

NEWS

Vol 2

North Texas IBM Personal Computer Users Group

No. 3

Special Interest Programs

Programmers

"Hard disk experience" was the theme of the Feb. 19 meeting of the programmers' group. Several members shared some of the joys and problems of attaching and using one of the available hard disk units with the PC.

Comments and watchouts included:

- Incremental backup capability is desirable
- Formatting flexibility is often important
- Some copy-protection schemes prevent software from being transferred to a hard disk

Mike Wiles will host the March meeting while group chairman Neil Bennett is attending the West Coast Computer Fair. The topic at the March meeting will be:

PC-COMPATIBLE COMPUTERS.

This topic generally includes basic discussions on the definition of compatibility, the level at which it is required for various users, and schemes for providing apparent compatibility. Members will be invited to share their experience on specific PC-similar computers. Additional comments on PC-DOS 2.0 can be expected at the discussion portion of the March meeting.

It has been suggested that an individual be designated as a single-point contact for member questions raised at meetings. Questioners would be requested to provide the contact with information so that other members could locate him/her later for providing answers and/or suggestions. The contact could also provide questions of general interest as input for the PC News QUESTIONS & ANSWERS column. The Programmers' Group will discuss appointing a member to be the Q & A contact.

Dick Gall

Address Label Sale

As a result of the vote at the February Group meeting, our mailing list will be offered for sale to advertisers associated with the IBM Personal Computer, it's look-alikes, and accessories. Proceeds from the sale will go into the Group's general fund.

A sheet will be circulated at the next meeting for you to mark whether or not you want your name included on the list. Check the appropriate box opposite your name to indicate your choice. If you are a new member and your name does not appear on the list, please print your name, address and preference on the sheet attached to the list. Be sure to make your selection before leaving the general session.

The mailing list will be sold in the form of a set of labels (size 15/16" by 3 1/2") to advertisers who represent interests similar to those of our Group. The price for each set of labels will be \$30 per hundred names. Professional members of the North Texas IBM PC Users Group will receive a discount of \$4 per hundred names for each set ordered.

If less than fifty percent of the members allow their names on the list we offer for sale, this project will be shelved for an indefinite period.

Agenda

During the main session of the next meeting the local rep for 3-COM will demonstrate and discuss EtherLink and Ethernet.

Chris Morgan

BASIC SIG

We'll be discussing keyboard input routines at the March meeting.

Chuck Loftin

Next Meeting March 19, 1983

Auditorium, Cox School of Business, Fincher Building

(1/4 Mile South of the Heroy Building)

SMU Campus

9:30 TO 12:00

*Leader's 9:00
meeting postponed*



North Texas PC NEWS

Published monthly by members of North Texas IBM Personal Computer Users Group for their use. Members each receive a free subscription; for others, price of the NEWS is \$1.00 per copy. Advertising is accepted; contact Editor for rates. Members are requested to notify the editorial staff in writing of address changes. Send all editorial correspondence to the Editor, PC NEWS, 2025 Rockcreek Dr. Arlington, TX 76010.

Editor John Pribyl (817)275-4109
 Programming Editor: Tom Prickett (214)387-3667

Copyright (c) by North Texas PC NEWS, 1982.
 All rights reserved. Use or reproduction
 in any manner is prohibited without written
 permission.

Deadlines:

All material for publication in PC NEWS (articles and ads) must be received by the NEWS staff no later than the first Saturday of the month in which publication is desired.

North Texas IBM Personal Computer Users Group

A non-profit, independent group, not associated with IBM Corporation. The Group meets on the third Saturday of the month at 10:00 AM in the Auditorium, Cox School of Business, Fincher Building, SMU Campus. (Check meeting notice on page 1 for any late changes.)

Officers:

President	Alan Elliott	(214)941-8475
Program Chairman	Chris Morgan	(214)446-0484
Treasurer	Bill Hood	(214)350-9784

Special Interest Groups:

Beginners	Mike Durbin	(214)271-8779
Business Applications	Alan High	(214)385-0553
Disk of the Month	Will Janoschka	(214)231-6449
Programmers	Neil Bennett	(214)238-7650
BASIC Applications	Charles Loftin	(214)492-3594

Special Group

Bylaws Committee	Fred Williams	(214)245-4319
------------------	---------------	---------------

Dues: Professional Membership - \$36.00 year
 Regular Membership - \$24.00 year
 Student membership - \$12.00 year

Payable in January, dues are prorated for balance of year when applicant joins after January. Dues should be mailed to W. A. Hood, 10437 Goodyear Dr, Dallas TX 75229.

B Y L A W S

We've had a quick look at the first draft of a set of bylaws for the Group. Fred Williams has done a commendable job. They are presently being reviewed by the officers and should be available for a vote by the members within the next 60 days.



DISK OF THE MONTH

by Will Janoschka

MARCH HIGHLIGHTS

PC-FILE. A general purpose "Data Base Manager" program from FREEMWARE, written with EASE OF USE as the overriding goal. Useful for mailing lists, price lists, inventories, printing labels, and as a front end data base processor for other programs. Also sorts data in almost any sequence. A real tool for learning how to use data base software. Supplied in compiled basic, with extensive documentation. 64K PC required, supports 40 and 80 character displays.

more >

L E T T E R S

New IBM-PC User Club

Dear Editor:

We in the Athens, Ohio area are in the process of organizing an IBM-PC user club. We have had two initial meetings, and I have been elected president.

We have begun to map out an education program for beginners who are interested in our club and who have access to an IBM-PC. We have planned a series of seminars on BASIC, presentations on word processing, spread sheets and data base management software.

We feel it would be helpful to us as a fledgling organization to establish contacts with other clubs and begin a dialogue which may lead to sharing of ideas, newsletters and software.

Currently, we have assembled a software disk with about a dozen programs to be made available to club members. These include games, music, application software and utilities. We would be interested in sharing what we have (limited as it is at this early point) and acquiring similar assistance from other clubs.

