

## A WALK ALONG DOS MEMORY LANE

By Robert Parkinson

The Intel 8088 processor chip used in the original IBM PC and XT could, for internal design reasons, address only 1MB of memory. This 1MB is actually 1,048,576 bytes, but who counts?

When IBM adopted Microsoft DOS as their operating system, they inherited a memory management system in which all memory is divided into 64KB segments. Each segment is subdivided into 16KB "pages", then into 16-byte "paragraphs", then into 4-byte "double words" and again into 2-byte "words". Thus the 1MB 8088-addressable limit translates into 16 segments. Of these, IBM reserved the top six segments (384KB) for system use.

At that time, it was thought that no user would ever possibly need more than the remaining 640KB of memory. My how times change! A few segments in the upper area were "reserved for future use" by IBM. Even today, some are still unused (at least by IBM).

The usual ways of referring to DOS memory are "low", "base", "conventional" or "user" memory under the 640KB limit, and as "high" or "reserved" memory between that limit and the 1MB addressable limit. Of course, some of the very bottom portion of low memory is also used by the system. The amount of memory used depends heavily on how complex a system configuration you create through your CONFIG.SYS and AUTOEXEC.BAT files, but it may easily take as much as 60KB or 10% of the available low memory space.

One question that users often ask is, "How is it that you are talking about 1MB of memory when I know that my machine only has 640KB of RAM?". Simply put, all memory isn't necessarily RAM, and all RAM isn't necessarily on your system motherboard. Read-Only-Memory (ROM) is

memory by any definition. The casual user may be aware of whatever combination of RAM chips his system holds, but seldom consciously thinks about ROM. Even if the operating system cannot write to these memory locations, it must be able to address them in order to read the code contained there.

Your system, if it has a hard disk drive and a colour monitor, will have ROM in at least three areas: your system ROM, your hard drive ROM and your display or video ROM. It may well have more if you have a specialized floppy drive controller, a network board or one of the other ROM-based add-ons. You may also have a video adapter with between 256KB and 1MB of its own RAM. Remember, in order for DOS to use these additional pieces of memory, their addresses must all fit, through whatever fancy footwork is necessary, into that 384KB of high memory.

In the case of your video board, the on-board controller will use some proprietary page-mapping scheme to fit the addressing of this hard-wired video RAM into the smaller allocated text and graphics address spaces in your high memory between 640K and 1MB.

We have seen how the "old" 8088 processor handles memory, but what about the 80286, 80386 and 80486? The successor to the 8088, the 80286 processor, can address 16MB of memory, while the newer Intel 80386 and 80486 processors can address 4GB (or, roughly, four thousand million bytes). This is all very nice, but can DOS use this additional memory? The short, but misleading answer is no.

To ensure full backward compatibility, Microsoft and IBM avoided a complete redesign of DOS, retaining the original 1MB DOS-addressable limit. What can these newer processors do with all those extra megabytes? If they are running the DOS operating system, they can't do much

(Continued on Page 2)

## NEXT MEETING

The next meeting of the Ottawa PC Users' Group will be held on Wednesday, October 31, 1990. This month's guest speakers will be Colette Lacroix from Ventura Software and Peter Heney and YS. Young from New Vision Technologies Inc.

Colette will give a demonstration of Ventura Gold for MicroSoft Windows 3.0. Peter Heney and YS. Yoong will talk about PRESENTATION TASK FORCE. A draw for door prizes will be held at the end of the evening.

\*\*\*

## FROM THE EDITOR

Hi folks! At the last General meeting, the new Constitution was adopted with an overwhelming majority. I'd like to thank all those members who took an active part in the process of amending it, especially Eric Clyde whose valuable contribution should not go unnoticed.

Our appreciation also goes to Steve Zucker of Borland who gave a fine presentation of Paradox, Quattro and C++. Happy reading!

## CONTENTS

Remarks in Batch Files	2
OPCUG Flea Market	3
Back to BASICS	4
Computer Translations	5
Bartender Wanted!	5
Beginners' Corner	6
PUB TALK	6
Executive	8
Membership Application	8

without special hardware and/or software.

The most common way of gaining more memory, although not without pitfalls, is to use expanded memory. I don't propose to get into the details of extended (above 1 MB) memory or expanded memory. Let's just say that if you have unused memory between 640KB and 1MB, or have an 80286 or 80386-based system with more than 1MB of memory on your motherboard, or have an add-on expanded memory board, there are many software drivers that can convert that memory to a form usable, in a limited way, by your programs.

