

BYTE

12490

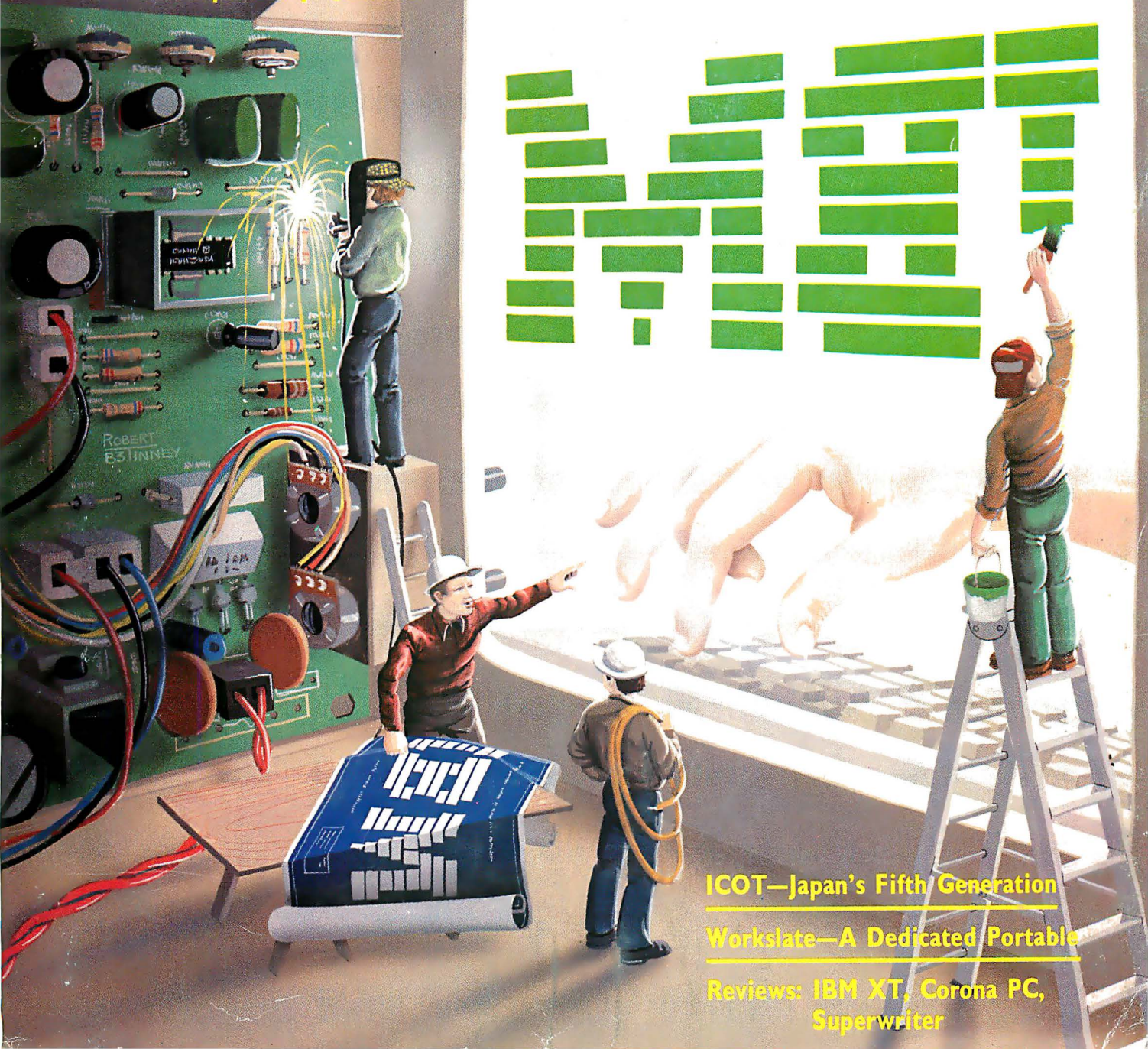


NOVEMBER 1983 Vol. 8, No. 11
\$3.50 in USA
\$3.95 in Canada/£2.10 in U.K.
A McGraw-Hill Publication
0360-5280

the small systems journal

INSIDE THE IBM PC

- Hundreds of Peripheral Boards
- Big Blue Goes Japanese
- The Compatibility Question



ICOT—Japan's Fifth Generation

Workslate—A Dedicated Portable

Reviews: IBM XT, Corona PC,
Superwriter



Inside Apple

Vol. 1, No. 3

Apple's new Monitor II. A sight for sore eyes.

