
        hi_priority=101;         { the highest priority in working range }
        lo_priority=106;         { the lowest priority in working range }

{///////////////////////////////}

        schedule=969;            { system lists used to find programs }
        dormant=971;
```

```
        name_length=6;          { string lengths }
        spec_length=32;

TYPE    INT=-32768..32767;

        names=PACKED ARRAY [1..name_length] OF CHAR;
        specs=PACKED ARRAY [1..spec_length] OF CHAR;

        list=RECORD             { Define list elements }
                name:names;            { programs name }
                prty:INT;              { programs priority }
                sesn:INT;              { address of scb }
                flag:BOOLEAN;
                lstt:INTEGER;          { last cpu usage }
                cntr:INT;              { counts threshold crossings }
                next:^list
             END;
        link=^list;

        times=ARRAY [1..5] OF INT;     { array for EXEC time }

        buff=RECORD             { array for SCB CPU usage }
                f1:ARRAY[1..7] OF INT;
                cpu:INTEGER;
                f2:ARRAY[1..32] OF INT
             END;

CONST   blank_spec=specs[spec_length OF ' '];

VAR     head:link;                     { head of list }
        this_time,last_time:times;     { time interval }

{*****************************************************************}

PROCEDURE Delay $ALIAS 'EXEC'$     { Delay rescheduling }
                (icode:INT;
                  iname:INT;
                    iresl:INT;
                      imult:INT;
                        iofst:INT);

EXTERNAL;

{*****************************************************************}

PROCEDURE Time $ALIAS 'EXEC'$      { Get system time }
                (icode:INT;
                   VAR itime:times);
EXTERNAL;

{*****************************************************************}

FUNCTION IGET(addr:INT):INT;       { Peek location }
EXTERNAL;

{*****************************************************************}

FUNCTION IDGET(prgm:names):INT;    { Get pointer to ID seg of prgm }
EXTERNAL;

{*****************************************************************}

PROCEDURE GTSCB(VAR b:buff;              { Get SCB of user }
                    len:INT;
                       VAR err:INT;
                          addr:INT);
EXTERNAL;

{*****************************************************************}

FUNCTION MESSS(b:specs; l:INT):INT;      { Execute system command }
EXTERNAL;

{*****************************************************************}

FUNCTION Get_Name(addr:INT):names;

{ This function returns the program name
    of the ID segment whose address was given in addr }

BEGIN

    Get_Name[1]:=CHR(IGET(addr+12) DIV 256);
    Get_Name[2]:=CHR(IGET(addr+12) MOD 256);
    Get_Name[3]:=CHR(IGET(addr+13) DIV 256);
    Get_Name[4]:=CHR(IGET(addr+13) MOD 256);
    Get_Name[5]:=CHR(IGET(addr+14) DIV 256);
    Get_Name[6]:=' '

END;

{*****************************************************************}

PROCEDURE Fill_List;

{ This procedure first uses Look to add any new programs to the
    list and mark all those in the list that are still present.
    All unmarked programs with no ID segments are then removed
    from the list.
```

```
        The total CPU time from the last call is calculated then
    the CPU usage of programs is used to calculate the percentage
    use for each program in the list.   }

VAR     step,temp:link;
        addr,e,pc:INT;
        total:INTEGER;
        buf:buff;

{----------------------------------------------------------------}

PROCEDURE Look(a:INT);

{ This procedure looks down a list of ID segments starting with
    the address given in a, and adds any new programs with
    priorities in the working range to the list }

VAR     temp:link;
        n:names;
        found:BOOLEAN;
        p:INT;

BEGIN

    WHILE a<>0               { step down list }
      DO BEGIN

          p:=IGET(a+6);           { get priority of program }
          IF (p=def_priority)
             OR ((p>=hi_priority) AND (p<=lo_priority))
            THEN BEGIN

              n:=Get_Name(a);      { get name of program }
              temp:=head;          { search for program in list }
              found:=False;
              WHILE (NOT found) AND (temp<>NIL)
                 DO IF temp^.name=n
                     THEN found:=True
                     ELSE temp:=temp^.next;
              IF found
                THEN BEGIN             { update priority if found }
                  temp^.prty:=p;
                  temp^.flag:=True
                END
                ELSE BEGIN
                                     { add new record if not found }
                  NEW(temp);
                  temp^.name:=n;
                  temp^.prty:=p;
                  temp^.sesn:=IGET(a+32);

                  temp^.flag:=True;
                  temp^.lstt:=0;
                  temp^.cntr:=0;
                  temp^.next:=head;
                  head:=temp