However, this does not in any way eliminate the DOS-addressable limit of 1MB. It merely uses memory-page switching techniques to trick DOS. More importantly, your application programs can't normally use this expanded memory unless they have been designed from the ground up to do so. Some people may take exception to this statement and say that any programs handled by "QRAM", "QEMM", "Software Carousel", "386Max", etc., do not require such a design.

My answer to this is that these very useful utilities were designed from the outset to use expanded memory, and they merely serve as the "carriers" or "warehouse managers" for other programs that do not inherently have this capability. However, now that EMS 4.0 has become a well-established standard, more and more application programs, including many shareware programs, are being revised to make use of expanded memory. Some programs make use of it for data only, others for both data and program code.

If you want to exceed the 640K DOS limit, other than through the use of expanded memory, you will have to adopt one of the following expedients, which are listed here in order of ease of use for the average user:

- Application programs that use "DOS Extenders" to do machiavellian tricks with extended memory and the protected mode of the 286, 386 and 486 processors. Of these, the Phar Lap 386/DOS Extender is the best known, although there are a number of others for all of these processors. An excellent article on these DOS extenders can be found in *BYTE*, April 1990, Page 287
- Microsoft Windows. The earlier versions of Windows were not too well received by users. However, the new 3.0 version seems to be a winner. It will

work, albeit very sluggishly, on an XT system or an early slow 286 system, but it is designed especially for 386 systems. Note that many of your favourite TSRs may well conflict with Windows. As an aside, it would seem that Windows 3.0 has given DOS new life. A multitude of applications are being developed or converted for Windows 3.0. It seems likely to be able to carry a single-user 286, 386 or 486 system through the next four or five years without any necessity far switching to OS/2, which provides a much better operating system.

- Changing your operating system to OS/2, UNIX, PC-MOS, Concurrent DOS or some such. While there are any number of fanatical supporters of each of these alternate operating systems, the average PC user looks forward to a complete change of his operating system with as much enthusiasm as root canal work at his dentist.

You will note that none of the above will work well, if at all, with an 8088-based system. The 1MB addressable limit and the slow speed of the 8088 processor prevent the effective use of any of these approaches, except for expanded memory boards.

### GENERALIZED DOS MEMORY MAP

When your PC is finished booting up, its system memory map will look like this:

#### DOS-Addressable High Memory Limit (1MB)

- System ROM
- ROM BASIC (1ME only)
- Page-frame windows to access expanded memory, if required. (These may be located in other unused areas of high memory.)
- ROM extensions for your video card, your hard drive controller, etc.
- Text display memory
- Graphics display memory

#### User Memory Limit (640KB)

- Transient portion of COMMAND.COM
- Normal application programs
- Resident portions of memory-resident programs, i.e., TSR programs, in the order specified in the AUTOEXEC.BAT file; see (\*).
- MASTER ENVIRONMENT BLOCK
- Resident portion of COMMAND.COM
- External device drivers (e.g. ANSLSYS) loaded by your CONFIG.SYS file through MSDOS.SYS or through IBMDOS.COM; see (\*).

Supplementary DOS data area, including FILES, File Control Blocks and SUFFERS above the default values,,. Also, any STACKS (if specified in CONFIG.SYS) and the Logical Drive Tables. This area also includes run-time DOS modifications made with such programs as SUBST.EXE, JOIN.EXE, etc.; see (\*).

- DOS basic code and data, including the default numbers of FILES and BUFFERS provided by MSDOS.SYS or IBMDOS.COM
- Internal DOS drivers (CON, AUX, PRN, COM, etc.) provided by IO.SYS or IBMBIO.COM
- Basic Input/output Services provided by the ROM BIOS and IO.SYS or IBMEIO.COM
- System interrupt vectors as provided by IO.SYS or IBMBIO.COM and the system ROM BIOS

#### Bottom of DOS Low Memory (OK)

- (\* ) Some or all of the functions or programs marked with an asterisk may be loaded into high memory (between 640KB and 1024KB) or expanded memory using Quarterdeck's ARAM or QEMM (or the equivalent from other software firms), thus freeing more low memory.

### REMARKS IN BATCH FILES

*By Robert Parkinson*

### DISCUSSION

Most books and magazine articles on the subject of DOS batch files make the very valid point that you should put comments into the batch files that you write, using the "REM" command. This tip is suggested in order to make your files more readable and understandable, especially to other users.

This is all well and good, but you should understand that this will slow down the execution of your batch files in direct proportion to the number of "REM" lines used. When we use a number of these remark lines, the increase in execution time becomes very significant. I have done some extensive testing and would like to give you the results. All tests are the average of ten passes and were carried out on a 10MHz.

(Continued on Page 3)

XT clone with a V20 processor using MS-DOS 3.30.