If you've been using a TV as a monitor, perhaps you can get a friend to read this for you:

Apple's brand new Monitor II will improve your vision.

It features all the latest ergonomic improvements in monitor technology.

For example:

Studies have shown that the leading cause of eye fatigue for computer users is lack of contrast between the displayed characters and their background.

So we designed the Monitor II around a high contrast green phosphor CRT that provides an extremely dark background. That means you can read text at a lower brightness. And that means you can be more productive — working longer and more comfortably.

Toward that same end, we also gave Monitor II a tilt screen. So you can angle it perfectly for your working position, without scooting your chair around or sitting on phone books.

And we made that screen antireflective to reduce glare from ambient light.

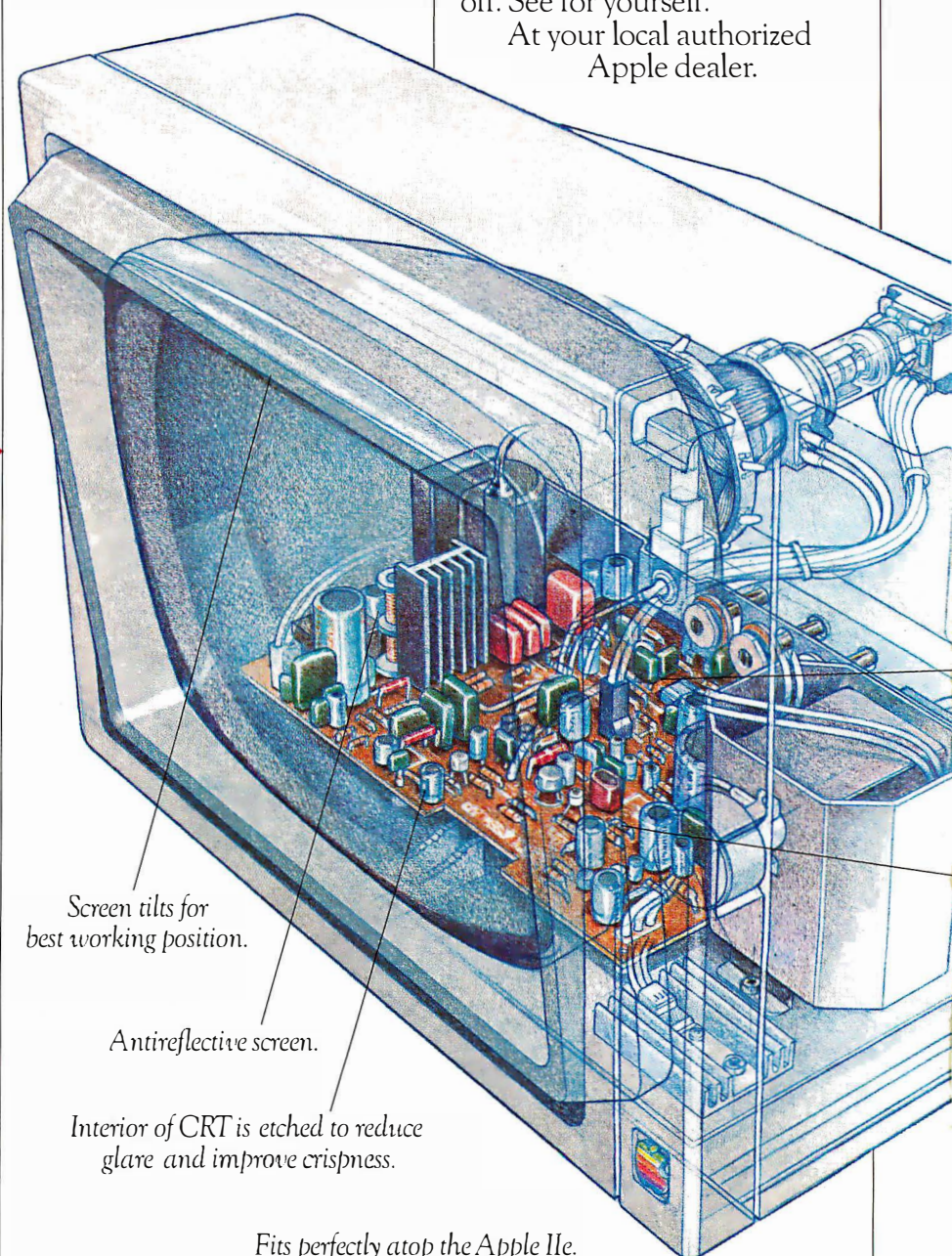
Monitor II also features a high bandwidth video amplifier and a high tolerance linearity circuit. The former keeps characters from smearing