                END

            END;

          a:=IGET(a)              { move to next ID segment in list }

        END;

END;

{----------------------------------------------------------------}


BEGIN   { of Fill_List }

    last_time:=this_time;             { get system time for interval }
    Time(11,this_time);

    Look(IGET(schedule));             { look down shcedule list}
    Look(IGET(dormant));              { look down dormant list}

    step:=head;
    head:=NIL;
    WHILE step<>NIL
      DO BEGIN                        { delete terminated programs from list }

        temp:=step;
        step:=step^.next;
        IF (temp^.flag) OR (IDGET(temp^.name)<>0)
          THEN BEGIN
            temp^.flag:=False;
            temp^.next:=head;
            head:=temp
          END
          ELSE DISPOSE(temp)

      END;

    total:=360000*(this_time[4]-last_time[4])       { calculate total }
          +6000*(this_time[3]-last_time[3])         { cpu time in the }
          +100*(this_time[2]-last_time[2])          { last interval. }
              +(this_time[1]-last_time[1]);
```

```
        step:=head;              { step down list }
        WHILE step<>NIL
          DO BEGIN

            GTSCB(buf,41,e,step^.sesn);     { calculate CPU usage }
            IF (step^.lstt=0) OR (buf.cpu=0)
              THEN pc:=0
              ELSE pc:=(100*(buf.cpu-step^.lstt)) DIV total;
            step^.lstt:=buf.cpu;

            IF pc>hi_threshold              { test against hi threshold }
              THEN BEGIN
                IF step^.cntr>=0                        { increment counter }

                  THEN step^.cntr:=step^.cntr+1
                  ELSE step^.cntr:=1
              END;

            IF pc<lo_threshold              { test against lo threshold }
              THEN BEGIN
                IF step^.cntr<=0                        { decrement counter }
                  THEN step^.cntr:=step^.cntr-1
                  ELSE step^.cntr:=-1
              END;

            step:=step^.next

          END

      END;
    {**********************************************************************}

    PROCEDURE Zap_List;

    { Look down list and change priority of programs if required }

    VAR     step:link;

    {----------------------------------------------------------------------}

    PROCEDURE Zap(up_dn:INT);

    { This procedure does the actual change of priority using MESSS }

    CONST   zap_cmd='PR,     ,  ';

    VAR     zapo:specs;
            i,p:INT;

    BEGIN

        step^.cntr:=0;                      { reset counter }
        p:=step^.prty;
        IF ((up_dn=-1) AND (p=hi_priority))     { calculate new priority }
          THEN p:=def_priority;
        IF ((up_dn=-1) AND (p>hi_priority))
          THEN p:=p-1;
        IF up_dn=1
          THEN IF p=def_priority
            THEN p:=hi_priority
            ELSE IF p<lo_priority
              THEN p:=p+1;

        IF step^.prty<>p          { if new priority is not the same as }
          THEN BEGIN              { the current priority then change it }

            step^.prty:=p;
            zapo:=zap_cmd;
            FOR i:=1 TO 5
              DO zapo[i+3]:=step^.name[i];
            zapo[10]:=CHR(48+((p DIV 100) MOD 10));
            zapo[11]:=CHR(48+((p DIV 10) MOD 10));
            zapo[12]:=CHR(48+(p MOD 10));
            i:=MESSS(zapo,12)

          END

      END;
    {----------------------------------------------------------------------}

    BEGIN { of Zap_List }

        step:=head;                         { step down list }
        WHILE step<>NIL
          DO BEGIN

            IF step^.cntr>=dn_delay
              THEN Zap(1);                       { decrease priority }

            IF step^.cntr<=-up_delay
              THEN Zap(-1);                      { increase priority }

            step:=step^.next

          END

      END;
    {**********************************************************************}
```

```
    BEGIN

        head:=NIL;
        Time(11,this_time);              { get initial system time }

      REPEAT

        Delay(12,0,2,0,-interval);       { delay program }

        Fill_List;                       { update list }
        Zap_List                         { change priorities }

      UNTIL False

    END.
```

System Utility Program BREAK

If you use the IFBRK routine to detect a break as set by the break mode BR command then the utility program BREAK could make your program more friendly to a low level user. BREAK uses a feature of the MUX drivers to prevent the 'S=XX COMMAND ? ' prompt appearing and instead sets the break flag directly.
See program comments for details.

C. Steller
BHP Central Research.

