Computer Systems

HEWLETT PACKARD

issue no. 5

Communicator 250

table of contents

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

system software

New HP250 Flexible Disc Drives and Media

by Gretchen Snowden

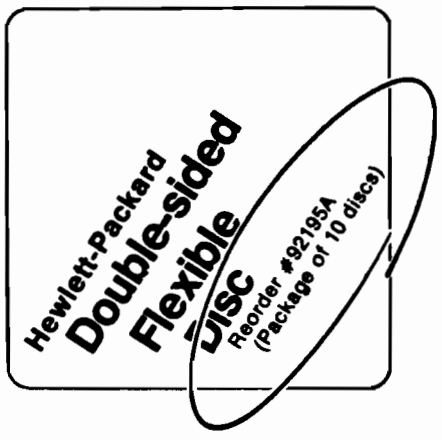
This fall, Hewlett-Packard made a hardware change in new HP250 systems being shipped. Because of supply difficulties, a new flexible disc drive unit is now being used. New media is also being used because the old media does not work in the new drives—it wears quickly.

The user should be aware of two important differences when identifying the new drives.

1) The "READY" and "IN USE" lights which were in the lower left-hand corner on the old drives are not on the new drives.
2) A label which has been placed in the lower right-hand corner of the new drives should help the user distinguish between the old and new drives. This label identifies the 9895K drives as well as specifying the acceptable media to be used in the drive.

Another major difference with regard to the new drives is that the new drives REQUIRE the new media. This requirement must be carefully noted. A large yellow caution sheet is being shipped with the new drives in order to designate which media can be used.

The new media can be identified by the number 92195A on the label. If you cannot locate this number, then it is not the new media and must not be used in the new drives.



Data on old media may be copied ONE TIME in a new drive and then should be discarded. Do not push your luck by trying it several times. IBM media may be used in the HP250 in conjunction with the MEDIA drom. Error rates should not be a problem although this cannot be guaranteed. IBM media has a hard surface and should not cause a wear problem.

The above information should answer all your questions on the new media and disc drives. If there are other areas of concern, contact your SE or CE.

Code Debugging Techniques

by Miles Kehoe

One of the most important phases of program development involves the debugging of a new applications program to verify its function under all anticipated conditions. This intensive QA is rightfully conducted after an application is considered "complete". However, the program author must perform several passes at debugging well before a program is finished, if just to verify operation of later sections of code. It is this on-line QA effort that will be discussed throughout the remainder of this article.

The HP250, designed as it is for the programmer as well as the end user, offers a host of tools for program checking. Some are similar to debug features on other computer systems, while others are unique to the HP250. We will discuss both, although reference will be made to existing documentation for the sake of brevity.

TRACE

The TRACE DROM is included with every Operating System disc. With this capability loaded, several additional commands are available to the HP250 BASIC:

TRACE
TRACE WAIT
TRACE PAUSE
TRACE VARIABLES
TRACE ALL
NORMAL

With these commands, the programmer can produce output to the SYSTEM PRINTER IS device as the program executes. These statements enable several most useful techniques in debugging running code, and are very well documented in Chapter 8 of the BASIC Reference Manual (HP Part 45251-90015). Browse through that document for a good explanation of TRACE.

HALT (Single Step)

The HALT key provides the programmer with a single step capability, permitting code to be executed one line at a time. When a non-RUN-ONLY program is loaded, pressing the HALT key will cause the lowest numbered line to display on the screen. A second press will cause the system to execute that line, and to display the next (sequential) line about to be executed. By continuing to press the HALT key, the programmer may step through the program to verify anticipated operation; additionally, variables may be examined, line statements changed, and program flow altered without the need to re-run the program. These procedures will be examined in greater detail later in this article.

HOP (Macro Step)

In Operating Systems after OS 2.0 (also known as Rev D), a new keystroke command has been implemented. Known as the "HOP" key, this function permits the programmer to skip over all lines of code until the next sequential line is about to be executed. Implement this feature by using both the "control" key (CTRL) and the rightmost ENTER key together. (This applies only to the rightmost ENTER key.) When this is done (for example when a CALL is about to be executed), the system will execute the CALL, perform all instructions in the subroutine normally, and stop prior to executing the line following the CALL. This functions exactly like the HALT single-step for most lines of code, but is most useful when a known section of code is about to be executed.

On-Line Use

Occasionally it is desirable to use a combination of the above features in debugging code, and there are some tricks which can make the task easier.

