CROSSTALK Journal of Hewlett-Packard Technical Computer User Groups OCT. 1982

CO-ORDINATORS COMMENTS

Many thanks to all those who have helped to get Crosstalk on its feet, through the submission of articles, assistance with the various administrative and organizational tasks, and the general interest shown. I have received many positive comments from readers, together with several helpful suggestions which have been implemented. Thank you for your interest and involvement.

The first issue of Crosstalk had a distribution of about 400 copies to members of three users groups. This, the third issue, will have a distribution of around 700 copies to members of the following five groups:

HP1000 Users Group (N.S.W.)
HP1000 Users Group (Vic.)
HP Desktop Computer Users Group (Vic.)
Canberra Technical Computer Users Group
Queensland Technical Computer Users Group

To organize contributions to Crosstalk, each users group has nominated their own group editor. This person has the task of collecting articles and advertisements from their group members, and editing these in terms of deciding what should be forwarded to me for publication. My role is to co-ordinate the inputs from the various groups, and forward these in suitable form for printing. A list of the local group editors and submission details for articles and advertisements will be included in each issue.

Your comments, suggestions and constructive criticisms concerning Crosstalk will be most welcome. Remember, your articles are vital to Crosstalk's survival, so keep involved and keep Crosstalk 'alive and kicking'.

Glenda Patterson,
Crosstalk Co-ordinator,
HP Melbourne

HP 2700 Series Color Graphics Terminals

The HP 2700 series of high performance color graphics terminals offers models designed for technical and business users. Process control, spectral analysis, circuit design, financial analysis, transparency preparation, and boardroom display are just a few of the many applications for which the HP 2700 terminals are well suited.

The HP 2700 has powerful local capability to store vectors, group vectors into objects, and then perform manipulations on those objects. These local graphics capabilities can be accessed by a host computer, greatly reducing the data communications and CPU overhead associated with changes to the graphics image. The host computer can distribute parts of the graphics processing to the terminal for an overall performance improvement.

HP 2700 application software permits users to generate charts locally (AUTOPLOT/2700) or sketch electronically on the display from a tablet (PAINTBRUSH/2700). The creation and subsequent modification of graphics images can be performed standalone or by sharing the workload with a host computer.

LOCAL VECTOR STORAGE

Vector list storage is the key to the local graphics processing power of the HP 2700. Graphical information is transmitted to the terminal as a series of X, Y co-ordinates. The HP 2700 stores the X, Y co-ordinates in its vector memory and also converts the graphical information into a raster image. Storing images in vector form allows the terminal to perform a wide range of functions and graphics manipulations the host computer formerly had to do. This distribution of graphics processing to the terminal dramatically increases performance, significantly reduces the

CONTENTS

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

strain on the CPU, and reduces the amount of data which must flow over the data communications line.

LARGE ADDRESSABLE WORKSPACE
The HP 2700 series boasts an addressable resolution from (—16383, —16383) to (16383, 16383) or 32K by 32K. This huge plotting space enables the terminals to store extremely complex images with a resolution of more than one billion addressable points.

TRUE ZOOM AND PAN
The large addressable workspace and local vector storage combine to allow a true local zoom and pan. From the keyboard or via software running on a host computer, the user can zoom from a map of the world to a close-up of a particular country, or from the full view of a printed circuit layout to an individual component. When the operator zooms in, details become visible which were unseen before. Raster memory can be expanded so that the HP 2700 will buffer redraws. When buffering redraws any screen updates will not be seen by the user until they are complete. Zooms and pans will appear smooth.

LOCAL 2-D TRANSFORMATIONS
The HP 2700 uses locally stored vector lists to define the basic characteristics (such as size, shape, and color) of graphics objects. Each graphics object can be scaled, rotated, translated, or copied locally. If an application needs to show the same symbol in several different locations significant data communications time may be saved by sending the vector list once and then instructing the HP 2700 to place copies of it at different locations.

MULTIPLE VIEWS
The HP 2700 can show any portion of its large workspace on any portion of the screen. Powerful terminal commands make this process simple. The host does not have to recalculate or resend the pictures associated with any of the views, since they are already stored in vector memory. Up to 255 different views may be specified.

INTERACTIVE COLOR SELECTION
Raster scan technology is the basis for the HP 2700's high quality color display. Any 16 of 4096 available colors can be displayed at one time. The wide choice of colors frees the user to select those colors which best suit the application.

A unique, interactive keyboard interface provides the user with two methods of color specification: 1) mixing red, green, and blue; or 2) selecting hue, saturation, and lightness. With the push of a key, a menu is displayed and the user is allowed to select one of the 16 pen colors to modify. By rotating the thumbwheels on the keyboard, the user can select any of the available 4096 colors. The screen picture is immediately updated to reflect the new color selections.

Any set of 16 colors can be saved within the terminal as a palette. Up to 255 palettes may be defined.

INDEPENDENT COLOR ALPHANUMERICS
The HP 2700 series of terminals offers forms mode, line drawing character set, insert/delete line, insert/delete character, cursor control, numeric keypad, 16 screen labelled user-definable function keys, and three and one-half pages of memory. (More or less memory can be assigned for alphanumeric display depending upon the user's requirements for vector storage).

In addition, the HP 2700 allows the user to select 16 foreground/background alphanumeric color pairs from an available selection of 64 colors. These colors are independent from the graphics palette.