Yours truly,

Charles F. Harrington
 Associate Provost
 Ohio University



DISK OF THE MONTH

continued

GOMOKU. A full-screen board game. Said to be easier than GO and harder than PENTE. You try to line up 5 men and the computer interferes in tic-tac-toe fashion on a grid containing 30 X 7 spaces. Difficulty levels 1-5. In interpreted basic. Contributed by Pete Lewis.

WSCONV. A compiled version of a previously-issued program for making Wordstar text files printable by standard ASCII print routines. Improves execution time over the interpreted-basic version.

XREF. A BASIC Cross Referencer. A handy utility which helps debug Basic programs. Prints a list of variable names and most Basic keywords and the line numbers in which each is used in a program. From SOFTALK, Feb. 83.

PREVIEW OF COMING ATTRACTIONS

RATBAS. A pre-processor for converting programs written in a structured language called RATIONAL BASIC into basic executable with the PC's standard basic interpreter. For Fortran fans, the concept is similar to RATFOR. A VOLUNTEER is needed to write an example in RATBAS, such as an implementation of the game "Life".

HELP!!! Call Will Janoschka at 231-6449 to submit programs and information for future diskettes.

DISK DETAILS

Price: \$5.00. Available at the general membership meeting. Media: DSDD 5" diskettes formatted single-sided. Public domain software only, standard full disclaimers. The February diskette sellout including PAC-GAL and MAZE is also available at the March meeting.

USING THE DISK OF THE MONTH

The Disk of the Month is set up to be used with an IBM PC using the standard PC-DOS Disk Operating System provided by IBM. The basic-language programs are designed for use with the BASIC interpreter programs provided with PC-DOS. Users should be aware of the following items when using the DOM issues:

DOS IS NOT INCLUDED. The IBM Program License Agreement prohibits the transfer of copies of licensed PC software. Therefore the three programs (two "hidden" and the COMMAND.COM program) that make up DOS cannot be

included in the distribution copies. A PC must be operating under DOS before use of the DOM issues is attempted. Bring up (turn on) the PC with a DOS diskette installed, and then the DOM diskette can be accessed. Also, since DOM material starts at the very beginning of the disks, space is not available for adding DOS to DOM issues. The "Using DOS" portion of the "Operation" portion of the PC GUIDE TO OPERATIONS manual and the DOS manual should be consulted for details.

WRITE PROTECTION. Write protect tabs are not installed on the DOM issues when distributed. Protection from accidental erasure/modification of DOM material by installing write-protect tabs is recommended for beginning DOS users.

"README" file. The contents of each issue are listed and a brief description of each program or item is available in the ASCII text file called README. To view this information, use the DOS TYPE command ("TYPE README"). To print it, press the Ctrl-PrtSc key sequence before TYPE README, or use COPY README PRN. DOS's line editor program EDLIN or most word processor programs can also be used to read README.

READING TEXT FILES. Written information and program documentation is generally provided in ASCII text files which can be accessed the same as the README file. Normally these files can be identified by their filename extension ".DOC" (example PC-TALK.DOC).

EXECUTING BASIC PROGRAMS. Programs having filename extensions of ".BAS" may be executed using the BASIC or BASICA interpreter programs supplied with DOS. Starting from the DOS "prompt" (A), for example, enter the appropriate basic interpreter name and then that of the basic program to be executed. For example, BASICA PC-TALK. Or start basic by itself, and then use the basic LOAD and/or RUN commands with the name of the program to be executed.

EXECUTING .EXE AND .COM PROGRAMS. Programs with a filename extension of ".EXE" or ".COM" are provided in executable object module format and may be initiated from the DOS prompt simply by entering the program name, with or without filename extension. For example, MAZE.

FURTHER INFORMATION. Members who submit programs are identified either in the README file or in the PC-NEWS announcement of disk of the month contents. Contact the submitting member directly with questions or requests for further information.

How 2 Write Programs Fast



by Neil Bennett

If you have a micro and you enjoy writing programs for the fun of it, you may not be interested in this article. I write programs for the fun of it; but my income is geared to the amount of code that I can write. Since I spend all my time writing code, I am interested in how to do it more quickly.

Not all things are improved by being able to do them faster. Playing the piano is an obvious example. I am sure there are others.

Here then, for your consideration, is a set of rules I use to turn out code. You may well find the advice to be unpalatable, but I think if you follow it you will increase your output.

The first step in writing a program is to turn OFF your computer. I heard of a sign posted in the computer room of a very large main-frame in the 60's. It said "THINK, but not in here!". [I find it completely irresistible to tell you about the sign in the IBM men's room. Above the place where you wash your hands was a big sign that said "THINK". Someone had drawn an arrow down to the right hand side of the place where you wash your hands and beside the arrow had scrawled "THOAP". I understand that person no longer works for the company]

In those days batch turn-around was of the order of many hours and if you were unlucky, it could be several days. In order to get anything done you learned very rapidly that the fastest way was to

...desk-check your programs very thoroughly.

desk-check your program very thoroughly. Unless you were about 95% confident that the test was going to be productive, you were foolish to send it in. Because computer time was scarce there was generally quite a few people all using the same batch machine. Your output was fairly public and if the other people saw that you were regularly sending in jobs which bombed in the compiler because of a missing comma or

similar, you got rubbished. If you were really wasting computer time and it effected other people's usage you got more than rubbished. You may even have found the card reader chewed up your deck of cards more regularly than anybody else's.

Today you have unlimited access to a very fast computer. And it is private. There is nobody around to see if you take 17 goes to get your job working. Unfortunately, there is a great temptation to put your program through the compiler as soon as you finish typing it in. I maintain that the same rule from the 60's still applies. The fastest overall way is to get a listing of your program and take it over to your other desk and desk-check it. When you have checked it through, then go back to

...there are times when the machine can find things for you faster than you can..

the machine and use the compiler. The compiler has no idea what your program is trying to do, only you have. There is a nasty phenomena known as error cascading; if the compiler finds a syntax error it may continue with the rest of your program and you end up with about a dozen errors. Only one of them is relevant, and the rest are misleading.