Reviewing Program Execution

Often, a program is known to function up to a certain softkey driven menu, but the programmer wishes to single step from the softkey interrupt. If the system is executing a WAIT for a key interrupt, you can force the system into single step by using a shifted HALT to terminate the WAIT. Pressing the HALT key once again returns to the WAIT state, except that any future interrupt (i.e., a softkey press) will continue in single step mode. After the press, the line reference in the ON KEY statement will be displayed. To execute that line, use HALT or HOP as before.

Two of the most useful applications of HOP involve calls to subroutine and functions, and execution of loops. When a subroutine or function is known to function properly, there is no reason to single step through its statements. This is expecially true if the subroutine handles FORM input/output. Rather than use the HALT key when the CALL is about to be executed, use the HOP key (CTRL-ENTER). This will cause the call to be executed, and to continue execution in normal operation until the next line after the CALL is about to be executed. Remember, if you use a function in a DISP or PRINT statement, the function will be executed! Use the HOP key prior to execution of the entire line.

Often there will be a need to verify proper operation of a FOR-NEXT loop for one or two passes only. If the first two times are successful, there is no need to perform the entire loop one step at a time. By using the HOP key when the final statement in the loop is about to be executed (NEXT, in this example), execution of the loop will continue until the next line is about to be executed.

If a loop or structure is exited abnormally (i.e., via a GO TO), program execution will continue until the original next line is about to be executed. This can be useful to establish that program flow is as anticipated in the case of a program producing unanticipated result.

Changing Execution Sequence

When used in a program, the GO TO statement alters normal flow of execution. Similiarly, when used in "Immediate Mode" as a command, GO TO will change the next line to be executed. This is often useful while a program is still under development, or to recover from an untrapped error during program use.

To re-start a program, a GO TO may be used to direct program execution to the first statement in a program. However, this is not the same as re-running a program. To do this, execute a STOP command. This will re-set system execution pointers, and a HALT or HOP key will perform pre-RUN initialization. After an immediate mode GO TO, execution may be resumed via HALT or HOP as described, or via CONT to continue normal execution.

Screen I/O

Because of its nature, single step can cause some strange effects on the CRT screen. A CURSOR positioning statement will move the cursor; however, the next line to be executed will be displayed immediately. This may cause unusual operation of screen functions, but is consistent with the use of single step.

When an INPUT statement is executed in single step (caused by either a HALT or a HOP press), the prompt is issued and the system enters an input state. If a line is entered via the ENTER key, that value is assigned to the target variable and the next statement displayed. If the HALT or HOP key is pressed instead of entering a value, program execution continues but without affecting the contents of any variables referenced in the INPUT statement. This could cause erroneous results, and should be watched for. If this does happen, you may of course assign a value in immediate mode and continue program execution normally.

Immediate Mode File Access

Often it is convenient to access a file or a data base in immediate mode, positioning file pointers manually. This can be done easily; however, variable types and dimensions must often be declared prior to reading a file. Unfortunately, such declarations cannot be done outside of a program environment. This restriction is overcome by entering program line which declares the desired attributes of variables, and then single-stepping through the program until all declarations have been executed. For example, type in this program segment:

```
1 DIM Buf$[80]
2 REAL R
3 SHORT S
4 INTEGER I
```

By pressing the HALT key five times, these five variables become available as declared. At this point, the file statements desired may be typed as commands and file accessed. (Remember to use the TYP function if you don't know what type of data is in the file.)

Some Things to Watch For

Before we finish, let me point out some cautions. First, practice these techniques on programs you know. Don't risk destroying a program accidently because you are unfamiliar with the techniques.

If, as you execute a program, you change one or more lines of code, remember to RESTORE the changes to the disc. However, programs will often delete lines from memory to conserve space. If your code uses such techniques (as many HP applications do), be sure NOT to RESTORE a program unless you are positive that no lines have been deleted (or added!) during execution.

Also, you will occasionally get an ERROR 42 when you try to replace or re-enter a line. This error results when the line you are changing is still being executed or when the line is part of an active subroutine or function. To replace the line, enter a STOP or do a SUBEXIT, SUBEND, or RETURN to exit the busy line. Then re-enter the line.

These tools can be most useful in tracing program execution or in debugging code. As with any tool, you can hurt yourself occasionally until you are familiar with its function. Be sure to have back-up copies of programs you edit, and be certain to store changes you may make on the proper disc file. With these tools, you can increase your programming efficiency!

Error 88—Causes and Recovery Procedures for Flexible Discs

by Chris Eckhardt

In an attempt to make error messages more meaningful we are revising documentation of some of the system error messages. The revised documentation will include both an explanation of the error and a recovery procedure(s).