DUAL INTEGRATED MINI-DISC DRIVES
The HP 2700 features two optional integrated 5-¼" disc drives. Each double sided, double density flexible disc can store up to 270K bytes of formatted information. The HP 2700 file system supports named, sequential, 8-bit data files. Files may be of indefinite length, limited only by physical media constraints. Disc commands are easily accessed through terminal defined function keys and a file directory can be displayed to show individual file names and characteristics.

APPLICATION SOFTWARE
Two local software packages address specific application areas and allow the terminal to operate as a standalone workstation.

AUTOPLOT/2700
AUTOPLOT/2700 is a powerful yet friendly software application package that lets the first time user create professional looking business charts in minutes. Once data has been entered into the menu, any chart type can be plotted to the display with a single keystroke. By having only one data menu for all chart types, the data can be viewed in pie, bar, line, or log form without re-entering the data, making it easy to select the chart type that best presents the data.

The same friendly interface extends to provide sophisticated, large scale periodic reporting capability. AUTOPLOT/2700 charts and data files are easily saved and recalled from flexible mini-discs. Data can also be read from a host data base. Since data files are stored separately from plot specifications, only changes to the data file itself are required to replot periodic charts with the latest information.

This truly interactive interface lets the user modify the chart while viewing it on the screen. For example, to change the color of a single pie section or the location of the entire chart, the user simply points with the graphics cursor to the object to be modified. Then the user points to the new color or location and the change happens immediately. In addition, text slides may be designed in a totally interactive environment. As the user rotates the thumbwheels, lines of text move across the screen and into position. Several lines of text can be moved at one time or changed to the same color, font, or text size. Two or more charts can also be merged with text to make a single combined chart. These highly flexible design options let the user create charts that communicate vividly and precisely.

Once designed the finished chart can be copied to a plotter, RGB output camera, or raster dump printer. Taking advantage of the chart advance feature of certain HP plotters and printers, AUTOPLOT/2700 provides for multiple copies of charts to be made unattended by the user. The finished charts can also be stored in raster or vector format on local disc or in a host computer data base.

PAINTBRUSH/2700
PAINTBRUSH/2700 is a software application package that provides a means for drawing, editing, and combining pictures, including AUTOPLOT/2700 charts. The user can also edit pictures generated by host software.

The 13273T Data Tablet is an ideal companion to PAINTBRUSH/2700. The tablet stylus provides easy, natural control of the cursor for freehand drawing and menu selection. In freehand drawing mode the user can select color, line types, area fill patterns, and pen tips. Aided drawing modes facilitate the creation of straight line segments, arcs, circles, boxes, and curves. The user can specify a grid for selecting vector end-points. This "graph-paper" drawing mode is useful not only for drafting-like applications, but also for flow charts and business diagrams.

While in PAINTBRUSH/2700 the user can select different pictures or parts of pictures to move, rotate, scale, or copy. The user can also add and edit graphics text labels. This allows the modification of pictures such as a schematic, PC layout, pie chart, or floor plan.

As with AUTOPLOT/2700, the final picture created in PAINTBRUSH/2700 can be copied to a plotter, RGB output camera, or raster dump printer. The finished picture can also be stored locally on disc, or by the host.

PAINTBRUSH/2700 and AUTOPLOT/2700 may be used together to give the user the ability to combine local and host driven charts with freehand graphics. Combined with the many hardcopy interfaces available, the result is impressive speed and flexibility for the generation of professional presentation graphics.

Using IMAGE from Pascal

Some of the deficiencies in PASCAL file handling can be solved with the use of the IMAGE subsystem. This provides PASCAL with a sophisticated data base capability. But the use of IMAGE from PASCAL is not described in any detail in the 1000 documentation.

There are several aspects of the use of IMAGE which could do with clarification. These are:

1. Unpacking buffers returned by IMAGE into variables.
2. Interpretation of the IMAGE status array.
3. Declaration of the external IMAGE routines.
4. Specification of search item parameters.

These problems essentially arise from the fact that some of the parameters which are used in IMAGE calls have varying types depending on the data base structure, but the parameter types in the EXTERNAL declarations must be fixed. Additionally, unpacking the buffers returned by IMAGE can be problematic as a series of routines need to be written to perform this function. FORTRAN achieves automatic buffer unpacking through the use of the EQUIVALENCE directive by which variables can be mapped onto buffer segments. Thus when the buffer is modified, so are the equivalent variables.

No such structure is directly available in PASCAL but can be achieved by using a facility known as 'VARIANT RECORD TYPES'. This facility is standard PASCAL and nothing 'tricky' is done to make it do what is required. Essentially the technique involves the declaration of a data type which can have several different structures. The type of structure used depends on the set being used. For example, consider a simple data base consisting of two data sets:

```
SET       DSETA
ITEMS     A1      X20  (key item)
          A2      I
          A3      I*4
          A4      R       15 word record

and

SET       DSETB
ITEMS     B1      I    (key item)
          B2      X40
          B3      R       23 word record
```

Whether these sets are master or detail has no relevance for the purpose of the exercise.

The first thing we need is a 1 word integer data type. Next we declare a type which contains the names of our data sets. Then we declare a buffer type which consists of a single word integer array as long as our longest set length, in this case 23 words long. Next we define a search item buffer type which is an array of single word integers as long as our longest search item, in this case 10 words.

For example:

```
TYPE

          integer2  = [-32768..32767];              (* SINGLE WORD INTEGER *)
          settype   = SET OF (dseta,dsetb,buffer);  (* SET NAMES *)
          setbuffer = ARRAY [1..23] of integer2;    (* DATA SET BUFFER *)
          keybuffer = ARRAY [1..10] of integer2;    (* KEY ITEM BUFFER *)
```

Note that the SETTYPE type includes a type buffer. This is used in the next step when we declare our buffer type.