But there are times when the machine can find things for you much faster than you can by desk-checking for them. This may seem a cop-out, but you can only judge by experience when those times may be. A rule of thumb is to remember that the machine is fantastic at high speed linear operations. Anything creative, pattern recognition or the like is best done by you. There is a concept of "yin" and "yang" in the chinese philosophy of "Tao" (pronounced "Dow"). The standard text is "Tao Te Ching" by Lao Tzu. (You may find both the title and author's name spelled differently depending on the translator. My copy is published by Penguin Classics.) Yang is easy to describe. It is the sun, the day, convex, linear, able to be described, the south side of the mountain which gets the sun, etc. Yin is the opposite. Yin is the moon, the night, concave, non-linear, intuitive, unable to be described, the north side of the mountain where the sun never sees. By it's nature, language is yang. You can describe yang by language, but you can't describe yin; you can only experience it. I hope it's clear that the machine is best at things yang, and you are best at things yin.

See PROGRAM FAST on page 6

How 2 Write Fast Programs



by Neil Bennett

The 80/20 rule says that 20% of the fishermen catch 80% of the fish. The 80/20 rule of computing says that control spends 80% of the time in 20% of your code.

Heisenberg's rule of computing says you can swap space for speed, and speed for space. A program can be speeded up at the expense of space, or can be made more compact at the expense of speed. An example is a software multiply routine. This can be written as 16 adds in a loop or as 16 adds one after the other. The first way is more compact but the loop control instructions take time (maybe as much time as the adds); the second is faster, but it takes up more space.

Note that this rule applies only to efficient programs; you may be able to make inefficient programs both more compact and faster. A trivial example is taking a redundant instruction out of a loop.

These two rules taken together can tell us a lot about how to write faster programs. The 20% of the code which is used over and over is called a "hot spot" (nothing like a "honkey tonk"). Some of the program just has to be there to handle initialization

**...you can swap space
for speed, and speed
for space.**

or special cases, but percentage-wise, it's almost never used. If you could make the 20%, (the "hot spots"), twice as fast at the expense of taking up twice the space, and the 80% half the space at the expense of halving the speed, then you would end up with a program which takes 80% of the original space and executes in 80% of the time. (On the other hand, if you choose your spots wrongly, you could make the 20% half the size and half the speed, and the 80% twice the size and twice the speed. That would give you a program which takes 170% of the original space and takes 170% of the time to execute.)

The problem is how to recognise these "hot spots". Unfortunately they don't have a big neon sign

outside. Sometimes you can tell easily. If you have a loop which is nested 4 deep then that innermost code probably gets done lots of times. Other times it is more subtle. If you have a real time system with devices causing interrupts at different priorities, you are usually grateful that it works, without wondering how to tune it. In this sort of situation, you could build instrumentation into the program to count how many times various parts of the program are executed. If you do this, you have to be careful that your choice of places to insert your probe does not influence your results. Back to the loop four deep. If you are calculating an expression which does not depend on your loop variables, such as $\exp(43.6)$, then you could calculate the expression just once at the start of the program and store it's value in a variable. It's much faster to fetch the value of the variable many times than it is to calculate $\exp(43.6)$ many times.

At this point, please consider the very different nature of speed and space. If a program

**...build instrumentation
into the program..**

takes 5 minutes to execute and you halve it's speed it takes 10 minutes; if you halve it's speed again it takes 20 minutes. The same thing does not necessarily apply to space. If you double the size of your program in a single user, single task system and it still fits in the machine, it has little impact on how you feel about the world. If you double the size again, and it won't fit in, then you have a problem. In terms of Heisenberg, you should make your program as fast as possible by taking up all the room in your machine.

All these comments only apply to compute bound jobs. If you have a program that is spending most of it's time waiting for you to type numbers in through the keyboard, then probably a course in typing would help. If the program is doing lots of floppy disk I/O then a hard disk may be the way to go. However, you must look at each situation afresh. Years ago, I was tutoring computer science students and every one of them thought that disk was faster than magnetic tape. I long forget the figures, but here's an example where tape is faster. Suppose you are reading short records and doing some computing, and then reading the next record. On disk, the record you want has just passed under the read head and you have to wait a complete revolution of the disk before

See FAST PROGRAMS on page 6

Fast Programs...

continued

you can get to it. On tape, you don't have any problem. [There are obvious remedies to the situation, such as buffering a complete track of the disk in memory]

Many of you have heard my comments about the 8087 chip. The chip does floating point arithmetic much faster than the software routines in the 8088. The Sieve benchmark takes 2020 seconds on the BASIC interpreter using integer variables; and it takes 2430 seconds using single precision variables (i.e. floating point arithmetic). If the 8087 can do single precision arithmetic as fast as the 8088 can do integer, then the 8087 will improve your speed by 17%. That would be rather disappointing if you expected a ten-fold increase in speed.

You may not be interested in trying to speed up a program. If you would spend more time making it faster than you would save overall, then it's not worth it. If you are going to execute the program many times, or speed will help your program compete in the market-place, it may be worth it. If the execution time is going to be about 5 minutes and you want to go and watch the football, you could just let it run. If you have a one time calculation and it's going to take 6 hours, you could run it overnight.

Usually the most dramatic increase in speed comes from solving a problem in a much better way. There is an old problem about two people on bicycles two miles apart heading towards each other. One is going at 8 miles/hour and the other at 12. A wasp which could fly at 60 miles/hour started from one rider and flew towards the other. When the wasp got there, it turned around and headed back; and flew back and forth until the two riders passed each other. How far did the wasp fly? There are two ways of solving this problem. The obvious way is to calculate the distance the wasp flew till it met the other rider, then calculate the second leg, etc. and add them all up. The other way of doing it is to consider that the two riders are closing at 20 miles/hour, the wasp can fly three times that speed, so while the riders cover two miles, the wasp will cover 6 miles. Look at the difference in the arithmetic involved!

Neil

Program Fast ...

continued

Back to the western world. Any computer program is best done by one person. No two people can understand each other as well as I can understand myself. There is no confusion about whose responsibility it is to see that an array gets initialized; if you are doing the whole thing, then it's your responsibility. You can't complain that this function is not well documented; you wrote it. On the other hand, if you don't like the way something is done, then you can change it.