Tying in with this issue's theme of MEDIA, Error 88 documentation is included below, giving the user a clearer understanding of the probable causes and remedies if the error is encountered on a flexible disc.

Error 88: Read data error—Unable to access file(s) on the disc.

Cause: The system is not able to read the data from the disc.

Note: Because of the extreme sensitivity of the new harder surface flexible discs, Error 88's can be caused if dust particles reside on the disc or in the drive. As a preventive measure, it is VERY important to care for discs properly and to make an effort to keep the computer area as clean as possible.

Remedy: Three alternatives exist for recovery for an Error 88 on a flexible disc. Choice is dependent upon whether a duplicate set of the files contained on the worn disc exist on a Backup disc, and if so, whether or not the backup disc is current.*

For each recovery alternative it is suggested that the disc drive be cleaned both before and after the recovery procedure.

*A backup disc is considered "current" if the data on the disc has been updated each time the data on the main (now defective) disc is changed or updated.

Recovery Alternative A:

A "current" backup of the files contained on the defective disc is available.

Procedure:

Using your application software operators guide as a reference, identify if a procedure exists which utilizes a backup disc when attempting to recover from an error. This function will allow the files to be restored to the condition they were before the error occurred. If this procedure is not available as part of the application software, the user must revert to Alternative B.

Recovery Alternative B:

This alternative utilizes the MEDIA TEST utility to attempt to recover from the ERROR 88. Because this procedure does not utilize a backup disc as part of the recovery, it should be chosen when the backup set of data is not considered "current".

The MEDIA TEST utility (described in the System Operators Guide, Chapter 7) allows the system to verify (scan) the defective disc, automatically logging a record of any disc errors encountered. The errors logged indicate the location of the worn or defective areas that exist within files on the disc.

After the errors are identified, the MEDIA TEST RECOVERY function (described below) may be tried in an attempt to rewrite (copy) the files. This function does not require a second disc, the files are merely rewritten to the same disc. There is a possibility that the files will not be able to be copied using the Recovery function. If they cannot be rewritten, RECOVERY ALTERNATIVE C must be taken. If the files can be rewritten, the Recovery is successful and the error is corrected. In the case of a successful recovery, be sure to take note of the "IMPORTANT" message below.

Procedure:

1) Run "TEST,UTILITY" (Chapter 7, System Operators Guide)
2) Press the MEDIA TEST softkey to select and test the disc.
3) Press VERIFY to start the media verification routine.
4) Press W/O RECOVERY so that the system will automatically log any disc errors encountered. The final display will indicate the number of errors logged and the location on the disc where each error occurred.

5) For those errors logged which were followed by "POSS" as an error type (see System Operators Guide, p. 43 for an explanation of the format of entries in the error list), an attempt should be made to rewrite (copy) the file back onto the worn disc in order to correct the error. To do this, reselect the VERIFY softkey, only this time press W/RECOVERY. If the Recovery fails, the files are unable to be copied and the Error 88 will have to be corrected using the procedure outlined in Recovery Alternative C.

   Those errors logged which do not contain a "POSS" in the error message are unrecoverable. In order to correct them, the user must revert to Recovery Alternative C.

Important:

The detection of an ERROR 88 may indicate that the media is worn or defective. If the VERIFY or RECOVERY is run successfully, the media should immediately be backed-up (copied) to a known good media to prevent further errors from occurring.

Recovery Alternative C:

No "current" backup disc is available, and damaged file(s) cannot be copied using the MEDIA TEST utility.

Procedure:

1) Using the worn or defective disc, copy the unaffected files to a clean, initialized disc. The XCOPY system utility or COPY statement can be used to copy the files. Keep a record of which file(s) cannot be copied.
2) Using the most current backup available, copy the files that were unable to be copied from the worn disc onto the new media (FCOPY or DUPL utility can be used to copy the file).
3) Having copied all files to the new media, these files must now be updated with all the transactions which exist on the defective disc, but not on the backup disc.

customer corner

Improving Image Calculated Access Times

by Frank Heartney

Several customers have mentioned that for certain master sets, the number of synonyms seems rather high. The following procedure describes a method for lowering the number of synonyms.

Introduction

The calculated access method for IMAGE master data sets is supported by a hashing algorithm. Image takes a key item value and calculates a logical record number from it. Different keys that map to the same logical record number are called synonyms and are linked together in a synonym chain. Some IMAGE users have experienced a problem where too many keys hash to the same logical record leading to performance degradation. The hashing function in IMAGE cannot be changed without invalidating existing data bases but a preliminary transformation can be applied to user key values to improve the randomness of the hash.