The next step involves defining a variant record type which will be the basic structure descriptor for the data base. One variable is defined for each data set in the VAR section and each of these is of this type. The type for our example data base is as follows:

```
TYPE
          dataset = PACKED RECORD
                    tag : settype;
                    CASE settype OF
                         dseta: ( a1: PACKED ARRAY [1..20] OF char;
                                  a2: integer2;
                                  a3: INTEGER;
                                  a4: REAL );
                         dsetb: ( b1: integer2;
                                  b2: PACKED ARRAY [1..40] OF char;
                                  b3: REAL );
                         buffer:( buff: setbuffer );
                    END;
```

The 'tag' field must be included but only exists to divide the declaration into separate data sets.

The last item defined is mandatory as this is used to specify the entire buffer entity later. Its name is arbitrary, but the type is important. Its type has been previously specified and is as long in words as the longest record structure defined.

NOTE Each item name in this declaration must be unique. If an item is used in multiple sets, it must be varied on each occurrence of it in the DATASET record definition.

Now it is possible to declare a series of buffers, one for each dataset, each of type DATASET. How these buffers are used will be revealed later.

Now it is necessary to define a buffer for your key items. This is done in a very similar manner:

```
TYPE
          keytype = PACKED RECORD
          tagf : settype;
          CASE settype OF
               dseta: ( a1: PACKED ARRAY [1..20] OF char );
               dsetb: ( b1: integer2 );
               buffer: (buff : keybuffer );
          END;

Now a series of key items can be declared each of which is type 'keytype'.
Our VAR declaration would look like this:

VAR
          dseta: dataset;
          dsetb: dataset;
          keya1: keytype;
          keyb1: keytape;
```

The next step is to declare the IMAGE routines as external and to declare the parameter sequences. The buffer variables are declared to be type SETBUFFER and the key item fields are declared as type KEYBUFFER.

For example:

```
PROCEDURE dbget ( base : PACKED ARRAY [1..8] OF char;
                  set  : PACKED ARRAY [1..6] OF char;
                  mode : integer2;
                  status:ARRAY [1..10] OF integer2;
                  ilist: PACKED ARRAY [1..2] OF char; (* always '@;' *)
                  buff : SETBUFFER;
                  key  : KEYBUFFER );
EXTERNAL
```

Now imagine that DSETA is a manual master and we want to perform a mode 7 (calculated) access. We first set the key item value we want to find:

```
keya1.a1:='THE VALUE OF MY KEY';
```

Now, we can call the IMAGE routine to retrieve the record with this key item value, we can

```
CALL dbget(base,'DSETA ',7,status,'@;',
                    dseta.buff,keya1.buff);
```

To access the entire VAR dseta as a complete entity for the purposes of the pass to DBGET we simply reference 'dseta.buff'. To reference the item a2 within the buffer we simply reference 'dseta.a2'. Nothing would stop us from referencing 'dseta.b2' but this would return 40 characters some of which are undefined and some of which are the internal representation of the real and integer values in the buffer.

What the variant record structure has said is that each variable of that type is a record which can have several structures. Which portion of the record is accessed depends on which component of the record structure is accessed. This addressing is determined at compile time, not at run time. Therefore the value of the tag field has no relevance. The tag field in the record definition exists only to allow the system to break up the various record structures.

This technique allows the IMAGE external declarations to declare one buffer type. That type is the same type as one of the record components, buff, which every data set buffer variable has in common. Additionally, the type of the search item parameter (or the key item) can be fixed despite the fact that the interpretation of the item in the buffer can be varied by specifying varied record components. The variant record type has made equivalent the single 'buff' entity and the various record structures of the data sets and the key items.

A similar technique exists for the decoding of the information in the IMAGE status array which is returned from IMAGE calls. The way that this array is structured is that some of the return codes are returned as single precision integer values and some are double precision (which is standard PASCAL integer). Mixed data types within one array are not allowed so it is necessary to declare a record structure for the status array. Several different structures can be declared, one for the each status return format provided by the IMAGE calls. For example, DBGET and DBFND return different values in their respective status arrays. However, included in the various record structures must be a single array structure so that the record as a whole can be passed to IMAGE. It is not possible to simply pass the variable which is of type STATUSRECORD to image as space is left in the record for the tag field variable even though it is not used at run time. Since IMAGE assumes that the first word of the parameter is the first relevant word, we must find some way of passing the body of the record without the attendant tag field. The only way to do this is to pass a component of the record which spans the entire record. This is also the rationale behind the structure of the data set buffers and the key item records.

Ralph Baer
Systems Engineer
Hewlett-Packard, Adelaide.

UNDERSTANDING COMPUTER TERMINOLOGY

PRIORITY INTERRUPT:	Yes, you may leave the room.
PUNCH & VERIFY:	Shoot first, ask questions afterwards
UNIBUS:	Student transport
TAPE CONTROLLER:	Worm powder
DISK CONTROLLER:	Juke box
ACCESS TIME:	Licensing hours
SEQUENTIAL ACCESS:	Pub crawl
RANDOM ACCESS:	Any pub will do
RESPONSE TIME:	Delay following YOUR SHOUT instruction
OVERLAY:	Achievement of over-productive hen

Attempting to Grow Money on Trees

A talk presented to the HP1000 Users Group (N.S.W.) by A. J. Turner, 4/8/82.

I intend to describe to you some aspects of what we are attempting to do at ACT Forests, but first I will give you a little history.