In a very short and simple batch file, (see TRIAL.BAT below), adding ten "REM" lines, each with about 53 characters or spaces, added 0.65 of a second to execution time. If we eliminate all spaces and characters after the "REM" itself, the increase is cut to 0.53 of a second. The conclusion is that the amount of text in a "REM" line is not a very significant factor; the greatest determining factor being the time that DOS takes to realize that it is a remark fine and go on to read the next line. This increase in execution time will be exaggerated if you insert "REM" lines inside a batch file loop. In this case, the remark fines will be reread each time the loop is executed.

One way to reduce this increase in execution time is to use a label designator (i.e., a colon) instead of the "REM" command. A variant of this is to use a double colon. This was tested using the same TRIAL.BAT, with ten "::" lines of some 53 characters and spaces each. This method resulted in a modest increase in execution time, over the non-commented file, of only 0.05 of a second. If we eliminate all characters and spaces after the "::", the increase was cut to zero. The conclusion reached here is that it is far better to use the "::" for comment lines in simple batch files.

You might imagine that this conclusion would not hold true in a batch file with a number of batch "GOTO" statements, as whenever DOS encounters a "GOTO", it starts at the top of your batch file to look for label lines starting with a colon. It reads each and every one until it finds the label it is looking for, and your "::" comment fines look like labels to DOS. It won't take any action on them, but it reads them. However, this supposition is NOT correct; there is virtually no real difference.

This was tested using a looping batch file, see LOORBAT below, using ten "REM" lines, each with some 53 characters and spaces, before the actual loop. This test, which was executed 20 times, produced an increased execution time of 2.16 seconds. The same test with ten "::" fines, gave an increased execution time of 157 seconds. Again, the use of label lines for comments produced a saving of execution time (0.59 of a second), almost identical to the saving in a non-looping batch file. This same pattern held true in all

cases with one to six "GOTO" statements with corresponding label lines.

There is one very excellent way to insert general comments into your batch files that will NOT slow execution time. However, it will not allow you to comment specific lines in the file. To use this technique, you must have on your PATH a zero-byte file called "QUIT.BAT". Make sure that it is zero bytes, as it then doesn't take up any disk space, merely a directory entry, and it must have a .BAT extension, so that DOS can find it on the PATH.

Having now created this file, you simply end your batch files at any point you wish with a "QUIT" command. DOS transfers control to "QUIT.BAT", which takes absolutely no action. It just drops you out to the DOS prompt. If you use this technique, (see EXAMPLE.BAT below) you can then insert all the "REM" fines that your heart desires AFTER the "QUIT". In fact, the fines don't even require a prefix of any kind, just type them in. DOS never sees them, so there are no error messages and there is no increase in execution time.

This technique can be further enhanced, as in EXAMPLE2.BAT, by the use of a label to push syntax help information, for example, down below the level of normal DOS scrutiny.

#### EXAMPLE BATCH FILES

Three example batch files are given below. Their usage was discussed above. Several points need some clarification:

#### TIMER SEQUENCES

If you have the Norton Utilities available, you may wish to use his Time Mark (TM.EXE) utility, in which case see the manual for instructions. But TM.EXE, although it does your elapsed time calculations for you, rounds off to the nearest second, which may not produce the exact times that you need. I have therefore incorporated a generic timer routine. It starts by appending an echoed "START" to a LOG.FIL Then it uses the DOS "TIME" command, piping it through the DOS "FIND" command, in the "I FIND "C" " sequence. This returns only the line "Current time is..(etc)." and appends that to LOGTTL. To avoid the "Enter new time:" fine generated by the "TIME" command, a carriage return is redirected to that command with the "< CR.DAT" sequence. CR.DAT is created by making a file of that name that contains only a carriage return.

The same fines, with an END in place of

START, are used to log the ending time.

If you wish to delineate the results in your LOG.FIL, you can enter a few more lines in front of the "ECHO \*\*\* START ...." line. For example, you could put in a line

```
"ECHO ----- >> LOG.FIL"
a line with
"ECHO This is a test of .....(etc).....>>
LOG.FIL"
and another line draw line.
```

If you only wish the results on screen, delete all the ">> LOG.FIL" sequences. The advantage of the log is that you can run many iterations of any batch file variant and then edit the log to do your mathematics and enter any comments or explanatory text that you wish.

If you want to further cut down the sequence sent to LOG.FIL, you can use the PC Magazine utility "MID\$.COM". In this case, change the timer fine to read "TIME < CR.DAT | FIND "C" | MID\$ 20 >> LOG.FIL" and only the actual minutes and seconds, to two decimal places, will be sent to the log.