This article first describes the existing IMAGE hash function and then provides some BASIC functions that will make user keys more hashable along with some suggestions for how to introduce these transformations into an existing data base.

The Image Hash Function

Whatever the data type of the key value, the hash function treats it as a sequence of 16 bit integers or words. All the words in the key are combined in a procedure known as EXclusive OR. This 16 bit result is then rotated to the right one bit position, with the least significant bit rotated around to the most significant bit position. This value is then divided by the capacity of the data set and the remainder of this division forms the logical record number for this key value.

As an example hash the 6 byte "HP250" into a data set with capacity of 2000 records.

ASCII	BITS	COMMENT
"HP"	01001000 01010000	standard ASCII
"25"	00110010 00110101	representation
" 0 "	00110000 00100000	
	01001010 01000101	bit-wise EXclusive OR
	10100101 00100010	right circular rotation
	42274	convert to unsigned 16 bit integer 42274 mod 2000
	274	

Thus the key value "HP250" will hash to logical record number 274. This hash function will not hash some kinds of keys effectively. The worst problems will be encountered when the key is a character string and only a limited set of characters is used in the string.

For instance if strings are limited to the ASCII characters "0". . ."9" then the upper four bits of each byte will be the same and will not contribute to the randomness of the hash. If the keys are limited to the characters "A". . ."Z" then the upper three bits of each byte are the same.

Transforming the Keys

A simple pair of BASIC functions has been developed to take a customer key value and transform it into something that occupies the same amount of space but that hashes better. Another BASIC function is provided to recreate the original key value. The function takes over every other word in the key and circularly rotates it 4 bit positions. This has the effect of randomizing every bit position so that maximum randomness is obtained. Copies of these functions are following.

```
1000 ! PROGRAM TO ILLUSTRATE A TRANSFORMATION ON HASH KEY VALUES TO
1010 ! IMPROVE THE RANDOMNESS OF THE HASH. EVERY OTHER WORD OF THE KEY
1020 ! IS ROTATED TO THE RIGHT FOUR BIT POSITIONS. THE RATIONAL FOR THIS
1030 ! TRANSFORMATION  IS THAT A SET OF CONTIGOUS ASCII CHARACTERS WILL
1040 ! ALL HAVE THE SAME HIGH ORDER FOUR BITS, SO THAT THESE BIT POSITIONS
1050 ! CONTRIBUTE VERY LITTLE TO THE HASH. THIS TRANSFORMATION WAS APPLIED
1060 ! TO AN AUTOMATIC MASTER WITH AN AVERAGE SYNONM CHAIN LENGTH OF 28.8.
1070 ! AFTER THE KEYS WERE TRANSFORMED, THE AVERAGE SYNONM CHAIN LENGTH WAS 1.
1080 !
1090      DIM Buf$[256]
1100      INTEGER S(9)
1110      DISP "/9"
1120      DBGET (B1$,Ds$,2,S(*),"@",Buf$,0)
1130      WHILE S(0)=0
1140           Inv_no$=FNPrehash$(Inv_no$)
1150           DBPUT (B2$,Ds$,1,S(*),"@",Buf$)
1160           IF S(0) THEN GOSUB Dberror
1170           DBGET (B$,Ds$,2,S(*),"@",Buf$,0)
1180      END WHILE
1190      IF S(0)<>11 THEN Dberror
1200      DISP
1210      DISP "DONE  "
1220      END
1230 Dberror:CURSOR (1,20)
1240      IF S(0)=43 THEN RETURN
1250      DISP "DATA BASE ERROR!!"
1260      DISP "S(0): ";S(*)
1270      STOP
1280 ! PREHASH TAKES THE ORGINAL KEY AND CONVERTS IT INTO A FORM THAT WILL
1290 ! HASH BETTER. NO INFORMATION IS LOST AND THE ORGINAL KEY CAN BE
1300 ! RE-CONSTRUCTED WITH FNPOSTHASH.
1310 ! THIS COPY OF PREHASH WORKS FOR 10 CHARACTER STRINGS.
1320      DEF FNPrehash$(Key$)
1330           INTEGER A(4)
1340           DIM Trans_key$[10]
1350           UNPACK USING P1;Key$
1360           FOR I=0 TO 5 STEP 2
1370                A(I)=FNRar(A(I))
1380           NEXT I
1390           PACK USING P1;Trans_key$
1400           RETURN Trans_key$
1410 P1:       PACKFMT A(*)
1420      FNEND
1430      DEF FNRar(INTEGER X)
1440           T1=X+65536*(X<0)
1450           T1=INT(T1 DIV 16)+4096*(T1 MOD 16)
1460           RETURN T1-65536*(T1>32767)
1470      FNEND
1480      DEF FNRal(INTEGER X)
1490 ! FNRAL ROTATES AN INTEGER TO THE LEFT 4 BIT POSITIONS.
1500           T1=X+65536*(X<0)
1510           T1=INT(T1 DIV 4096)+16*(T1 MOD 4096)
1520           RETURN T1-65536*(T1>32767)
1530      FNEND
1540 ! POSTHASH RE-CALCULATES AN ORGINAL KEY VALUE FROM THE TRANSFORMED KEY.
1550 ! THIS COPY OF POSTHASH WORKS WITH 10 CHARACTER STRINGS.
1560      DEF FNPosthash$(Key$)
1570           INTEGER A(4)
1580           DIM Trans_key$[10]
1590           UNPACK USING P2;Key$
1600           FOR I=0 TO 5 STEP 2
1610                A(I)=FNRal(A(I))
1620           NEXT I
1630           PACK USING P2;Trans_key$
1640           RETURN Trans_key$
1650 P2:       PACKFMT A(*)
1660      FNEND
```