Pine plantations in the ACT were first established during the early 1920's as part of the 'beautification' of Canberra and erosion control in the Cotter Catchment. Small plantings then continued up to the beginning of the Depression.

From about 1930 onwards, the planting rate increased under unemployment relief programs, so that by the start of World War II, a sizeable amount of land had been established. From then until the 1950's only small areas were planted. The 1960's saw an upsurge in plantation establishment, as occurred in the states with the Commonwealth Forestry Agreements until a maximum area was planted in the early 1970's. At present, we are about 2000 hectares short of our approved target. Although some plantations were established after clearing of Eucalypt forest in the 50's and 60's, current policy is that establishment only occurs on resumed farming land that at best can only be considered as marginal for farming purposes.

Also in the early 1970's the management of the forest was changed with the establishment of a large utilisation plant in the A.C.T. This meant that stands that were approaching maturity could be exploited and that stands could be properly thinned to maximise the production of large size logs. This was abruptly interrupted when 600 hectares of pre-war plantation was windblown and had to be quickly salvaged before it deteriorated in 1974.

These factors left an estate with a non-continuous distribution of age classes that had to be managed to yield a continuous supply of logs of a certain quality to meet our commitments to the local purchasers.

It is obvious that the range of alternatives for management is enormous and it would require a lot of computations to calculate the optimum management regimes for the estate, especially

considering the enormous constraints of the close juxtaposition of the forests to the National capital. This juxtaposition is continually impressed on the forest managers by people outside the organisation — one example being that there are approximately 200,000 visits to the plantations per annum.

Over the past twenty years regressions were developed that enabled a representation of the total estate to be made by the statistical sampling of the forest and computer programs to estimate the availability of timber. However, this sampling only occurs every 5 years and a model was necessary to predict the way in which different stands of timber would grow over time. Such a model was developed by a consultant, who demonstrated a relationship between growth, age, site and rainfall.

All the above estimates are on a per hectare basis.

Another important aspect of yield estimation is the accurate estimation of areas. The program of forest surveys was falling behind in 1978 and some individual units had estimates to + or − 20%. The Australian Survey Office was approached and they agreed to Digitise the boundaries of the plantations and extract area data to an accuracy of + or − 2%.

This program is nearly complete.

In the late 1970's, we found that our access to computers for scientific processing was becoming more and more limited, and when we suggested that we would want to plot our compartment boundaries and even re-digitise them as they changed with time, the local computer manager threw up his hands in horror. We also wanted to use the model to optimise the yields from the estate as well as to do some gaming to test what was the most economic management techniques on individual stands of trees. In the environment of a large mainframe with a multiplicity of on-line systems, there was just not the available resources for the type of processing we were attempting.

FINALLY we went to tender for a small computer which would have a powerful processor and also be capable of handling a limited amount of graphics with a digitiser and small plotter.

The result was the following configuration:

HP1000 F series processor with vector instruction set
256 kbytes of memory
7920(H) Disk Drive (50 Megabytes)
7970E Tape drive
2648A Terminal
2631G Printer
9874A Digitiser
9872C Plotter

As you can see, we are configured purely as a single user system so why didn't we get a desktop? — We already are short on terminal resources and currently are assessing quotes for a HP2622 and multiplexor.

Experience with the system has indicated approximately 200% increase in efficiency of the scientific officers who are using it as well as being able to immediately respond to requests from management. Quite frankly it has been worth the wait but it arrived about two years late.

Now to get to the topic of the talk. This lies in the type of gaming mentioned earlier. If a stand of trees is left to grow as it was originally planted, the increase in basal area is distributed over all the trees. This is known as basal area increment. If a stand has a proportion of its trees cut out, then that total increment for the stand will drop in the first year but will then return to about the same as previous years if all other factors remain the same. As a result, there will be a greater proportion of diameter increment per tree than in an unthinned stand and we will get larger trees faster. However this also usually results in a loss of volume production over the life of the stand. Now large logs are a preferred product from the forest, so much so that there are quite large price differentials for larger logs. So if we continue to thin a stand the final crop will yield a much higher value per unit volume than if the stand is left in its original state. Our objective then is to maximise the value return on the forest without losing excessive amounts of volume — or in other words, growing money on trees.

NEW PRODUCTS

AO Size Bertha is Here

Mick O'Rourke/SDD

HP is proud to announce the 7585A drafting plotter, the newest and biggest member in the HP plotter family. The standard paper size is "AO" size (841mm × 1189mm). What we learned from you when we introduced the smaller 7580A last year was that this larger size was a must for many applications. The result is a product that is compatible with the 7580A, and that gives HP a plotter family second to none.

MANY FEATURES BUT LOW PRICE

The 7585A provides many unique contributions:

- Compactness — Light weight, mobile, low power consumption.
- Pen system — Automatic settings, capping, eight pens.
- Intelligence — Circles, character sets, dashed lines.
- Quality output — 0.00012" mechanical resolution with 0.001 programmable addressable resolution.
- Ease of Use — Easy media loading, automatic media settings. In addition to this it is priced considerably lower than the competition.

New HP-86 is HP's Lowest-Priced Personal Computer

Steve Ruddock/PCD

HP's newest, lowest-priced personal computer, the HP-86, offers HP quality at a competitive price. And a host of new software means a broad range of solutions for all professionals are available on the HP-86 at a lower cost than comparable systems.

The modular HP-86 features keyboard and microprocessor in one package, industry-standard 9" and 12" monitors, and a new flexible disc drive offering low-cost mass storage.

All existing Series 80 BASIC software is compatible with the HP-86, and the new machine runs CP/M software with the addition of the HP 82900 CP/M System. All other Series 80 peripherals run with the HP-86.