If you are writing a large system all by yourself, it becomes much harder to prove it to be correct. If you wrote it, how could it possibly be a bad way of doing it? You just can't check your own code. The best approach here is to build in extra code to constantly check the reasonableness of the data you are dealing with. If something should only be 42 or 77, then test for both 42 and 77. If it's not either, then give a very loud and rude error message. (The theory is that only you will ever see it!) Build a system which either works perfectly or it falls on it's butt. It is best to catch the train going off the rails as soon as it happens. If you only find the wreckage 47 milliseconds after the derailment, the machine has done about 20,000 instructions and it's difficult to find what happened.

A second line of defence is to build "instrumented" systems. The machine is a fantastic tool to show what is happening inside the program. Use tools like this to check at every stage that the program is really doing what you want it to do. The machine will do exactly what you tell it to do. You have to be sure that what you tell it to do and what you want it to do are identical. Both the continual checking and the instrumentation must be designed into the system right from the start. They cannot be added in later on.

And now for the best trick of them all; the theory of the point of maximum confusion. (Not to be confused with Confucius). When you start a project, you have not done anything and hence there is no confusion. When you finish a project there is no confusion. If there is, then you haven't finished.

See PROGRAM FAST on page 7

Program Fast . . .

continued

Somewhere in the middle is the point of maximum confusion. The art of maintaining your sanity in a large project is to work at minimizing the maximum confusion. You should work so you tackle easily chewable pieces of the project, and return as closely as possible to zero confusion. Whenever you stop work on the project you should have a working system; it won't be complete, but it should be working.

The TOTPOMC (as it's affectionately called amongst it's followers) also tells you which path to follow through the project. Obviously the path which yields the least confusion is the path to follow. Once again, only experience can tell you how to find that path. Once you become used to the subtleties of

the TOTPOMC it becomes part of the design process, and leads you along the correct path. [That is the point of the theory of maximum confusion.]

A few things I forgot to mention. First of all get a complete set of manuals and read them. Words of wisdom remain so many blobs of ink if they are not read. If possible, get several books on the same subject by different authors and go through them all. Make sure you know the language you are using inside and out. Make sure you are certain about what you want to do and that you know exactly how you want to do it. Don't be afraid to count on your fingers.

The very last piece of advice. Follow these rules for at least five years. Ten years would be better, and twenty years better still.

Neil



BEGINNERS' CORNER

By Mike Durbin

I have had my PC for ten months now and you would think, after that much time, that I would have discovered just about everything of a fundamental nature that there was to know about the PC. But, of course, you would be wrong, micro-breath, if you should think such an erroneous thought. Why just the other day I discovered something which I should have taken note of the very first day I brought my PC home. Of course I was all excited on that particular day, so my failure to notice this particular fact is certainly understandable. And after failing to notice the fact in question on that first perusal of my new toy my senses must have become numb to the fact. Only thus can I justify the fact that now, after ten months, I have finally discovered what a yucky color IBM chose for the cover of the BASIC Programming Manual which comes packed with each and every little PC. All the other PC manuals which IBM puts to print have nice appealing colors, but the BASIC book comes to us in a sickly old yellowish-brown which I now realize has absolutely no aesthetic appeal to me at all. You will notice that I did say "TO ME" since I am sure there are some of you out there who absolutely love brown, and that is certainly within the bounds of your constitutional freedoms. As for me, however, IBM has done me no favor by so coloring the one book in the PC library which I make reference to more often than any other. And the color which I like the most, baby blue, they have wasted on their Asynchronous Communications Support

Manual, to which I absolutely never refer. So thanks a lot, big Blue, for blowing it in your color selection for the PC BASIC Manual. I had previously noted that not nearly as many IBM PC owners were into Basic programming as owners of other micros, and I really had not been able to pinpoint any good reason for this fact. Now I know! It's the color of the book. Face it! Would you put negative vibes into your subconscious mind by studying a book encased in yellowish-brown cardboard? Really now!

I doubt if many PC owners appreciate the way that IBM implemented the pause (CTRL + NUM LOCK) and break (CTRL + SCROLL LOCK) functions. The two-handed activation sequences required seem unnecessary and awkward (not to mention having to set down one's cup of coffee). The situation can be ameliorated somewhat as follows: to pause the machine, depress simultaneously the CTRL and 'S' keys; to stop the machine (perform a 'break') depress simultaneously the CTRL and 'C' keys. You will note that these multiple-key sequences can at least be performed with one hand, though it may require some practice to master the timing.

At the March meeting the Beginners' group's special subject will be Basic Programming; we will try to overcome the yucky yellowish-brown color chosen by IBM and make the Basic Programming Manual pleasant to all.

CHALLENGE OF THE 80'S
COMPUTERS

COMPUTER EDUCATION CONSULTANTS

WILL HELP YOU MEET THIS CHALLENGE!

ADULT CLASSES

ADULT COMPUTER LITERACY
MARCH 22, 24 & 29
7:00-9:00 p.m. \$75.00

HOME ACCOUNTANT+
WED. MARCH 30 & APRIL 6
7:00-9:00 p.m. \$60.00

ADULT COMPUTER LITERACY
APRIL 7, 11 & 14
7:00-9:00 p.m. \$75.00

BUSINESS CLASSES

VISICALC
MARCH 21, 23 & 28
7:00-9:00 p.m. \$100.00

APPLE WRITER II
APRIL 19, 21 & 26
7:00-9:00 p.m. \$100.00

EASY WRITER II
APRIL 11, 13 & 15
1:00-3:00 p.m. \$100.00

or
APRIL 13, 18 & 20
7:00-9:00 p.m. \$100.00

BUSINESS & THE P.C.
APRIL 12
1:00-4:00 p.m. \$35.00

INDIVIDUAL
CONSULTATION
\$50.00 per HOUR

STUDENT CLASSES

BEGINNING BASIC
TUES. MARCH 22 & 29
4:30-6:30 p.m. \$50.00

BASIC PROGRAMMING
THURS. APRIL 7, 14, 21 & 28
4:30-6:30 p.m. \$125.00

GRAPHICS
APRIL 4, 6, 11, 13, 18, 20, 25 & 27
5:30-6:30 p.m. \$125.00

COMPUTER TYPING
MON. APRIL 11, 18, 25 & MAY 2
4:30-5:30 p.m. \$50.00

REGISTER NOW!
SUMMER CLASSES FILLING RAPIDLY.