on the screen and eliminates the annoying "ghosts" left by a fast moving cursor. The latter keeps characters crisp, legible and prevents "keystoning" right up to the edges of the display. Both add up to superior display of 80-column text and extremely

accurate graphics.

Designed as the perfect system partner for the Apple IIe Personal Computer, Monitor II requires no monitor stand. It's a perfect fit, aesthetically as well as technically. So it's pleasing to the eye even when it's turned off. See for yourself.

At your local authorized Apple dealer.



Now Apple plots color.

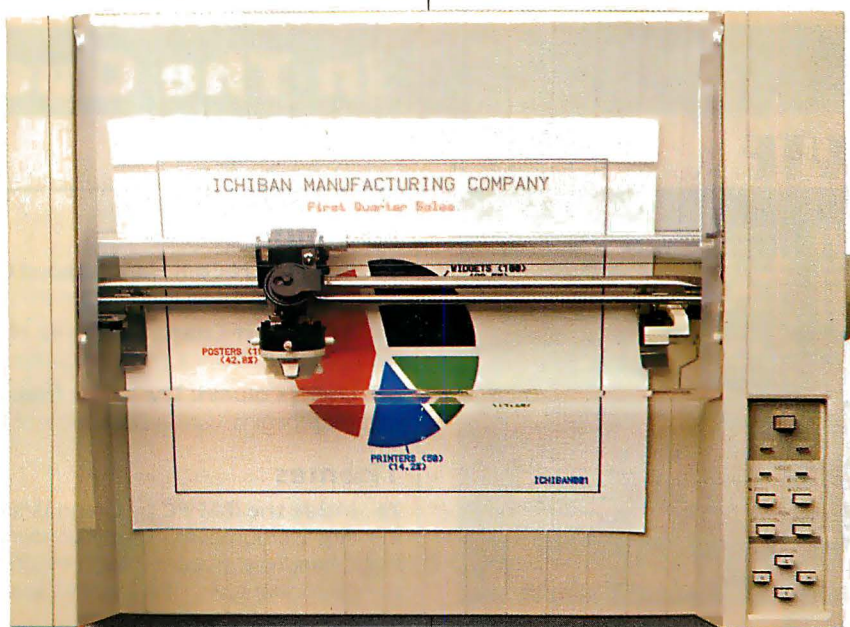
Since color graphics are becoming ever more important in business, we've been hearing more and more calls for a color plotter as reliable as an Apple.

Here it is:

Apple's new Color Plotter can generate all kinds of presentation graphics, engineering drawings or anything else you have to illustrate in up to eight brilliant colors.

And it can perform its art on any size paper up to 11" x 17". Or, with optional transparency pens, it can draw right on transparent film for overhead projection.

Measuring just 4.8" H x 16" W x 12" D, it's the smallest four-color, wide bed color plotter you can buy — about half the size of conventional flatbed plotters. So it takes up less space on your desk and can easily be



moved to someone else's desk.

There are two color plotter accessory kits to choose from to assure a perfect marriage with your Apple II or IIe, or Apple III.

Each kit comes with eight color pens — red, blue, green, black, burnt orange, gold, violet and brown. Plus a starter package of plotter paper. Plus all the manuals, documentation and cables appropriate to

your particular kind of Apple. So you can get up and coloring right away.

Apple also offers a complete selection of 24 different pen packages — so you can choose whatever colors you need in a variety of widths for a variety of applications and media types.

As you might expect, all of the above is available at many of our authorized Apple dealers.