#### LOCATION OF COMMENT LINES

I have marked (e.g. "<Location #1>") the spots in the example files where you might wish to insert a number of comment fines for testing. You choose the number to insert, I used 10. Here are the three types of fines that I used, testing 10 iterations of the batch file for each type in each location. Note that in the third and fourth test fines, nothing at all follows the "REM" or the "::".

- REM This is a comment statement, commencing with an "REM".
- :: This is a comment, commencing with a double colon label.
- REM
- ::

I will continue next month with more example batch files.

\*\*\*

#### OPCUG FLEA MARKET

The OPCUG's annual flea market will be held on Sunday, November 18, 1990 at the R.A. Centre, 2451 Riverside. The table set-up time will be at 12:30 p.m. and the doors open at 1:00 p.m. Non-members are welcome. For further information, call Harry Gross at 733-7989.

## BACK TO BASICS

By Harry Gross

### DATA TYPES

The computer program decides on how to handle and store data by knowing what type it is (as distinguished from its value). In BASIC, we have two broad categories, string and numerical.

First consider strings. These represent a very powerful tool for storing and handling non-numeric data. It was this capability that led me, several years ago, to use BASIC for many applications instead of FORTRAN. I became irritated when, on entering a date, say August 12, in a FORTRAN program, I had to count on my fingers to find out that August was the eighth month of the year. How much easier to enter "Aug,12", instead of "8,12" or, in error, "12,8".

A string variable is indicated by the \$ after the variable name, such as Month\$, Day\$, Name. The variable Month is a numerical variable and unless the program establishes some connection, Month and Month\$ have nothing to do with each other. A string constant is denoted by the use of quotation marks. So we can have

```
Month$="January"
```

and later on in the program,

```
Month="March"
```

The variable Month\$, has changed its value from "January" to "March". We can also have string arrays, as follows;

```
DIM Month$(12)
```

```
Month$(1) ="January" : Month$(7) ="July"  
Month$(2) ="February": Month$(8) --"August"  
Month$(3) ="March" : Month$(9) ='September'  
Month$(4) ="April" : Month$(10)='October'  
Month$(5) ="May" : Month$(11)="November"  
Month$(6) ="June" : Month$(12)="December"
```

A fuller discussion of arrays will be done later on, but for now we will use only simple examples. In interpreted BASIC, the length of a string is limited to 255 characters, while the compiler allows 32,767 bytes.

Several operations are available for strings. They can be concatenated, or joined together using the + sign.

```
A$="Happy": B$="Birthday": C$="New Year"  
D$="Paul": E$="Oliver"
```

```
IF Month$="February" THEN
```

```
Q$=A$+" "+B$+"", "+E$
```

```
ELSEIF Month="September" THEN Q$=A$+" "+C$+"", "+D$  
END IF
```

```
PRINT Q$
```

Also, as shown above, strings may be compared for equality. If two strings are of the same length and match character for character, then they are identical. For this purpose, upper and lower case are different, so "A" does not equal "a", and talk of equality naturally leads to inequality. Here details are not quite as obvious. Look at a copy of the ASCII table and consider two single character strings, x\$="A" and y\$="B".

In the above-mentioned table, "A" has an ASCII value of 65, and "B", 64. A string is less than another if its ASCII value is less. So for the three possible comparisons:

1) x\$ = y\$

2) x\$ < y\$

3) x\$ > y\$

only the second one is valid. If x\$="a" and y\$="A", then relation three is valid, because 97, the ASCII value of "a" is greater than 65, the ASCII value of "A". This becomes important when sorting lists, as the results may be different from what is expected when capital letters, blanks, and pronunciation marks enter the picture.

For longer strings, a comparison is made on a character-by-character basis until a difference is found, and then a decision is made. If a shorter string is identical to the start of a longer one, then the shorter is the lesser.

```
Evil$(1)="Patronage"
```

```
Evil$(2)="Burgling"
```

"Patronage" is the greater of two Evil\$.

We must be aware of a string and its ASCII value and how to go from one to the other. For this we have a pair of functions.

```
g$=CHR$(g): g=ASC(g$)
```

So, if we let g=52, then PRINT CHR\$(g) will produce the value "4". If x\$="4", then PRINT ASC(g\$) will give us 52. We can now write a small program to list all the ASCII values.

```
FOR n= 32 TO 255
```

```
  LPRINT n "-" CHR$(n) " ";
```

```
NEXT n
```

This is quite useful when introducing yourself to a new printer, especially the laser family. The ASCII values from 128 to 255 do not always print the same from character set to character set, and the above is a useful way of seeing what is around. I skipped the values 0 to 31 as they are control characters and may produce weird effects, such as page and line feeds, etc. Especially notorious is 27, the infamous Escape character. Here, results may really be wild, but try it for fun.