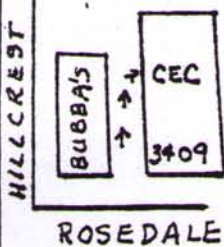
CALL 363-3715 TO MAKE RESERVATIONS OR COME BY THE COMPUTER LEARNING CENTER AT 3409 ROSEDALE SUITE 200.

CLASSES TAUGHT ON APPLE II+ AND IBM PC COMPUTERS WITH "HANDS-ON" LESSONS

LIMIT ENROLLMENT

SOFTWARE SOLD AT BELOW RETAIL PRICES

S.M.U.



PC COMM



by Chris Jacobs

What do you do with all the money that your PC is earning for you? Use the Dow Jones News/Retrieval Service to invest it and make more money.

How do you access Dow Jones?

1. Use your PC, modem, and communication software as described in earlier articles

2. Subscribe to Dow Jones News/Retrieval (suggested route: Radio Shack package number 26-2224, approx. \$19.95)

The package listed above does not contain software, only the subscription information for Compuserve and Dow Jones. The package will save you the \$50.00 Dow Jones initiation fee and will give you free time on both systems as well. It is well worth the money.

What does Dow Jones give you:

Dow Jones News	Dow Jones Quotes/Current
Corporate Earnings Estimator	Dow Jones Quotes/Historical
Disclosure II	Media Gen Financial Serv.
Weekly Economic Survey	Weekly Economic Update
Wall Street Journal Hilites	Wall Street Week Online
News/Retrieval Sports Report	News/Retrieval Weather Rpt
Cineman Movie Reviews	Online Encyclopedia
Free Text Search	Free Update For Users

and much, much more. For instance:

Dow Jones News:

- 80 news categories
- 6000 companies
- 50 industries
- 700 Canadian companies
- Wall Street Journal, Barron's, Dow Jones News Service
- stories as recent as 90 seconds, as far back as 90 days

Dow Jones Quotes:

- common stock, preferred stock, warrants
- corporate and foreign bonds
- mutual funds
- U.S. Treasury issues
- options
- monthly summaries back to 1979
- quarterly summaries back to 1978

Wall Street Journal:

- electronic version of the printed edition
- as early as 7 AM EST

Free Text Search:

- search for any combination of words, dates, or numbers
- back to June 1979
- search for specific data on any subject in Dow Jones

The features are too numerous to number, let alone list. I suggest you try Dow Jones if you have any interest in investments or finance.

What does it cost to use Dow Jones:

Depending on the database from

- \$.90 per minute to
- \$1.20 per minute

Seems like a lot of money, right? However, to access and retrieve your needed information does not take a lot of time and can be done for the average portfolio for as little as 15 minutes per month.

As an added plus, Dow Jones is preparing software for the IBM PC that will work with their data and will automatically update and evaluate your personal portfolio, analyze your earning trends, and other sophisticated analyses.

If you try Dow Jones, you will find your needed financial and economic information, fast.

Next month, a surprise database for bibliographic data.

Chris





Programming Topics

Can A Programmer Do It All?

It's finally finished! You have written and debugged the most marvelous application program the world has ever seen. Now all you have to do is tell the world about it, and you are going to be a very rich person. Visi-Calc scenarios are running through your head.

If your market (the people who will buy and use your program) is intended to be people who are not computer programmers, the hardest part of your job has just begun. First, you must be able to convince a population of people who are not like yourself that they need to use your program, and, secondly, you must tell them how to use it. If you have never done this before, it may sound easy. Believe me, it is not. In fact, you should carefully examine whether you are even qualified to sell and document your program.

Let's examine your market a little closer. Who will buy your program? It may not be the same person as the end-user. This is especially true if you have a program that may be used by medium or large companies. This selling job must be done in their terms, i.e., productivity gains, cost savings, etc. You won't even mention the special features and all those bells and whistles to this crowd. This seems to be extremely hard for most programmers to do. Unless you are convinced that your buyer and end-user is the same person, screen formats and example dialogs are rarely appropriate.

The second aspect of your market is those who will be using your program. Again, as a programmer, are you qualified to write that all important overview? You need all of the expertise of a good novel writer to get their attention in the first chapter. And you don't need a feature by feature explanation of how to do it. You need a feature by feature explanation of what it does for them. The command syntax is of interest only after a person has been convinced that it is going to do something useful. The documentation must be speaking the users' language, using terminology that they understand. Without this, all those wonderful features may never be used.

I would like to suggest that four distinct phases are required in the marketing of a software product.

1. Establishing why you need it.
2. Explaining what it does.
3. Explaining how to do it.
4. Documenting how it gets done.

I would also like to suggest that three very different people are needed to successfully accomplish this.

Programmer

Application expert

Writer

We all know the programmer is essential for phase (4). The programmer is needed for phase (3), but so is the application expert. Another term often used for (3) is the user interface. Would it surprise you if I said the application expert is rarely a computer programmer? Phase (4) may not even be necessary in the marketing of the product. It is sometimes a carefully guarded secret.

The application expert is essential for (3) and Phase (2). This expert must communicate with the programmer to develop the user interface. The expert must communicate with the writer to develop (2). The programmer is not even needed for (1) and (2). The writer is required to communicate with the expert to come up with an understandable description of (2). The writer is responsible for (1).

A computer program is a a complicated product and it requires a carefully assembled group of people to be successful. Attempting to wear more than one hat is often very costly to the product, and often disastrous.

Tom Prickett



Programming Topics

(continued)

Print Spoolers and Disk Emulators

Last month I provided an introduction on print spoolers and showed how extra memory on your machine could be put to good use in providing better productivity. Now the disk emulator, or electronic disk will be discussed, and then we'll look at how this type of software is packaged.