Carry on with AppleCareSM Carry-In Service.

No matter how long you've owned your Apple system, you can now get a long term service contract at a very reasonable cost.

AppleCare Carry-In Service is a service plan that will cover most Apple-branded components in your system for one full year.

It covers an unlimited number of repairs and is honored by over 1500 authorized Apple dealers nationwide.

Apple-trained technicians assure you of the highest quality service, fast — in most cases less than 24 hours.

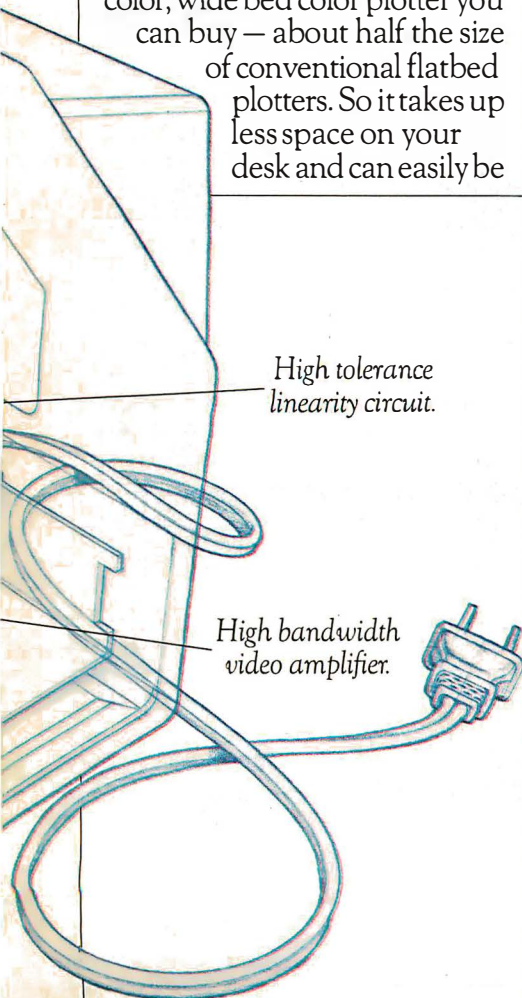


AppleCare Carry-In Service is ideal for anyone who needs to know ahead of time the cost of maintenance for their system.

So check out the details — you'll find it's the lowest cost health plan an Apple can have.

High tolerance linearity circuit.

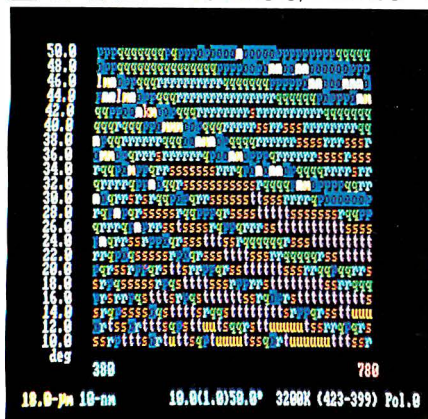
High bandwidth video amplifier.