Another useful pair of functions are VAL(x\$) and STR\$(x).

```
x$="1234.67": x=VAL(x$): PRINT x$,x
```

will give us

```
1234.67            1234.67
```

They look the same, but are not.

IF x\$="a12.3x", VAL(x\$) will return 0, while x\$="12.3a23" will return 12.3. The VAL function operating on a string will start at the beginning and only use that part of it to the first non-numeric character. Commas, such as in 1,230.00 are treated as non-numeric characters, while the first decimal point is numeric.

There is a full set of string operations available, but we will leave others to later.

Three other string functions are HEX\$(n), OCT\$(n), and BIN\$(n). They will return a string containing the hexadecimal, octal, or binary version of an integer number.

```
a=200: WRITE a, HEX$(a), OCT$(a), BIN$(a) will give  
200,"C8", "310", "11001000"
```

For the numerical constants, there are four types.

(1) INTEGER

covers the range -32,768 to 32,767. As implied by the name, values change by unity. Two bytes of storage are needed for each value. A variable is indicated by the % sign, A%=32000.

(2) LONG INTEGER

covers the range -2,147,483,648 to 2,147,483,647. Again the values change by unity. Four bytes of storage are needed for each value. Long integer is not available in the interpreter. A variable is indicated by the & sign, A&=32000043.

(Continued on Page 5)

(Continued from Page 4)

**(3) SINGLE PRECISION**

extends from 10 to the power of -38 to 10 to the power of +38. However, in this range, the number only carries six digits of accuracy. So, applications dealing with numbers like 23456.37 might lead to significant errors when they are subjected to mathematical operations. Four bytes of storage are needed for each value. The variable is indicated with the exclamation mark, A!=3278.91.

**(4) DOUBLE PRECISION**

extends from 10 to the power of -308 to 10 to the power of +308, with 16 digits of accuracy. Eight bytes of storage are needed for each value. The variable is indicated with the number sign, A#=327891.98424. The speed of processing decreases from the first to the last of the above list, so select what to use according to the job on hand.

By default, a program will treat all untyped values as single precision, so the variable A is the same as A!. Be warned that A!, A%, A&, and A# are all different variables with no relationship, other than those made explicit in a program. We can use these statements:

```
DEFINT a-b
DEFLNG a-b
DEFSNG a-b
DEFDBL a-b
DEFSTR a-b
```

to preset our variable types without adding a type symbol to the variable. a-b represents some range of letters for the typing, and any variable starting with a letter in that range will have that type unless there is an explicit assignment made.

My own practice for most programs (but not all) is to use DEFINT A-Z at the start of the program to make integer the default type and to use others with the appropriate symbol as needed. I find this works out quite well as most of variables are used for counters in loops or indices in arrays. This saves memory and speeds up execution.

There is a difference in operation here between the interpreter and the compiler. For the interpreter, the program flow of execution must go through the statement for typing to be active. If there is a jump around these statements, they are not executed. For the compiler, they must be physically located in the program flow before execution, no matter where the logic goes.

Be careful about the location of these statements in the program, especially if you change type.

Try writing a short program with these concepts, and experiment with statements like these:

```
A!=345.235: B!=345.98
A=A! : B=B!
PRINT A,A!,B,B!
C%=43000
PRINT C%
```

The last will return an error message, as the value of C% is out of range. In the next article, I will return to strings and more of the operations that may be done with them.

\*\*\*

*By Jean Fortier*

This article is intended to give proper French translations of the general parts of a computer.

**THE COMPUTER ITSELF**

computer	- ordinateur
component	- élément, composant
power supply	- bloc d'alimentation
motherboard	- carte maîtresse
microprocessor	- microprocesseur
chip	- puce
connector	- connecteur
hard disk	- disque rigide
hd controller	- contrôleur de disque rigide
floppy drive	- unité de disque souple
expansion slot	- fente d'expansion
light	- voyant lumineux
switch	- interrupteur

**AROUND THE COMPUTER**

peripherals	- périphériques
monitor	- écran
keyboard	- clavier
printer	- imprimante
modem	- modem
mouse	- souris
cable	- câble
power bar	- barre d'alimentation

This concludes an overview of the computer. In future articles, we will take a closer look at each component. If any member is experiencing particular translation problems, please feel free to leave me a message on the PUB.

\*\*\*

**BARTENDER WANTED!**

*By Mire Schupan*

Do you like talking with people, helping people out, and offering your advice? Do you have any desires to serve a small community of friendly, outgoing individuals? The PUB could be just the place for you!