A disk emulator is a device or program that imitates the activities of a disk drive and diskette. This discussion is limited to the software type of emulator, which is frequently referred to as an electronic disk because it uses the electronic memory of your computer to perform this task. This imitation of a mechanical disk drive accepts the same data and achieves the same results as an actual drive - but with one major difference - it is much faster. There is no wait for the drive motor to start or for the diskette to reach the proper speed for writing or reading. There's no read/write head to be stepped to the proper recording track and no delay while the diskette rotates to the sector of data which is being accessed. All of these mechanical delays are eliminated; the data only needs to be moved from one area of the system's memory to another. The electronic disk operates up to 100 times faster than a mechanical drive, so the portions of your work that require extensive disk input and output can be speeded up tremendously by placing your programs and data files on the electronic disk.

There are a couple of disadvantages in using disk emulators, however. A large amount of extra memory may be necessary, depending on your needs; to emulate a double sided drive, for example, would require 320K of memory to be allocated to this function. If you have important data on an electronic disk and lose power, all of that data will be lost. Using the disk emulator, therefore, requires some amount of planning. You must plan on how much memory to set aside for the disk, what to put on it, when to put it on, and when, if necessary, to copy it back to a regular diskette for safekeeping. But if your application can be speeded up and the system's response improved, it is well worth the extra trouble. And you might even be able to get by with fewer diskette drives than you thought you needed.

For an example, let's pull out that old Slow-Typer word processing program we used last month and try to speed it up. Slow-Typer is not very smart - it was written a long time ago to run on a less advanced machine and then translated to run on the IBM PC. It doesn't know how to use the 320KB of memory on your machine; all it can use is 64KB. It is very well aware of diskettes, however and makes generous use the disk. Part of the file you are working on is placed on a temporary "spill file" on disk and portions of the program itself are brought in from disk whenever you request a function such as paragraph alignment or line deletion. As you use the program, it continuously pauses, sometimes for several seconds, while it grinds away on the diskette. This was alright several years ago when cassette tape and paper tape were the going thing in storage devices, but these delays aren't necessary today.

To speed up Slow-Typer, we need to do a few things before starting the editing session. First the system has to be started with the electronic disk active. For most electronic disk programs, this can be done automatically once you have installed the program on your diskette. Then we have to copy a few files to the electronic disk. Let's assume that the Slow-Typer program is on a file named "ST.COM" and that it has additional parts of the program on files named "STOVLY1.OVR" and "STMSG5.OVR". Also assume that we want to do some extensive editing operations on a document named "INSTRUC.DOC", which resides on Drive B. To get things started, we type the following commands:

```
COPY A:ST*.# C:
COPY B:INSTRUC.DOC C:
C:
ST INSTRUC.DOC
```

The first statement copies all of the Slow-Typer files on Drive A that start with "ST" to Drive C, which is our electronic drive. The next statement copies the document file we are editing to the electronic drive. The third statement sets Drive C as the "default", or standard drive, so that Slow-Typer will always look there first for programs and data. And the final statement actually starts Slow-Typer and gives it the name of the document to edit. These statements could be put into a batch file if we tend to perform this process very often - then one command could start the entire process.

▷

Programming Topics

(continued)

As you begin editing using the combination of Slow-Typer and the electronic disk, you can feel quite a difference in the way the system responds. Paging through the document a screen at a time, or even from the begin to the end of the document is now very quick. The additional functions such as text deletion and movement of text from one part of the document to another are now instantaneous. The time savings here only amounts to several seconds at a time, but this can have a positive influence on your productivity. You tend to stay alert and are able to keep your train of thought as you work on the more responsive system.

When you complete your editing session and end Slow-Typer, a couple of things need to be done. First, the document needs to be copied back to your data diskette in Drive B. This could have been avoided if you had saved the file directly to that drive while you were using Slow-Typer. The second thing to do is to restore Drive A as the default drive. The following two statements handle the situation:

```
COPY INSTRU.DOC B:
```

```
A:
```

Note that these statements could also have been placed into the batch file which controls the entire edit session with one command.

Another type of application that can benefit from an electronic disk is one which has extensive disk input/output, such as a sorting or merging program, a database application, or a program compiler. Sometimes the time savings on this type of application can be much greater. This also eliminates a lot of wear on the diskette drive.

Once you decide you want to use a disk emulator and/or a print spooler, you need to determine which features are important and which product meets your needs. There are as many different products of this type as there are spreadsheet applications, and even though they achieve similar results, the way they are implemented and the features they offer can make a significant difference in their usability.

One of the primary considerations is in the method that the spooler/disk emulator is activated. This type of program is different from most types of applications

because it must "hook in" to the system, which means that it must find a safe place in memory to hide itself and its data, it has to fool DOS into thinking that there is less memory now available, and it has to supplement the normal printer and/or diskette handling part of the system with additional logic to simulate the additional functions. Since DOS versions prior to 2.0 have no standard way of adding these types of programs, each software producer has to take DOS apart to find out how it works and how to add the new functions. Therefore the end result varies considerably. It also makes these programs incompatible with one another and with similar software such as hard disk support.

There are basically two different ways that these programs are added to the system. In one method, DOS is actually modified. In the other method, an

**...also eliminates
a lot of wear on the
diskette drive.**

additional command must be given to activate the new functions. There are also a couple of products that require both a modification and a command. Each method has its advantages and disadvantages. The DOS modification method, for instance requires running an installation program which alters one or both of the "hidden" DOS files on your diskettes. The disadvantage of this method is that you must start your system only with the modified DOS diskette to have the new functions available. An advantage of this method is that this will happen automatically each time the system is started with the modified diskette.