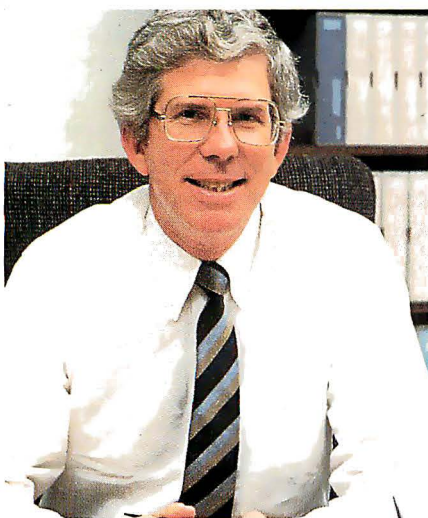
In The Queue

BYTE

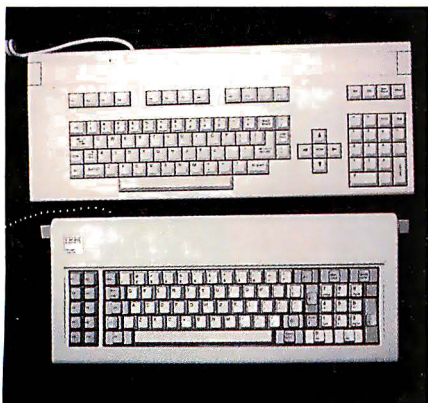
Volume 8, Number 11



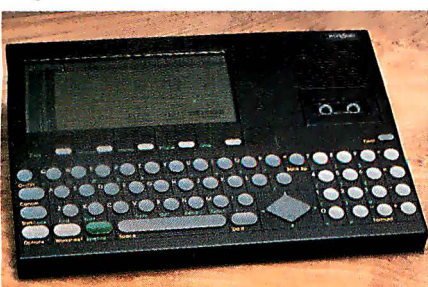
Page 78



Page 88



Page 232



Page 52

Columns

36 Build the H-Com Handicapped Communicator by Steve Ciarcia / The Intel 8748 is the basis for a scanning communicator that users can control with just one switch.

52 BYTE West Coast: California Hardware by Barbara Robertson / A look at four new products, from a portable computer to bubble-memory boards.

65 User's Column: The Latest from Chaos Manor by Jerry Pournelle / This month's potpourri begins with a discussion of disk formats.

Themes

76 Inside the IBM PC by Gregg Williams / IBM's famed Personal Computer spawned the largest group of third-party vendors the microcomputer industry has ever seen and single-handedly enabled microcomputers to assume a greater percentage of the world's computational tasks. This month's theme articles explore the ubiquitous machine from a wide variety of angles.

78 IBM PCs Do the Unexpected by Steven S. Ross / The IBM PC can conquer a fascinating array of scientific, business, and educational tasks.

88 IBM's Estridge by Lawrence J. Curran and Richard S. Shuford / In an interview with BYTE's editors, the president of IBM's Entry Systems Division talks about standards, the PC's simplicity, and a desire not to be different.

99 Enhancing Screen Displays for the IBM PC by Tim Field / With a program called Screen, you can take full advantage of the capabilities of both monochrome and color displays and adapt them to your own needs.

121 POKEing Around in the IBM PC, Part 1: Accessing System and Hardware Facilities by Hugh R. Howson / How to use BASIC's PEEK and POKE commands to realize the speed and flexibility of machine-language code without sacrificing the convenience of a high-level language.

135 Could 1,000,000 IBM PC Users Be Wrong? by Frank Gens and Chris Christiansen / Everyone knows the IBM PC has had a profound effect on the personal computer market. But what direction will it take in the future?

144 Big Blue Goes Japanese by Richard Willis / The capabilities of IBM Japan's new 5550 Multistation will make it a formidable competitor in the red-hot Japanese market.

168 Expanding on the IBM PC by Mark J. Welch / A survey of expansion boards including 17 fact-filled tables.

188 Installable Device Drivers for PC-DOS 2.0 by Tim Field / A look at the importance of device drivers and how they work with the PC.

199 A Communications Package for the IBM PC by Richard Moore and Michael Geary / How one company's communications software package evolved as a result of user feedback.

211 A Graphics Editor for the IBM PC by Charles B. Duff / A graphics editor called GLYPHE makes drawing with the PC's graphics characters fun as well as efficient.

232 Comparing the IBM PC and the TI PC by Bobbi Bullard / They may look alike, but each of these computers has its own special features.

247 Technical Aspects of IBM PC Compatibility by Charlie Montague, Dave Howse, Bob Mikkelsen, Don Rein, and Dick Mathews / The IBM PC's success paved the way for IBM PC-compatible computers. But it takes more than an 8088 board to create a plug-compatible machine. The authors explain why.

254 The Making of the IBM PC by Brian Camenker / The success of the 70-year-old International Business Machines Corporation can be explained in one word: marketing.