There is a serious drawback to this procedure however. The keys which have gone through the prehash routine can no longer be used in the FIND and SORT BY statements or by QUERY. This drawback can be overcome by retaining the original value as another item in the set. The FIND, SORT and QUERY could use the original value.

These functions were tested on a user data base with 20000 entries. The original keys hashed very badly, producing an average chain size of 28.8 entries. After transforming the keys and writing them out to an empty copy of the data base, the average synonym chain size was reduced to 1.9 entries.

A sample program is provided to determine the average synonym chain size for any data base. If the chains are longer than 8 or 10 entries then your application would probably benefit from the use of transformed key values.

Implementation

If you are starting a new application, data bases and programs can be designed to handle the transformed keys. If you wish to convert an existing system, you must change the code and then re-build the data base. This will have to be done with a BASIC program which would take the data from the original data base, do the transformation and put the data into the expanded data base. Before these changes are made, the system should be carefully analyzed to verify that all functions (THREAD, SORT, etc.) will work on the transformed keys.

```
900   ! PROGRAM TO DETERMINE THE AVERAGE LENGTH OF A THE SYNONYM CHAIN FOR A
910   ! MASTER DATA SET. THE LONGER THE CHAINS THE MORE DISK ACCESSES ARE
920   ! REQUIRED TO GET A RECORD IN CALCULATED ACCESS MODE 7. AN IDEAL CHAIN
930   ! SIZE IS ONE OR TWO AND ANYTHING MORE THAN 10 ENTRIES PER CHAIN IS
940   ! EXECISIVE.
1090      DIM Buf$[256]
1100      INTEGER S(9)
1110      DISP "/9"
1120      GOSUB Initialize
1230      DBGET (B$,Ds$,2,S(*),"@",Buf$,0)
1240      WHILE S(0)=0
1250           IF S(5) THEN
1260                Num_chains=Num_chains+1
1270                Chain_len=Chain_len+S(5)
1280           END IF
1290           Entries=Entries+1
1300           CURSOR (50,10)
1310           DISP Entries
1340           DBGET (B$,Ds$,2,S(*),"@",Buf$,0)
1350      END WHILE
1360      IF S(0)<>11 THEN Dberror
1370      DISP
1380      DISP "DONE  # OF CHAINS & AVERAGE CHAIN SIZE:"
1390      DISP Num_chains,Chain_len/Num_chains
1400      END
1410 Dberror:CURSOR (1,20)
1430      DISP "DATA BASE ERROR!!"
1440      DISP "S(0): ";S(*)
1470      STOP
1500 Initialize:DISP " SYNONM CHAIN COUNT PROGRAM"
1510      INPUT "ENTER THE DATA BASE NAME ";B$
1520      B$="  "&B$
1530      INPUT "ENTER DATA BASE PASSWORD";Pw$
1540      CURSOR (1,YPOS-1)
1550      DISP "
1560      INPUT " ENTER THE NAME OF A MASTER DATA SET ";Ds$
1570      DBOPEN (B$,Pw$,1,S(*))
1580      IF S(0) THEN Dberror
1590      DBASE IS B$
1600      IN DATA SET Ds$ DIM ALL
1610      CURSOR (20,10)
1620      DISP "ENTRIES EXAMINED"
1630      RETURN
```

RJE Being Shipped

by Gretchen Snowden

The RJE/250 software package and INP/250 hardware are now being shipped. In case you are not familiar with those acronyms, let me explain.