The command method does not require changing the DOS files; instead, a new command is available or a program is run which hooks the new functions into the system. This method can be made automatic by including the command in the AUTOEXEC.BAT file, which is the file of commands that are given automatically upon system start. The advantages of this method are that the command can reside on any of your diskettes and that the DOS files are not changed. The disadvantage is that the AUTOEXEC.BAT file must be created or modified to support automatic start of the new functions. ▶

Programming Topics

(continued)

Another important difference between various products is in the machine setup requirements. Some spoolers/disk emulators require special switch settings of the option switches on the system board inside the machine. This might be necessary to tell DOS that there are additional diskette drives available or to control the amount of memory that DOS finds. It may be necessary to install memory at a very non-standard location in order to hide it from DOS. These requirements can cause big problems for you if you need to switch to different operating systems or to operate without the added functions.

The amount of memory to be used for the print spooler and disk emulator is also specified in different manners. One method requires setting the disk amount

Some spoolers/disk emulators require special switch settings..

at install time when DOS is modified and the print spooler amount after the system is started with the electronic disk active. In this case the spool amount reduces the disk amount. Most of the products use the command method of startup and specify the memory sizes by parameters on the command line. There are differences here, too. You might have to make your memory request in terms of kilobytes, 512 byte blocks, 32KB blocks, or in other special codes. One print spooling program is actually supplied as a number of programs with the program size built into the last part of the name.

The maximum print buffer space is 64KB on most of the spoolers. Some of the electronic disk emulators limit the disk space to a maximum of 160KB (equivalent to a single sided drive), some to 320KB (same as a double sided drive), and others allow even more. This space can be specified as either one or more electronic drives, depending on the program. When both a print spooler and disk emulator are supplied by the same company, you might find a single program that handles both functions or separate programs for the spooling and disk emulation functions, depending on the product.

There are a number of features and other frills that can be found in the various products. While no one product has all of these features, you might find a certain combination of the following features desirable for your environment:

- . Clear Printer Buffer. This feature is almost a necessity, since you can generate printed output so quickly. It provides a means to delete, or kill printer data that is in the spooling buffer but not yet printed, and is implemented in a number of ways. One method is to have you turn the printer off for a few seconds while the output goes off into never never land. Another method requires you to run a special program to clear the data. This method could not be used if a program fills the print buffer and is waiting for the printer to complete printing. You would have to either wait or restart the machine. A more desirable method is one which allows a special keyboard command sequence at anytime that performs this function.

- . Pause the Printer. This convenience item allows you to temporarily halt the output of data to the printer. Your applications can continue to print to the print buffer area (unless it fills). You can then resume the printing operation at any time. This function is either controlled by a special program or keyboard command sequence.

- . Additional Keyboard Functions. A spooler/disk emulator that uses the keyboard method of controlling the pause and clear functions might have a number of additional frills that are activated by keyboard commands, such as the ability to insert printer setup characters (for setting compressed or expanded mode), or a one-key display screen pause.

- . Drive Reassignment. Since many applications are internally coded to use certain drives, such as Drive A for the program and Drive B for the data, it is difficult to use an electronic drive if it is Drive C. This function allows you to change the drive assignments around. For example, you could copy all of the programs on Drive A to Drive C (the electronic drive), and then use this function to now make Drive A refer to the electronic drive. This greatly simplifies the usage of the electronic drive. ▶

Programming Topics

(continued)

. **Support For Nonstandard Disk Formats.** At least one program provides support for additional sectors per track (10 instead of 8 for a 25% increase), quad density drives, 8" drives, and a number of hard disks.

. **Restartable Electronic Disk Area.** A couple of the programs allow the system to be restarted (reset) while retaining the contents of the electronic disk drive. This can be a useful feature, but must be used with caution if you must reset the system to recover from an error. In this case, the data on the electronic disk might have been damaged due to the error.

. **Quick System Start.** At least one program allows you to set the computer's option switches to indicate a small amount of memory, such as 64KB, even on a system with additional memory installed. This function will quickly search for the additional memory, bypassing the lengthy power-on memory checkout procedure.

. **Printer Switching.** Several of the programs allow spooling to be switched between the first, second, and possibly a third printer attached to the system.

. **Serial Printer Support.** Several of the programs allow spooling to a serial printer attached to the RS232 communications adapter.

. **Adjust Printer Priority.** A couple of the programs allow an adjustment of the time spent on the printing function. More time can be assigned to support a faster printer at its rated speed and less time can be assigned if the printer is causing interference to higher speed modem communications (1200 BPS or greater).

. **Status Report.** Some of the programs provide a method of reporting on the amount of memory allocated to the various functions. This might be part of the start-up process or might be a separate program, depending on the product.

As you can see, there are quite a few features to be found in these programs, and no single program will have all of these. This list is not complete, either, as new features crop up all the time and there are a number of the programs that I am not familiar with. I will conclude this series next month with additional information and some techniques for using the products.

Michael Wiles



PC Update

The following news items are reprinted with permission from Computer Industry Update published by Industry Market Reports, Los Altos, CA.

IBM selected the PFS:File and PFS:Report by Software Publishing of Mountain View, CA as part of the Personal Computer software line. They are priced at \$40 and \$125 respectively.

Celtic Technology is offering a video film recorder, the VP-01-35, to provide the IBM PC user with the ability to produce presentation quality 35 mm slides for photographic prints of the computer's CRT images. The small unit is color coordinated with the PC but can be used with a variety of other computers. It costs \$2,495 and is being distributed by Computer Mate of Richardson, TX.

CXI of Los Altos, CA announced the Pcox interface board and software which allows a Personal Computer to use coaxial cable to connect to the Model 3274 and 3276 cluster controller. The interface board is \$910 while software is licensed for \$285.

FutureNet of Canoga Park, Ca introduced a schematic designer, the Dash I system, which turns the PC into a workstation for the design engineer. The system includes software, a proprietary graphics controller, a "mouse", a memory network interface card and a printer/plotter added to the PC. It has RAM memory of 256K bytes for system software and 64K bytes of buffer storage. Installation in existing PCs requires no hardware or software modification. The Dash I costs \$5,960 as an addition to the PC or \$12,960 for a complete system.

Keytronic of Spokane, WA is offering the PC-compatible P2402-19 keyboard with the left shift and return keys in familiar locations. It is available with serial ASCII TTL or RS422 interface. The keyboard costs \$235 in quantities of 1-9 and is available from stock.