New software features solutions for professionals including accounting, word processing, data base management, critical-path method analysis, and personal and business tax-planning.

Low-cost, modular hardware, together with the new and existing software solutions, make the HP-86 attractive to professionals in business and office environments, as well as to HP's traditional base of engineering and science professionals.

Four ports in the back of the computer let the user expand the HP-86's power. The HP 82900 CP/M System, the HP 82950 Modem, HP Memory Modules and additional ROMs all can be plugged into the ports.

Dedicated interfaces connect the computer to a display monitor and one or two disc drives. The HP 82912 (9" diagonal) and HP 82913 (12" diagonal)

NEW PRODUCTS

monitors both display text, in an 80-column by 24-line format, and graphics.

The HP 9130 flexible disc drive features 270kb capacity in a new, small package, and uses double-sided, double-density discs.

At less than ⅓ cent per byte, the new drive offers one of the lowest mass storage costs of all flexible disc drives.

The HP 82905B dot matrix impact printer, with compressed and expanded type, and a variety of print formatting options, is expected to be most popular with the new HP-86.

HP-87XM
HP also is introducing the HP-87XM (for "extra memory"). As the increased-memory version of the HP-87A, introduced in March, the HP-87XM features 128kb of user memory, expandable to 640kb.

The HP-87XM will be HP's most expandable personal computer, for those who need to add a large number of peripherals or instrumentation.

HP9826/36 TERMINAL EMULATOR NOW RELEASED

The new 9826/36 Asynchronous Terminal Emulator software is now available. The software manages the keyboard, CRT, and mass storage so that the 9826A or 9836A responds in the same manner as an HP2621 terminal. Written in Pascal, the emulator runs independently of the language system, as does Visicalc.®

The terminal emulator is menu driven and incorporates file transfer utilities, character and line modify modes and supports either ENQ/ACK or XON/XOFF handshakes.

The hardware required to run the terminal emulator is as follows:-

* 9826A or 9836A
* 256 K bytes of RAM or greater
* 98628A Opt. 100 datacomm interface and any cable option

First H.P.D.C.U.G.V. Workshop well attended

It was standing room only at our first workshop held at H.P. Blackburn on September 9th. Forty-six members and friends crowded into the room to hear Phil Greetham talk about Structured Programming. Phil's excellent presentation was enhanced by a number of observations and comments from the more experienced members present and from the questions asked it was apparent that his audience had found it quite thought provoking. Thanks Phil for an excellent start to our workshop series.

Next we were treated to a quick sketch by Ron Davis describing how he had used an HP 85 to organise and control the mixing and dispensing of the ingredients for an ice-cream making process. By enabling the operators to control the mix within tighter limits, waste of expensive ingredients was minimised and the HP 85 paid for itself in its first year.

Whilst members stretched their legs an HP 86 was unveiled and its performance and features explained. We hope to hear more about it at our next meeting.

Then Chris Simpson took the floor to describe programming techniques he had used with a 9845 to gather and organise data from a natural gas supply network. The 9845 was also used to present text and graphic displays to inform and alert the operators of the gas distribution system status. Chris had developed a powerful program using the 9845 system of interrupts to ensure that operating priorities were correctly maintained and important tasks dealt with first. Chris supported his presentation with a number of overhead transparency displays and it was evident that he had put quite a lot of work into his talk which we all appreciated.

With such a full program we ran well overtime but there was still a good crowd for the refreshments supplied by H.P. at the end of this excellent meeting.

B. T. O'Shannassy

QUICKSORT

The following SORT routine written in HPL is based on the QUICKSORT ALGORITHM (see reference). Requiring little code, it is efficient in both execution time and memory requirements. In this form, the string array will be sorted into lowest to highest. It can be modified to reverse this order or to handle numeric arrays.

The following variables are passed to the sort routine:

P initialised to 2 and serves as pointer to r variable stack.
r1 lower bound for sort
r2 upper bound for sort
A first character of key
B last character of key
Z$ dimensioned to length of longest string and serves as temporary storage during exchange.

The following example sorts 100 strings length 20 characters. Sort time 5 secs.

```
dim C$[100,20],Z$[20]
.....
.....
2→P;1→A;20→B
1→r1;100→r2
cll 'SORT C$'
.....
.....
end

"SORT C$":
r(P-1)→L;r P→U;P-2→P

L→C;U+1→D

if C<U;if C$[C+1→C,A,B]<C$[L,A,B];gto+0
if C$[D-1→D,A,B]>C$[L,A,B];gto+0

if C<D;C$[C]→Z$;C$[D]→C$[C];Z$→C$[D];gto-2

C$[L]→Z$;C$[D]→C$[L];Z$→C$[D]

if L<D-1;L→r(P+1→P);D-1→r(P+1→P)

if D+1<U;D+1→r(P+1→P);U→r(P+1→P)

if not P;ret

gto "SORT C$"
```

pop sort bounds from stack
initialise pointers for comparison
compare successive records with lower bound
exchange records if out of order
place lower bound in correct position
do sort on lower subtable
do sort on upper subtable
if stack empty, sort is complete

The routine can also be written as a recursive procedure using p numbers. However, execution time is slower.

Reference: COMPUTER SCIENCE, page 478.
TREMBLAY AND BUNT
McGRAW HILL KOGAKUSHA

Ivan Francis,
Mont Park Pathology Centre,
Mont Park, Vic., 3085.