Well, it's not exactly a bartender position, but being SYSOP for The PUB does give you all the same benefits. The Ottawa PC Users' Group is looking for someone to run the Group's bulletin board system.

If you are interested in working with people over a modem instead of a bar, manipulating files instead of mixing drinks, and managing accounts instead of bouncing boozers, then think about SYSOPing at The PUB. Even if you only want to help out, what bar would be complete without waiters and waitresses?

Leave a message to anyone on the Executive and start your career in electronic socializing today!

\*\*\*

## BEGINNERS' CORNER

Immediately prior to the monthly OPCUG meeting, there is a special meeting intended for beginners, starting at 7 p.m., usually in Room 1125 of the NRC Building on Sussex Drive. At each meeting, there are many opportunities to ask questions or to discuss problems. Also, there is usually a special topic for discussion each evening. If you would like more information, call Eric Clyde at 749-2387.

### "DON'T, OR HOW TO CARE FOR YOUR COMPUTER"

*By Eric Clyde*

This heading is the title of a book published a few years ago. While the contents are mostly familiar to experienced microcomputer users, the ideas are worth repeating for new users, who may want to consult the book at their public libraries.

#### GENERAL POINTS

Microcomputers are, in general, surprisingly reliable if certain precautions are taken.

- a) Non-portable computers should be kept on solid surfaces (e.g., not card tables), since mechanical shock and vibration can damage components and loosen boards, chips, etc., causing erratic operation.
- b) Proper ventilation is essential. Make sure that air intakes are kept clean and are not blocked. Overheating causes chips to fail early. As a general rule, if it is uncomfortably hot for you, it is too hot to be using the computer.
- c) The electrical outlet you plug your computer into should not be on the same circuit as any major appliances. For example, when the refrigerator turns on, an electrical surge can come down the line and cause problems. If you live in the country or suspect that you may have problems with the power lines, you would be advised to get a good surge protector. It's good insurance, wherever you live.
- d) Don't use your computer during a thunderstorm. Electrical surges can cause loss of data. Also disconnect your modem.
- e) Dust, smoke, and pollution can cause problems to many components.
- f) Keep coffee cups, etc., away from the computer, keyboard, disks, etc.

## FLOPPY DISKS

- a) Handle floppy disks with respect for the data on them. Don't touch the disk surface or fold the disk.
- b) Don't expose them to dust, young children, smoke, spilled coffee, etc.
- c) Label disks clearly, but don't use hard tipped pens to write on the label.
- d) Don't leave disks lying around. Store them in a closed container.
- e) Avoid exposure to stray magnetic fields. Valuable data can easily be accidentally erased.
- f) From time to time, check the alignment of your drive, either by using special software or by checking that the disks you produce are readable on someone else's machine.

## HARD DISKS

- a) Park the heads before switching off the machine.
- b) Avoid mechanical shock. Don't drop the computer, and don't move it without parking the heads first.
- c) For emergencies, make several bootable floppy disks. Then, if something horrible happens and you can't boot from your hard disk, you have a partial safety net.
- d) For safety, give your COMMAND.COM the read-only attribute. Then if you copy a bootable disk to your machine, you won't accidentally overwrite it.
- e) If you use software from other sources, be careful of your source and use virus protection software.

## SOFTWARE

- a) Make backup copies of new software before you use it. Put the original in a safe place and use the backup.

## DATA (letters, reports, spreadsheets, etc.)

- a) If you are making changes to your files, save your work frequently.
- b) If you have made more than minor changes, save the changed file before you print it out or do extensive data manipulation.
- c) Try not to shut off or reset the computer in the middle of a program. Some of your files may be damaged.
- d) At the end of each session, backup at least the changed files from your hard disk to floppy disks (or, on floppy disk based systems, make a backup copy of your floppy disk).

I hope I haven't alarmed you. If you treat your computer well, it should be reliable for a long time. Happy computing.

\* \* \*

## PUB TALK

### QUERY: MARK BLEVIS

New PUB Messages

How can I only read new mail? Is it the "-"command?

### REPLY: CHRIS TAYLOR

Each time you log off, TBBS records the highest message number read in each of the message areas. Make sure you don't just <R>ead <R>everse unless you want to make sure you backtrack all the way through all unread messages. Likewise, if you choose to read new messages, and you start to follow threads, make sure you read \*all\* new messages. By following a thread, you may jump around, for example, from message 13000 to a reply at 13010. If you then quit, TBBS will record the message number 13010 as the last message read and you will never see 13000-13009 by choosing <N>ew messages in the future, even though these messages were unread.