RJE stands for Remote Job Entry and refers to the capability to communicate with a large mainframe system and use job control on the system. This is accomplished on the HP250 by emulating either an IBM 2780 or 3780 data entry station. Then the HP250 can communicate with any large mainframe that has a batch spool monitor and accepts 2780/3780 input. Since a 2780/3780 can also communicate with another 2780/3780, the HP250 using RJE can communicate with another HP250 or HP3000, for example, that is also emulating one of those machines.

INP stands for Intelligent Network Processor and refers to the hardware used by RJE. The hardware consists of a board for the card cage and a back panel which provide synchronous communications for the HP250. Now in addition to asynchronous communications to remote consoles and terminals, the HP250 can provide increased accuracy of synchronous communications.

With an order for product 45122A, a customer receives the INP hardware, RJE software, utilities, diagnostics and documentation.

The Current Operating System

by Gretchen Snowden

The last issue of the A.03.00 Communicator referred to operating system A.03.00. When the new operating system and update package arrived, you probably discovered that the version was in fact A.03.01 which was fixed prior to being shipped. In general, the last digit of the operating system will be increased for an operating system that contains new features. In keeping with the philosophy, the next major release will be A.04.00.

You will also note that the Query and Utilities discs remain version A.03.00 because no change was made to them.

As this Communicator goes to press, another operating system is nearing release. This will be O.S. A.03.04 and it is primarily a bug-fixing release. However, included will be support of the HP 2608 high-speed printer and a new binary for an ACCEPT statement which will allow data base passwords and similar information to be accepted without echoing to the screen. More information on these features will be included when the system is distributed.

service information

Handling Flexible Discs

by Hal Good

The flexible disc is basically maintenance free, but it is delicate and MUST BE HANDLED CAREFULLY. Remember, the disc contains your valuable data and programs and should be treated accordingly. A good rule of thumb is to treat your disc as you would a valuable record album. Here are some specific Do's and Don'ts to avoid loss of data or damage to your discs.

Even a little carelessness in disc handling can dramatically reduce the life of the disc!

Do

Return Disc to Storage Envelope When Not in Use.

This is the single most important thing to remember about handling your disc because it prolongs disc life by protecting it from dust and scratches. Between uses, discs should be stored upright in a dust free container. The box the discs are shipped in, or a similar container, is a good choice.

Remove Disc from Drive When Not in Use.

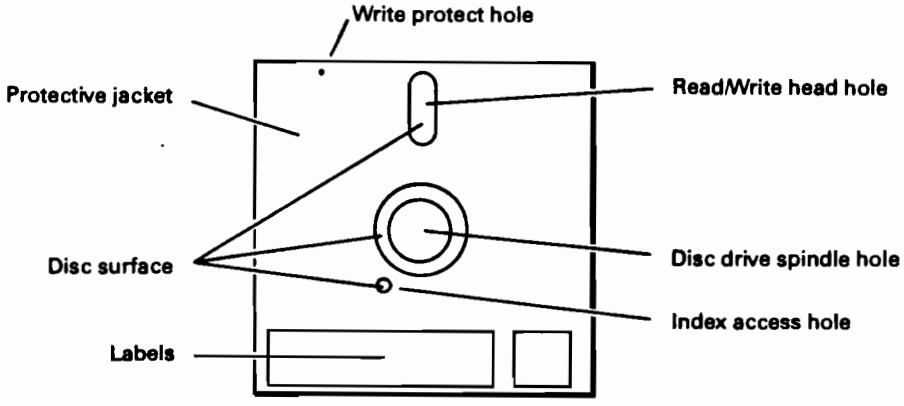
Remove the disc completely from the drive when access is not needed for an extended period of time. The disc continues to rotate as long as it is in a drive with the door closed, even if it is not accessed. This rotation will eventually wear the disc out over long periods of time.

Operate Your System in a Clean Environment.

Airborne contaminates and particles accidently dropped onto the disc will cause your disc to wear out prematurely and may cause unreliable data storage and retrieval operations. Some of the most common contaminates are DUST, SMOKE, ASHES, ERASER CRUMBS, and BREAD CRUMBS. Chemical vapors may also cause premature wearout.

Maintain Proper Temperature and Humidity.