```
          PROGRAM BREAK
C
C     This program will set the break flag of the users currently
C      running program on an unsolicited interrupt from the terminal.
C      This will only work for mux terminals, it has no effect on
C      other terminals ( break mode will still work ).
C     This is enabled with the  EXEC call
C
C     CALL EXEC(3,1473,IDGET('BREAK '))
C
C         ( See program scheduling in the MUX User's Manual )
C
C     This will set up this program to be scheduled instead of
C      the PRMPT break mode program when an interrupt from the
C      keyboard occures. When scheduled this program will search
C      down the list of scheduled programs to find a program with
C      the same session ID as that obtained from the EQLU call
C      ( the EQLU call at the start of the program returns the LU
C      of the terminal that caused the interrupt). This program
C      then uses MESSS to issue the system command 'BR,name'.
C
C     After the above EXEC call your program can use IFBRK to test
C      the break flag in the usual way. NOTE that your program must
C      do an EXEC(3,1473,IDGET('PRMPT ')) before the program ends to
C      restore PRMPT as the interrupt program. Also it is a good idea
C      to leave out these EXEC calls until after the operation of
C      the break detect section of your program has been tested with
C      the normal method of setting the break flag.
C
C     Loading
C       1.  Must not be loaded LB or EB.
C       2.  Must be RP'ed when wanted or made permanent.
C       3.  Should have high priority (~10)
C
C     C.Seller      12/81
C     BHP Central Research.
C
          DIMENSION icmnd(14)          ! buffer for the BR command

          CALL EQLU(lu)                ! get interrupting LU

          link=IGET(IGET(1711B))       ! get schedule list start

10        IF (link.EQ.0) GOTO 999      ! if end of list then end
          id=IAND(IGET(link+31),377B)  ! get SESSN ID from ID SEG
          IF (id.EQ.lu) GOTO 20        ! is it the one we want ?
          link=IGET(link)              ! if not then step down list
          GOTO 10                      ! and loop again

20        icmnd(1)=41122B              ! form command 'BR'
          icmnd(2)=20054B              ! ' ,'
          icmnd(3)=IGET(link+12)       ! get name word 1
          icmnd(4)=IGET(link+13)       ! get name word 2
          icmnd(5)=IAND(IGET(link+14),177400B)+40B

          IA=MESSS(icmnd,9)            ! send command to system

999       END
```

COMING EVENTS

28th May:	RTE-A Programming & System Management Course, HP Sydney.
28th May:	Series 200 Basic Operating and Programming Course, HP Melbourne.
4th June:	HP 9000 Introduction to HP-UX, HP Sydney.
6th June:	HP Technical Computer User Group (N.S.W.) Meeting.
18th June:	IMAGE/1000 Course, HP Sydney.
2nd July:	RTE-A Programming & System Management Course, HP Melbourne.
11th July:	HP 1000 Users Group (Vic.) Meeting at HP Melbourne.
16th July:	RTE-6/VM Session Monitor Course, HP Melbourne.
16th July:	HP 9000 HP-UX System Administrator Course, HP Sydney.
30th July:	RTE-6/VM System Manager Course, HP Melbourne.
30th July:	HP 9845 Operating & Programming Course, HP Melbourne.
6th August:	IMAGE/1000 Course, HP Melbourne.

NOTE: If you have requirements for training on any HP products not covered in our formal training programme, please contact Lea Thompson on (03) 895 2505.

FOR SALE

One 2645A DISPLAY STATION
— Block or character mode, 4096 bytes of RAM memory plus HP13234A 4kbyte extra terminal memory installed.
— option 007 integrated dual cartridge tape-mini datastation including device support firmware.

One 2647A INTELLIGENT GRAPHICS STATION
— option 007 integrated dual cartridge tape-mini datastation HP 12966A Buffered Async interface and HP-IB interface.

One 2648A GRAPHICS STATION
— option 007 integrated dual cartridge tape-mini datastation including HP13231A display enhancements and HP13234A extra 4k RAM terminal memory

One HP7906H 19.6M Byte ICD DISC DRIVE
One HP12786B 256 kbyte MEMORY PACKAGE
— To suit HP1000/E (HP2176D)
— Replaced in upgrade to high-performance memory system
Three HP12966A Buffered Async Communication Interfaces

PRICES ARE NEGOTIABLE.

Please contact Peter C. Grant on (03) 420 7295 if any of the above items are of interest.