### QUERY: ANDRE LEFRANCOIS ALEX

What can anybody tell me about a Bell Canada System called ALEX?. It seems to be all the rage in Montreal.

### REPLY: JEAN FORTIER

ALEX is a terminal rented from Bell Canada, which can be used to shop electronically in the Sears catalogue. Users who have their own computer and modem can access this service with software provided by Ma Bell. Unfortunately, this service is only available right now in the Toronto and Montreal areas.

From: PAUL COOPER  
Computer Chronicles

Has anyone else noticed that COMPUTER CHRONICLES is back on CH 16, PBS, at a new time 5.30 PM on MONDAYS? [Hope you all have VCRs].

### QUERY: DOUG HEWKO Archiving Utility

Isn't there a shareware program out there that is supposed to Iharc (or was it zip, at any rate compress) files so that they run in compressed mode? Are these compressors good, bad? Can they ruin the original program? What about virii?

### REPLY: CHRIS TAYLOR

You are looking for LZEXE. It is on the board. Try the Utility/archivers area. Wore^ on most programs that don't use overlays. Works great. Several instances of programs

that won't run. like SCAN that does a CRC check on loading. Since they have been modified, SCAN detects that. They are actually a boon virus-wise. As long as you are sure they are fine to begin with, they should be virus-resistant, as they won't decompress properly if they get modified by a virus. I have saved a couple of hundred Ks in my utility directory with this gem of a program.

From: DOUG HEWKO  
Databases

Pick up the new issue (Oct.) of PC WOLRD. It has some excellent articles on databases (i.e. SQLs), tech support, and a listing of the best shareware programs. For \$3.95 in Canada, you get 338 pages, compared to the 310 pages in Sept.

From: LAIRD GORDON

New Magazine On The Block

There is a new magazine called PC Sources now available on the newsstands. It is a magazine size clone of Computer Shopper published by the same publisher, Coastal Associates (now owned by Ziff-Davis). Two of the contributing editors are Winn L. Roach and Dave Rowell. Roach has an excellent Upgrade Clinic article on memory improvement. The other nice thing is that the Premier Issue has a special price of \$0.95. That's right, and it would appear that the monthly price will be \$1.95, no indication of a higher price in Canada.

QUERY: GORD PHILLIPS

Ottawa Amateur Radio Club

Where and at what time are the Ham Club meetings? Are non-members (and non-amateurs) welcome?

REPLY: BRICE WIGHTMAN

The Ottawa Amateur Radio Club meets each first Wednesday of the month in the auditorium of the NRC, 100 Sussex Drive at 8 p.m. The Ottawa Valley Mobile Radio Club meets each third Thursday of the month at (temporarily) a multi-purpose centre on Heron Road, south side, just east of Alta Vista. It is the building under construction across from St. Pat's High School. Yes, visitors are by all means welcome.

QUERY: TOM REDDICK

Unzipping Files

Whenever I try to unzip a file from the PUB I get a warning message. Can anyone tell me what I am doing wrong?

REPLY: CHRIS TAYLOR

You probably have an old version of PKZIP. Try downloading the current version. It is in the Utility/Archivers file area.

QUERY: JOHN WHELAN

Message CCs

I'm trying to talk to two different users at once in the private area, and I'd love to use the cc function, but I can't seem to make it work. Could you remind me exactly what to do?

REPLY: CHRIS TAYLOR

The first line in the message should take the form cc: Chris taylor,John Whelan,Joe User. If any of the names are not currently in the userlog file, (I believe) the whole procedure aborts.

QUERY: DANA WEBBER

Windows 3.0

After all the press over Windows 3, this may seem weird, but what can Windows actually do?? I installed it two days ago, and so far XTGold does better. Does anybody know about context switching and how Windows compares to Deskview/386? Windows seems much more cumbersome and less powerful.

REPLY: CHRIS TAYLOR

There are few people who run Windows in order to tun Windows. The excitement behind it is the level of activity of applications being written for the Windows environment.

The great thing (from a developer's point of view) about Windows is that all screen and printing services are handled by Windows as well as memory management (probably disk I/O as well), so they can concentrate on the functions of the application itself.

From the user's point of view, Windows provides a more standardized user interface. You don't have to remember Shift-F7 is a print menu in one program, while Alt-P may do it in another, and Ctrl-Alt-O may do it in another. In Windows applications, it's on the File menu ...always.

Windows 3 has a much prettier face than previous versions. It also has better memory management facilities built in and is the one program that can run on any of the 8088/80x86 processors. All you have to do is look at Corel DRAW!, Charisma, Designer, Powerpoint, Word For Windows, Excel, etc., to see sane really exciting products.

From: DANA WEBBER

Easy for the user? It took two hours to find the context switch, and I still can't split the screen like I can with Deskview.