Quadram of Norcross, GA is offering the QuadScreen 17" monitor with 64K bytes of memory for the PC. The display is 160 characters wide by 64 lines using a 5 by 7 character matrix, and the bit-mapped graphics display has a resolution of 960 by 512. It has reverse video and forward and backward smooth scroll. The unit is available with white or green phosphors for \$1,950. >



PC Update
(continued)

Bytek of Berkeley, CA announced the Cogen Cobol program generator for the PC. It is \$950.

Digital Research reduced the price for CP/M-86 to \$60.

GMS Systems of New York, NY introduced Powerbase, a relational data base management system initially designed for use with the PC using the PC-DOS operating system. It has a search, sort, and select feature, special search keys for "sounds like" retrieval, a front end editor, and capability for handling arithmetic expressions. It costs \$475.

Graphic Communications of Waltham, MA is offering a version of its Graphwriter business software for the Personal Computer. The basic set is \$395 and the basic and extended sets are \$750.

Hayes Microcomputer Products of Norcross, GA introduced the Smartcom II communications software for use with the PC and the Smartmodem 300 or the Smartmodem 1200. It can route data to disk or printer. It costs \$119 and became available in February.

Mark of the Unicorn of Arlington, MA announced the PC/Intervox which provides DEC VT-100 emulation for the Personal Computer for \$99. A similar product is offered by Persoft of Stoughton, MI. The TE100 Version 1.1 is \$125. A bidirectional file transfer version is \$150.

Stoneware of San Rafael, Ca announced the DB Master file management program for use on the PC. It allows a 3,000 character record. The price is \$499.

Strohl Systems Group of Plymouth Meeting, PA introduced a menu and function key driven stock system for the PC which logs on to the Dow Jones News Retrieval Service. It also automatically updates stock and historical prices daily. It costs \$495.

Synergistic Software of Renton, WA is offering the Data Reporter data base management package for the PC. It is \$250.

Techland Systems of Mount Vernon, NY is offering a schedule and expense management package and an investment management package for the PC. Shoebox 1.1 is a system designed to help executives work more efficiently. It is \$295. The Portfolio Master PC performs a variety of investment-related procedures and links with the Dow Jones subscription service. It is \$195.

Today Graphic Communications of Waltham, MA is offering a business graphics software package for the PC. It is \$395.

User-Friendly Software of Melville, NY is offering the Apartment House Manager system of apartment building managers. It runs on the PC and is \$395.

VisiCorp of San Jose, CA introduced VisiWord and VisiSpell word processing packages for the PC. VisiWord retails for \$375, and VisiSpell is \$225. Both will be available in the first Quarter.



**SHOP
& SWAP**

(Four lines free each month to members, otherwise cost is 15 cents per word. Ads will run three consecutive months unless notified earlier to cancel. Ads must be renewed after three months if continuation is desired. Mail ads to the Editor.)

INSTRUCTION COURSES for IBM PC (Architecture, DOS, BASIC, WORDSTAR, VISICALC, dBase II); Private custom programming services (free estimates). Contact Mike Durbin at 840-9344.

WANTED: Advertisers for PC NEWS. Rates are reasonable, exposure is great! Circulation approximately 140 PC users/owners. Contact J.P. Pribyl, at phone 817/275-4109 for particulars.

Street Electronics' ECHO-6P VOICE Synthesizer. \$299. COMPUTER GOLD Microsystems, 2417 Coleshire Drive, Plano, TX 75075. We accept Visa or M/C. Phone (214) 596-2598. Southwest dealer for Street Electronics

FREE Consultants Directory Listing: DEADLINE MAY 1
Microsystem Consultants Agency, 2116 E. Arapaho Suite 155, Richardson TX 75081, or call 690-4224. Extra lines, listings, ads avail., starting at \$5

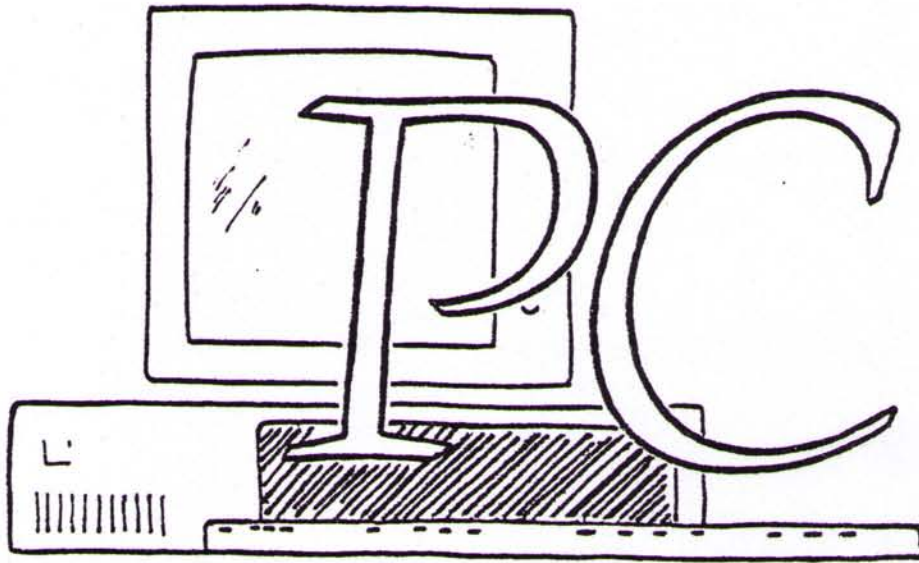
If you know someone who is interested in the IBM Personal Computer and is not now a member of our Group, fill in their name and address and send the form to the Editor. We'll send them a complimentary copy of the News and a membership application blank.

 Name ----- NEW APPLICANT -----
 Address ----- CHANGES -----

 City ----- State ----- ZIP -----
 Phone: Home ----- Metro? -----
 Work ----- Metro? -----

by:

18



North Texas PC News
2025 Rockcreek Drive
Arlington, TX 76010



FIRST CLASS