257 Concurrent CP/M by Joe Guzaitis / This operating system efficiently uses computer and operator resources.

BYTE is published monthly by McGraw-Hill Inc., with offices at 70 Main St., Peterborough, NH 03458, phone (603) 924-9281. Office hours: Mon—Thur 8:30 AM — 4:30 PM, Friday 8:30 AM — Noon, Eastern Time. Address subscriptions to BYTE Subscriptions, POB 590, Martinsville, NJ 08836. Address changes of address, USPS Form 3579, and fulfillment questions to BYTE Subscriptions, POB 596, Martinsville, NJ 08836. Second-class postage paid at Peterborough, NH 03458 and additional mailing offices. USPS Publication No. 528890 (ISBN 0360-5280). Postage paid at Winnipeg, Manitoba. Registration number 9321. Subscriptions are \$21 for one year, \$38 for two years, and \$55 for three years in the USA and its possessions. In Canada and Mexico, \$23 for one year, \$42

272 The IBM PC Meets Ethernet by Larry Birenbaum / By adopting Ethernet technology, IBM PCs can share peripherals and information.

285 MS-DOS 2.0: An Enhanced 16-bit Operating System by Chris Larson / The most recent version of Microsoft's popular single-user operating system offers installable device drivers, Xenix compatibility, and background tasking.

Reviews

294 The IBM PC XT and DOS 2.0 by Rowland Archer Jr. / With the XT, IBM took a conservative developmental step; PC-DOS 2.0, on the other hand, took more of a leap.

308 The Corona PC by Rich Malloy / Compatible with the IBM PC, the Corona PC features an 8088 microprocessor, 128K bytes of memory, a high-quality display, and the Multimate word-processing program.

328 A Look at the HP Series 200 Model 16 by Berry Kercheval / Hewlett-Packard's 68000-based microcomputer offers a lot of power in a small package.

352 Three Generations of Business Charts for the IBM PC by Jack Bishop / Reviews of Graphics Generator from Robert J. Brady Co., Chartmaster from Decision Resources, and Business Graphics from Business and Professional Software Inc.

370 A Versatile IBM PC Word Tool: Sorcim's Superwriter by Richard S. Shuford / A powerful and easy-to-use word-processing program, Superwriter provides many functions that are useful in a business environment.

Features

394 Japan and the Fifth Generation by Phil Lemmons / A look at Japan's efforts to develop artificial intelligence.

402 Speech Images on the IBM PC by A.J. Cote Jr. / With an experimental speech-input card, the IBM PC can plot sounds that can prove useful as speech aids for the deaf.

410 Lmodem: A Small Remote-Communication Program by David D. Clark / Written in the BDS version of the C programming language, the Lmodem program provides terminal emulation, text capture, and transfer of files.

430 The Software Tools: Unix Capabilities on Non-Unix Systems by Deborah K. Scherrer, Philip H. Scherrer, Thomas H. Strong, and Samuel J. Penny / This package includes utility programs, a command interpreter, and a large programming library.

449 Double the Apple II's Color Choices by Robert H. Sturges Jr. / How to get your Apple II to provide a wide selection of colors without sacrificing resolution.

467 A Character Editor for the IBM PC by Raymond A. Diedrichs / A BASIC program called Font lets you substitute custom symbols for a portion of the computer's standard character set.

560 Statistical Programs for Microcomputers by Peter A. Lachenbruch / Test the accuracy of statistical microcomputer software with these tools.

Nucleus

- 4 Editorial: Growth vs. Quality
- 7 MICROBYTES
- 12 Letters
- 481, 502, 518, 524, 552
Programming Quickies
- 487, 494, 507 Technical Forums
- 532, 538, 544 Book Reviews
- 575 User to User
- 591 Clubs and Newsletters

- 596 Ask BYTE
- 610 Software Received
- 622 Event Queue
- 640 Books Received
- 646 What's New?
- 717 Unclassified
- 718 BYTE's Ongoing Monitor Box
and BOMB Results
- 719 Reader's Service

Cover painting by Robert Tinney

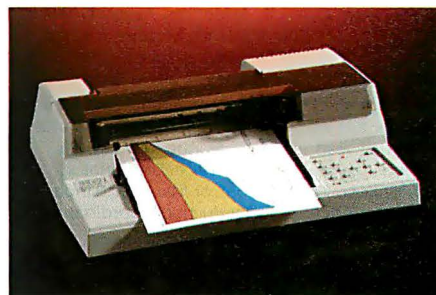
for two years, \$61 for three years, \$53 for one year air delivery to Europe, 17,100 yen for one year surface delivery to Japan, \$37 surface delivery elsewhere. Air delivery to selected areas at additional rates upon request. Single copy price is \$3.50 in the USA and its possessions, \$3.95 in Canada and Mexico, \$4.50 in Europe, and \$5.00 elsewhere. Foreign subscriptions and sales should be remitted in United States funds drawn on a US bank. Printed in the United States of America.