REPLY: CHRIS TAYLOR

Splitting the screen for a Windows application is easy. Just grab the edge of a window and drag. DOS applications can only run in a window if you are running in 386 enhanced mode. Otherwise they must run full-screen. I admit there are lots of rough spots. SETUP performs less than admirably in many situations when re-configuring a system. Icons do not stay in place if you have "auto-arrange" on and your window is not large enough for all to be visible (i.e. when you re-start Windows). I was not very impressed with previous versions of Windows, but Windows 3 is nice. Version 3.1, whenever that appears, will hopefully polish the rough edges.

FROM: LAIRD GORDON

Warning: Windows 3.0

A column by Patrick Honan in the October, 1990 issue of PC Sources warns about conflicts between Windows 3 and other programs. He states that the most dangerous situation which can damage or destroy files is when you run Windows' SmartDrive caching with a commercial disk partitioning program. There are two ways that this can happen; first, using a disk partitioning program & SmartDrive on a hard disk with more than 1,024 cylinders and second, using a disk partitioning program, SmartDrive, DOS 3.3 or higher, and using a hard disk that is not supported by your ROM BIOS. Honan also says that there are problems running Windows on a network. He specifically mentions NetWare and Vines. As Honan says "Let the buyer beware". I received Windows 3 from Microsoft Canada and the files are all dated May 90! I thought that by waiting four months to upgrade I would be safe from some of the early bugs.

REPLY: CHRIS TAYLOR

Although I don't know about Vines, I do know that the fix to Netware 386 has been available for about 3 months now. You can download all files you need to upgrade Netware 386 from Compuserve or get the upgrade from Novell. Also, I understand that QEMM has a new version that will work fine with Windows 3.0. I can only assume that 386MAX is in the same position by now.

\*\*\*

**OTTAWA PC USERS' GROUP**

**EXECUTIVE**

Chairman	David Terroux	238-4895
Past Chairman	Stu Moxley	832-1278 or 832-4378
Treasurer	Tony Frith	671-0401
Secretary	Claude Jarry	521-3366
Newsletter Editor	Bonnie Carter	236-1015
Software Librarian	Chris Taylor	723-1329
BBS System Operator	Mike Schupan	820-0293
Convener	Douglas Poulter	745-8768
Membership Chairman	Carl-Henri Gomez	731-1462
Hardware/Software Broker	Terry Mahoney	225-2630 226-2615 FAX

**ASSISTANTS**

Assistant Newsletter Editor	Jean Fortier	236-1015
Newsletter Assistants	Marc Riou	733-2092
	Chris Taylor	723-1329
	Ann Falkner	722-3226
	David Smith	837-9291
Software Assistant	John Ings	235-8132
Membership Assistant	Marc Riou	733-2092
BEGINNERS' SESSIONS	Eric Clyde	749-2387

**DISK OF THE MONTH**

One-year subscription (10 Disks) 5.25" \$25.00  
One-year subscription (10 Disks) 3.5" \$35.00  
Individual 3.5" disk surcharge \$2.00

**THE OTTAWA PC USERS' GROUP MEMBERSHIP APPLICATION • Please Print**

Last Name: \_\_\_\_\_ First Name: \_\_\_\_\_  
Mailing Address: \_\_\_\_\_  
Postal Code: \_\_\_\_\_ Telephone - Home: \_\_\_\_\_ Office: \_\_\_\_\_  
Profession: \_\_\_\_\_ Date of Birth: \_\_\_\_\_ Sex: M \_\_\_\_\_ F \_\_\_\_\_  
Disk of the Month: YES \_\_\_\_\_ NO \_\_\_\_\_ Size: 5 1/4" \_\_\_\_\_ 3 1/2" \_\_\_\_\_ Amount Enclosed \$ \_\_\_\_\_  
Are you: A new member? \_\_\_\_\_ Renewing your membership? \_\_\_\_\_ Sponsor's name? \_\_\_\_\_  
What in particular interests you in the Group? \_\_\_\_\_  
Can you help in Group activities? Check off the activities that apply: Programming language Instruction \_\_\_\_\_  
Newsletter Input \_\_\_\_\_ Memberships \_\_\_\_\_ Software Library \_\_\_\_\_ Promotion/Publicity \_\_\_\_\_  
Hardware Techniques \_\_\_\_\_ Meeting Locations \_\_\_\_\_ Agendas & Speakers \_\_\_\_\_ Advertising \_\_\_\_\_  
Bulletin Board \_\_\_\_\_ Other \_\_\_\_\_  
What hardware/software do you own and/or use? \_\_\_\_\_