The proper operating range is 10°C (50°F) to 40°C (104°F) and 20% to 80% relative humidity. While temperature is usually easy to control it may be necessary to make special provisions to keep the humidity in the proper range. Although the disc will continue to operate outside the normal humidity range, it will wear out more quickly and will have a higher error rate.



Flexible Disc Locations and Nomenclature

Backup Discs Frequently.

There is always a chance of losing data when mass storage devices are accessed. There are many causes in any computer system—a programming bug, operator error, power failure, or hardware failure. In the case of flexible discs, another mode is possible—media failure from contamination or wearout. Your only protection against data loss is frequent backup of your files.

Replace Discs Frequently.

Although discs are designed to provide several million revolutions of useful life, they will eventually wear out. The life of a disc is VERY dependent on how much it is used. A disc used sparingly (less than 20 minutes a day) should last over a year. A disc that is used heavily (more than 2 hours a day) should not be expected to last more than 3 months. To be safe, you should copy your data to a new disc and discard the old disc every 3 months for a heavily used disc or at least once a year, even for lightly used discs.

Use Only HP-Approved Flexible Discs.

The performance of non-approved discs cannot be guaranteed.

Avoid Magnetic Fields.

Since the data is stored as a pattern of magnetic fields on the disc, it can be erased by an external magnetic field. Avoid placing a disc near power transformers, magnets or large disc drives. Additionally, while HP goes to great lengths to confine the magnetic fields produced by it's CRT deflection shields (so well that some of our disc drives are mounted in the same cabinet as the display) CRT's with magnetic deflection systems have been known to wipe out discs, and it is a good idea to avoid placing discs on top of CRT's.

Use a Felt Tip Pen to Label Your Disc.

Use a soft felt tip pen to label your disc, and be careful to write only in the label area. Avoid the exposed media while labeling the disc. If possible, write on the large labels provided BEFORE applying them to the disc.

Don't

Do Not Touch the Surface of the Disc.

The thickness of the fingerprint is enough to lift the head off the disc and cause errors. The oils in the fingerprints will also collect dust which can cause a disc to wear out sooner than it normally would.

Do Not Bend or Fold the Disc.

The disc is flexible, but will not operate if it is creased. Using ball point pens, rubber bands, paper clips, etc., can crease the disc.

Do Not Try to Clean a Disc.

The inside surface of the disc jacket is covered with a special material that cleans the disc as it rotates. Any other method of cleaning may cause solvent damage to the media or scratch the disc, causing loss of data. If a disc becomes dirty or scratched, immediately transfer the data to a new disc and dispose of the old disc.

Never Remove Discs from Their Sealed Protective Jackets.

1980 Media Replacement Program.

The following is a copy of the letter which has gone out with the media cleanup kit to HP250 customers. This letter outlines the current MEDIA Exchange Policy.

HEWLETT PACKARD

DESKTOP COMPUTER DIVISION · 3404 E. Harmony Road, Ft. Collins, Colorado 80525, Telephone 303 226-3800

"CUSTOMERS"

To provide products and services of the greatest possible value to our customers, thereby gaining their respect and loyalty.....

(From Hewlett-Packard Corporate Objectives)

24 September 1980

Dear 250 Customer,

Hewlett-Packard expresses its appreciation for the outstanding customer acceptance of the System 250.

Since the introduction of the System 250, we have continued to invest engineering resources to make it an even more reliable product. It was our contention that we could make a quantum jump in reliability if we could improve the mass storage subsystem within the 250.

We feel that with the further improvements in media and electronics that have taken place over time, we are now in a position to significantly impact the performance of your system.

Since Hewlett-Packard values you as an extremely important customer, we are sending out our Customer Engineer to you today to upgrade your system and bring it up to the levels of performance that System 250's are capable of today.

One of the key components of this program is the use of new media - the HP 92195A. The head/drive/media interface is an extremely crucial one and your Customer Engineer will be giving you complementary pieces of this new media for you to record on. At the same time, he will check out and clean your mass storage subsystem and update your mass storage firmware. In addition, he will go through all your media and identify those pieces that provide questionable performance. Having identified "bad" media, he will put caution stickers on these for your benefit. These actions will ensure that your system takes advantage of all the improvements that we know of today.

We would like to request you to <u>immediately</u> take all of the data/programs you have on the "bad" media and record them on your new media. Please continue

to use only the HP 92195A for all your new media requirements. By using the 92195A you will help us to help you in achieving optimum system performance.