Subscription questions or problems should be addressed to:

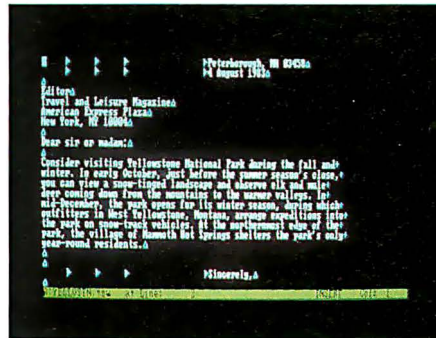
BYTE Subscriber Service, POB 328, Hancock, NH 03449



Page 308



Page 328



Page 370



Page 394

The Software Tools

Unix Capabilities on Non-Unix Systems

This package includes utility programs, a command interpreter, and a large programming library

by Deborah K. Scherrer, Philip H. Scherrer, Thomas H. Strong, and Samuel J. Penny

The Software Tools package is a set of programs and subroutines that provides the power and elegance of Bell Laboratories' Unix on non-Unix computer systems. The tools offer Unix-like program development features that complement systems ranging from microcomputers to mainframes.

Available in various forms from several sources, the Software Tools package includes more than 60 utility programs, a command interpreter (*shell*), and a large programming library.

Code sharing, coupled with early feedback from users, has allowed developers to build on each other's work and has produced a dynamic environment in which new ideas are rapidly tried and proven. The natural selection process that results produces high-quality, useful utilities that have been tried, improved, tested, and accepted by many users with varying needs and a variety of systems.

The Tools

The Software Tools utilities provide a framework for executing most common computing tasks. Each tool is a powerful but general software module designed to do one thing well.

The tools are easy to learn and use. They perform functions such as organizing and manipulating files,

creating, editing, and rearranging text, examining files, preparing documents, and transforming language and data. Frequently used tools are:

diff	determines the differences between two files
ls	lists the file names in a directory
ar	maintains multiple small files nested inside a larger one
sort	sorts lines of a text file in several ways
find	locates text patterns in a file using a flexible expression syntax
field	rearranges data columns in a file
sedit	performs serial editing functions on a file
format	formats a document for publication or distribution

The complete set of Software Tools provides most of the functional capabilities of the Unix tools. Table 1 is a list of the tools and their Unix equivalents.

The Shell

The Software Tools shell is a command interpreter that reads lines from the user terminal or a file and interprets them as requests to execute programs. The shell includes mechanisms to redirect the input and output of the tools to the user terminal,

files, or other programs. It also enables the user to group commands together to make up new commands. The ease of generating and executing complex user-tailored commands from simple ones distinguishes Unix and the Software Tools from other systems in which utilities are often clumsy. The text box "Software Tools Shell" describes the shell in greater detail.

The Library

The Software Tools library provides a framework for accessing system services by both the tools and user programs. The library includes basic system operations as well as groups of functions satisfying common programming needs. These include:

- Unix-type I/O (input/output) functions
- file and directory manipulation
- dynamic memory allocation
- string manipulation
- linked-list handling
- symbol-table creation
- text-pattern matching
- data-type conversion and manipulation
- date and time formatting
- command-line argument handling
- process control

Table 2 describes the library functions in detail.