INSTANT RESPONSE FROM 9825

One way to save having to press the "CONTINUE" key after giving a single character answer to a program generated question involves the use of the "rdb" function available on the General I/O ROM. The short listing shown demonstrates this method.

```
Ø: dsp "IS DATA CORRECT? (Y or N)"
 ;rdb (Ø) → A; if A>128; A-128 → A
1: if A=121; gto "process data"
2: if A=11Ø; gto "modify data"
3: gto Ø
*26Ø98
```

Peter Hendy,
I.G. Hendy Manufacturing,
36 Alfred Street,
Blackburn, 3130.

H.P.D.C.U.G.V.

A reminder to all those who have not yet paid their subscription fee. This will be the last 'Crosstalk' mailed using the old mailing list. After this edition the mailing list of financial members only will be used for all communication. You may pay your $10 subscription to our treasurer personally at the next meeting or mail to:

Treasurer, H.P.D.C.U.G.V.
Ron Davis,
Ridge Road,
Mt. Dandenong, Vic.

INTERRUPT PROBLEMS ON HP9845 DESKTOP COMPUTER

A multi-tasking program was written which enables simultaneous analysis of data, by graphing or printing, while collecting other data under interrupt control from the IEEE-488 bus (HP-IB) and storing this data in files on a mass storage device for future analysis. The analysis part of the program was written to include graphing facilities which enable parts of the graph to be expanded by digitizing a new scale and redrawing the graph.

A number of problems arose in the writing of this program because all interrupts occur as end of line branches in basic language. Those which are worth stating are:-

1. The 'ON INT # 7, Priority CALL Service' statement must appear in the base program with the highest priority number. This requires 'Service' to be a sub-program with data passed to and fro via a 'COM' declaration. All mass storage assignments used with 'Service' must be declared locally in 'Service' itself.

2. Any 'INPUT' statements must be replaced by a multi-line define function that uses an 'ON KBD' statement to achieve the same result. To accomplish this a special binary program is required so that the 'ON KBD' will operate inside the sub-program or multi-line define function. This is available from Hewlett Packard with their Utility II Package. The multi-line define functions were written to accommodate either numeric or string inputs. The use of the menu selection function "FNMenu" from this package made program operation easier.

3. Any 'DIGITIZE' statements must be replaced in a similar manner to 'INPUT'.

4. Care must be taken with plotting statements like 'LIMIT, LOCATE & CLIP' because these can include an inherent digitize command.

5. Other input type statements like 'EDIT, LINPUT & LETTER' are not permitted unless sub-programs are written to replace them. These statements can be replaced by using the "Useform" sub-program supplied by Hewlett Packard in their Forms Utility Package.

6. Mass storage statements like 'COPY & CREATE' should not be used with very large files, as interrupts cannot be serviced until the statements have completed.

7. Mass storage statements, plotting statements and matrix statements that use 'MAT' should only be used with short arrays. Large arrays should be broken down into smaller ones.

8. A long 'WAIT' must be broken down into a number of shorter 'WAITs' using 'FOR.....NEXT' statements.

9. Because of the inclusion of priority interrupt for 'ON INT' and 'ON KEY' the sub-programs "Useform" and "FNMenu" require a priority parameter added to their 'ON KBD' statements.

With the above restrictions and changes the program now works as required with few interrupts lost when data is collected from one to eight sources at a maximum rate of one interrupt from each source per minute. The data blocks transferred contains between 200 and 2000 characters, these character strings are checked for correctness and then stored on a mass storage file in a form which conserves mass storage space.

Anthony J. Stevens,
TELECOM AUSTRALIA,
Research Laboratories,
(03 630 7981)

CAUGHT & IDENTIFIED

— an elusive "intermittent" bug

A 9845 System bug which could cause hours of wasted time in debugging has been identified. It is associated with PRINT USING Image statements containing imbedded characters/spaces or floating text. Only numbers in the range of −.09 to −.01 are affected.

Consider the following code:-

```
10 Image:   IMAGE 10XD.3D              ! 10 spaces imbedded in IMAGE
20                                     !
30 Title:   PRINT USING "K,4X,K";"Actual Value","Value with IMAGE"
40                                     !
50 Print_loop:FOR I=.25 TO -.25 STEP -.025
60              PRINT I;TAB(10);       ! Postion Printhead ready for IMAGE
70              PRINT USING Image;I    ! Print I using the IMAGE
80          NEXT I
90    END
```

```
Actual Value    Value with IMAGE
 .25              .250
 .225             .225
 .2               .200
 .175             .175
 .15              .150
 .125             .125
 .1               .100
 .075             .075
 .05              .050
 .025             .025
 0                0.000
-.025             .025    NOTE:  MISSING SIGN
-.05              .050           between -.01 and -.09
-.075             .075
-.1              -.100
-.125            -.125
-.15             -.150
-.175            -.175
-.2              -.200
-.225            -.225
-.25             -.250
```

```
FIX ..... 10 Image:   IMAGE 10X,D.3D       ! Separate Spaces from
                                           ! Print List with Comma.
```