We cannot promise that the new media will work forever. Flexible discs wear out much like the tires on your car. What you can expect, however, is that your time between media changes (much like the time between tire changes) will be longer now. An additional benefit is that our new media is also a better match for the head and should be easier on your whole flexible storage system.

I have included media handling guidelines with this letter for your benefit. Following these guidelines carefully, will enable you to get the best possible performance from your system.

Continuing to assure you of our support.

Sincerely yours,

HEWLETT-PACKARD COMPANY

Srini Nageshwar

Srini Nageshwar
Marketing Manager
Greeley Division

SN/bb
Attachment as stated

Getting Started on an HP 7910

by Doug Hoffman

The following procedure outlines the steps necessary for putting a system on a HP 7910. Section A (Steps 1–8) describes the initialization procedure; therefore, if the HP 7910 has already been initialized, begin with Step 12. If the system has never resided on the HP 7910, the switch on the Processor Board must be set to 0 or 3 and the device number for the HP 7910 must be 7. Contact your CE for further information.

Section A: Initializing an HP 7910

1. Insert the Operating System flexible disc (Version A.03.01 or later)
2. Insert the current UTILITIES flexible disc.
3. RUN "INIT"
4. Press "INITIALIZE"
5. Select "L2"
6. Press "DIRECTORY 2:800"
7. Enter the value "4" for number of tracks in main directory.
8. Press "CONTINUE"
9. Compare the list of defective tracks from INIT to the list on the HP 7910 casting (below the left side of the DSU CONTROL an DSU ANALOG PCA's).
10. If any tracks on the casting label were not spared by INIT, press "SPARE TRACK" and enter the track numbers.
11. Press "EXIT PROGRAM"

Section B: Copying a System to HP 7910

12. RUN "ROUTIL"
13. Insert the Operating System flexible disc. Press "SYSTEM & DROMs"
14. Press "COPY"
15. Press "SYSTEM:F2"
16. Press ":L2"
17. Press "COPY ALL"
18. Press "EXIT"

Section C: Copying RUN-Only Programs to HP 7910

19. Press "COPY"
20. Press "SYSTEM:F2"
21. Press ":L2"
22. Press "COPY ALL" or select desired RUN-ONLY programs (CONFIG, etc.)
23. Press "EXIT"
24. Press "EXIT ROUTIL"

The system is now ready to power-on from the HP 7910.

NOTE: If you wish to copy your UTILITIES and QUERY to the HP 7910, use Section C replacing the flexible disc before pressing "COPY".

How to Identify File Name Using a Specified Address

by Doug Hoffman

The process of finding which file is at a specified address consists of two parts. The first part is an arithmetic computation of the relative record number. The second part involves comparing this sector number to the catalog in order to find the file.

The following table gives the constant values to be used along with the HEAD, CYLINDER and SECTOR values (obtained from VERIFY or INIT) for the computation.

	H	C	
:F2	30	60	**NOTE:** HEAD value will be
:L2	32	64	1 or 2, values of 0 and 1
:C2	48	96	should be substituted.
:D2	48	96	

Record number =
(H*HEAD) + (C*CYLINDER) + SECTOR

For example:

On a flexible disc, if an error is detected by VERIFY at 1:20:4 the record number would be computed as follows.

HEAD = 1; H = 30
CYLINDER = 20; C = 60

Record number = (30*1) + (60*20) + 4 = 1234

To find the file which contains the error, compare the computed record number to the address listed in the CATALOG of the disc. Given the following catalog, the file would be MFIG. This is determined by finding the last file which begins before the address. MFIG starts at 1200, while the next file (RFIG) begins at 1254.

NAME	PRO	TYPE	REC/ FILE	BYTES/ REC	ADDRESS
SYSTEM:F2,6,0					
SYSTEM	*	SYST	600	256	1609
RIO		DROM	18	256	2209
TIO		DROM	19	256	2227
COPY		DROM	4	256	2246
CONFIG	RO	PROG	94	256	1060
AFIG	RO	PROG	47	256	1153
MFIG	RO	PROG	54	256	1200
RFIG	RO	PROG	79	256	1254
ROUTD	RO	PROG	43	256	1333
ROUTL	RO	PROG	73	256	1376
ROUTK	*	KEYS	1	256	1449
ROUTIL	RO	PROG	86	256	1450
FCLEAN	RO	PROG	46	256	1536
TEMP1		PROG	27	256	1582
MMCMDS	*	BPRG	20	256	2458
DUPL	RO	PROG	77	256	2478
LARGEST AVAIL SPACE			1945		
TOTAL AVAIL SPACE			3153		