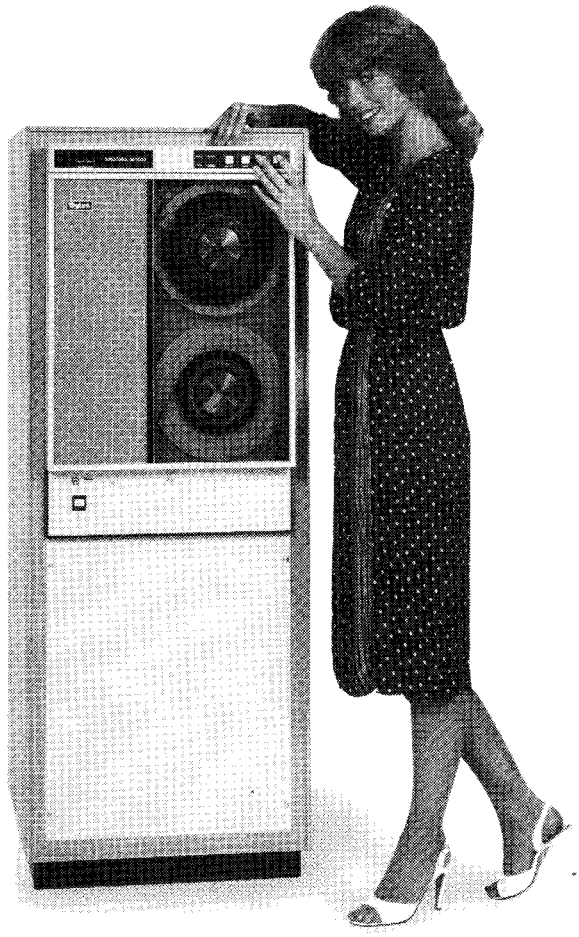
Now you can get high speed, dual density tape drives for your HP-1000L Series

Dylon Series Ten Gives You:

Dual Density (800/1600 B.P.I.)
7", 8½" or 10½" Reels
Standard or Vacuum Column Versions
Rack mounting or Packaged Styles
Transport speeds up to 125 ips

And

They all plug right into your HP-IB interface with complete RTE/L and RTE/XL operating system compatibility.



DATAMATIC PTY LTD

SYDNEY 438 2699 MELBOURNE 818 0674
ADELAIDE 297 8510 BRISBANE 31 2350
PERTH 382 1744

This is only one of many Dylon products available. Consult Datamatic for your HP-IB Tape requirements.

Focus 1000
PRINTING TERMINALS

What can you use to get a hardcopy listing of your program when you happen to be separated from your computer (and printer) by 3 floors and a busy road?

STANDARD SOLUTION
(Any resemblance to any salesman living or dead is purely co-incidental).
— Buy a 262x terminal with an integral thermal printer (printer option =∿ $1700).

JUST A MOMENT, I've already got a perfectly good 2382 terminal and besides, I don't like thermal paper because it fades and costs too much.

— Well then, you can buy a 262x terminal with a printer port and a 2631B printer. It will only cost you about $7000.

TOO MUCH, and I've still got the 2382.

— How about a 2635 printing terminal for only about $5800.

That would waste too much paper, and I couldn't use screen mode EDIT.

— Use your 2382 for editing and use the 2635 for printing.

But that would use up two mux ports and another very long cable.

A CHEAPER SOLUTION
A HP2382 terminal is a small screen terminal which can very easily fit on your desk and has capabilities very similar to a HP2624 terminal. Like most HP terminals it can be used in multipoint mode and as a result it must use tri-state drivers (or something similar) for asynch communications. This allows many terminals to be placed in parallel on the same interface.

WHAT TO DO
So why can't you place a RS232 printer in parallel with the terminal?

— Build (or buy) a RS232 double adaptor box.
— Disconnect the receive data line from the socket into which you'll plug your printer and connect it in series with a simple on/off switch.
— Buy yourself a RS232 printer (I bought an ANADEX DP9501 = ∿ $2400).
— When you want to print, just turn on the switch and whatever is received by the terminal is also printed on the printer.

DRAWBACKS
As the printer I obtained is only a 120 cps printer with a 2700 character buffer (about 4 pages), long listings will tend to overflow the printer's buffer.

The workaround you can use:
— slow the baud rate to match the printer speed.
— write a listing program to insert an appropriate delay at the end of each line.
— hit a key to get breakmode prompt.
  * Press (cntlX) (delete breakmode prompt from the printer's buffer).
  * hit the backspace or delete key to remove the (cntlX) from the computer's buffer.
  * wait until the printer catches up to the terminal and then press. (RETURN)
— disable the terminal from responding to the ENQ/ACK handshake by connecting the printer BUSY line to the terminals CLEAR TO SEND (CTS) line.

DISCLAIMERS
This set up does work and it's the one I've used to print this article, however I haven't tried it with any other terminal/printer combinations. If you can't make this work don't scream to HP for help as this is not a HP supported solution.
If you make a wiring mistake and blow something up, DON'T BLAME ME.

Corrado DiQual,
STANDARD TELEPHONES
AND CABLES P/L,
(02) 699 0044

Backup System and Group Cartridges Using Saver

```
:**BACKUP SYSTEM AND GROUP CARTRIDGES USING SAVER
:**SAVER IS USED INSTEAD OF THE OTHER TAPE UTILITIES BECAUSE IT ALLOWS YOU
:**TO SPECIFY WHICH FILES TO SAVE USING A WILD CARD SPECIFICATION
:**BUT MOST IMPORTANT IT ALLOWS YOU TO RESTORE INDIVIDUAL FILES FROM TAPE
:**
:**SAVER IS ALSO USEFUL FOR ARCHIVING FILES ONTO TAPE BY SPECIFYING TO
:**PURGE THE FILES JUST SAVED
:**
:**
:**SET SEVERITY TO INHIBIT ECHOING OF COMMANDS
:SV,2,,IH
:**SET UP LU10 AS A BIT BUCKET
:SL,10,0
:DP,MOUNT TAPE IN DRIVE,PRESS <RETURN> WHEN READY
:**
:**RUN SAVER WITH THE FOLLOWING OPTIONS
:**TT    TERMINAL IS TELETYPE. THIS TELLS SAVER NOT TO OVERWRITE THE SCREEN
:**SO,7 SORT TAPE DIRECTORY BY SECURITY CODE AND CARTRIDGE REFERENCE
:**SE,::AX  MASKED FILE SEARCH  FOR ALL FILES ON AX
:**T6    SAVE TYPE 6 FILES AS WELL
:RU,SAVER,TT,T6,SO,7,SE,::SY,SE,::AX,SE,::GL
:**NOW RUN READER TO GET A LISTING OF THE TAPE DIRECTORY ON THE LINE PRINTER
:RU,READR,LI:6
:DP,CHANGE TAPE IN DRIVE
:DP,REMOVE DIRECTORY LISTING FROM PRINTER AND STORE WITH MAG TAPE
:DP,PRESS <RETURN> WHEN READY
:**NOW WAIT FOR USER TO PRESS RETURN
:ST,1,10
:**REPEAT PROCESS WITH NEXT TAPE
:RU,SAVER,TT,T6,SO,7,SE,::42,SE,22,SE,::RV,SE,::CD
:RU,READR,LI:6
:DP,REMOVE TAPE FROM DRIVE
:DP,REMOVE DIRECTORY LISTING FROM PRINTER AND STORE WITH MAG TAPE
:EX,SP
:**CORRADO DIQUAL
```

RTE-6/VM BUG

PROBLEM

When switching to a new system or re-booting, the system file $SYENT would not be created on cartridge 2 as required.

CAUSE

Our system has a lot of EMA programmes, and as such we have several large Mother partitions. The remaining space is divided into normal 'BG' partitions, the largest of which is 20K.

During 'GENERATION' we sized the 'LOADR' to the max. size allowed, as advised in the generation guide, which was restricted to 21k by the 'RT6GN' programme. It appears that at bootup time 'FMGR' has a look for file '$SYENT' and if it doesn't exist, calls the 'LOADR' to create and seed or at least seed it. As the loadr will not run in a mother partition, unless assigned, and there was not a normal partition large enough it would not run when called by 'FMGR'. This error condition is not reported to the system console, nor is it documented, as far as we can see.

WORK AROUND

Do a 'slow boot' and either:

(1) re-size the loadr or
(2) re-define the partitions or
(3) assign the 'LOADR' to a mother partition.

David Ifield,
IFIELD ENG P/L,
(02) 651 2122

SPECIFICATIONS FOR SUBMISSION OF ARTICLES AND ADVERTISEMENTS

Crosstalk will be published each even numbered month. Articles and advertisements must be received by the appropriate group editor by the third week of the preceding month.

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,
HP1000 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.

Ms. Barbara Harrison,
Canberra Technical Users Group
C/- Australian National Parks & Wildlife,
4th Floor, Adelaide House,
Phillip, A.C.T., 2606.
Phone: (062) 897 919.

Mr Keith Crellin,
Queensland Technical Users Group,
C/- Cameron McNamara Pty. Ltd.,
131 Leichhardt Street,
Spring Hill, Qld., 4001.
Phone: (07) 228 9125.

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

H.P.D.C.U.G.V. advertisements only to:
Advertising Editor,
HP Desktop Computer Users Group (Vic.),
C/- 47 Bursania Ave.,
Ferntree Gully, 3156, Vic.

COMING EVENTS

Nov. 4:	H.P.D.C.U.G.V. meeting – Data acquisition using Desktop Computers.
Nov. 8:	RTE-6/VM Session Monitor Course, HP Adelaide.
Nov. 8:	HP 9845 Operating & Programming Course, HP Sydney.
Nov. 15:	RTE-A System User Course, HP Melbourne.
Nov. 15:	RTE-A System User Course, HP Sydney.
Nov. 22:	RTE-6/VM System Manager Course, HP Adelaide.
Nov. 22:	RTE-A System Designer Course, HP Melbourne.
Nov. 22:	RTE-A System Designer Course, HP Sydney.
Nov. 22:	HP 9845 Operating & Programming Course, HP Melbourne.
Nov. 22:	HP 85 Basic Programming, HP Sydney.
Nov. 29:	Pascal Programming Course, HP Melbourne.
Dec. 1:	HP 1000 Users Group (NSW) — Annual General Meeting.
Dec. 6:	PASCAL/1000 Course, HP Sydney.
Dec. 6:	IMAGE/DBMS Course, HP Melbourne.

CLASSIFIED ADVERTISEMENTS

FOR SALE

9845A Desktop Computer with '64' K memory & Mass Memory ROM
9885M Floppy disk drive (8")
9825A Desktop Computer with '24k' memory & lots of ROMS
9831A Desktop Computer with Mass Memory ROM
9869A Card Reader

Enquiries should be directed to:
Chris Simpson
Phone: (03) 859 6643

HP85 including all Roms except advanced programming.
82901M twin 5¼" floppy disc drive.
2631B printer.
9111A graphics tablet.
HP125 with BASIC operating system.

Enquiries should be directed to:
Mark Levenspiel
Phone: (059) 85 3731.

2 off HP 9111A Graphics Tablets (unused).
Price: approx. 25% off the HP listed price.
Contact: Ian Browning, (02) 699 0044.

WANTED

Cross-Assembler program (8080/85).
For use on HP85.
Call Ron Davis (03) 751 1526.

Serial and HP-IB interfaces for 9825/35/45.
I collect surplus desktop equipment.
Chris Simpson (03) 859 6643.

WANTED

11265 ROM for 9810 Calculator
Contact P. Rumpf, (03) 316 2469