Text continued on page 436

Text Manipulation			Managing Files and Directories		
Software Tool	Unix Utility	Description	Software Tool	Unix Utility	Description
e, edin	ed	editor	ls	ls	list files
sedit	sed	stream editor	cd	cd	change directory
ch	gres	change text patterns	pwd	pwd	print working directory name
tr	tr	transliterate characters	mv	mv	move/rename file
find	grep	locate text patterns	rm	rm	remove files
fb		find text patterns in blocks of lines	ar	ar	archive files
isam		build index sequential access list	n.a.	chown, chgrp	change owner/group of files
xref		cross reference of symbols	n.a.	chmod	change mode of file
field		manipulate fields of data	find	find	search for files
mcol	pr -n	produce multicolumn output	ln	ln	link files
sort	sort	sort lines	mkdir	mkdir	make a directory
lam		lamine lines of files together	rmdir	rmdir	remove a directory
uniq	uniq	strip duplicate lines	sum	sum	validate a file (checksum)
rev	rev	reverse order of characters	tar, tp	tar, tp	tape archiver
number		number lines	touch	touch	update last-change-date
detab		convert tabs to spaces	file	file	determine file type
entab		convert spaces to tabs	Document Preparation		
crypt	crypt	crypt and decrypt files	format	roff, nroff	text formatter
cpress		compress files		troff	text formatter for typesetter
expand		expand compressed files	form		form letter generator
os		convert backspaces for printing	spell	spell	spelling checker
	col	convert reverse line feeds for printing	lookup	look	look up words in dictionary
pl		print specific lines in file	kwic, unrot	ptx	generate permuted index
	awk	pattern scanning and processing language		deroff	remove nroff commands
	join	join lines with identical fields		eqn	generate equations for nroff
	prep	put words on single lines		tbl	generate tables for nroff
Manipulating Files				refer	find and insert literature references
cat	cat	concatenate/copy files		pubindex	make index for "refer"
crt		paginate files to terminal		tc	translate troff output for Tektronix 4015
cp	cp	copy files	Process Control		
pr	pr	paginate files for printing	sh	sh	command-line interpreter (shell)
show		show all characters (control too)	run		run a tool (without shell)
tail	tail	print last lines of files	which		print full pathname of command
tee	tee	copy input to output and named files	reset		reset system after media change
includ		include files within files	logout	logout	log out of shell
split	split	split up file	n.a.	at	run process at specific time
cmp	cmp	simple file compare	n.a.	login	log into system
diff	diff	differential file compare	n.a.	kill	kill (background) process
	diff3	three-way differential file compare	n.a.	nice	run process at low priority
comm	comm	print lines common to two files	n.a.	ps	process status
ll		print longest, shortest line lengths	n.a.	sleep	suspend termination for specified period
wc	wc	count words, characters, lines	n.a.	wait	wait for completion of a process
	dd	convert and copy a file		time	time a process
				prof	display profile data

Table 1: The Software Tools and their Unix equivalents.

Table 1 continued on page 432

Table 1 continued:

User Support/Information Retrieval			Software Tool Unix Utility Description		
dc		desk calculator		F77	FORTRAN compile
date		print/set time and date		struct	convert FORTRAN-66 to RATFOR
echo		print command-line arguments		lorder	find ordering relation for library
man		print manual entry		nm	print name list of object files
n.a.	passwd	set/change password		od	octal dump
n.a.	tty	get terminal name		size	print size of object file
n.a.	who	list users on system		strip	remove symbols and relocation bits
	true, false	commands which return true or false		ranlib	convert archives to random libraries
	basename	print basename of file			
	cal	print calendar			
	calendar	remind user of appointments			
	expr	evaluate arguments as an expression		Miscellaneous	
				graph	draw a graph
	factor	factor a number		plot	graphics filter
				spline	interpolate smooth curve
	test	condition command		tk	paginate for the Tektronix 4014
	units	quantity conversions	n.a.	write	send message to another user
			n.a.	mesg	permit or deny messages
Language Translation/Program Development			tcs	sccs	test maintenance system
macro	m4	macro processor	msg	mail	send/receive mail
ratfor	ratfor	RATFOR preprocessor		learn	computer-aided instruction about Unix
fsort		sort FORTRAN declarations		lpr	print spooler
rc	rc	RATFOR, FORTRAN, link, load		make	maintain program groups
fc	fc	FORTRAN, link, load		cu	call another Unix machine
ld	ld	load		uucp	Unix-to-Unix copy
tsort	tsort	topological sort		uux	Unix-to-Unix command execution
yacc	yacc	compiler-compiler			
lex	lex	lexical analyzer		stty	set terminal options
	adb	debugger		tabs	set terminal tabs
	as	assembler			
	bas	basic interpreter			
	bc	arbitrary-precision arithmetic language	Key:		
			n.a.:—not applicable to single user/single process systems like CP/M.		
	cc, pcc	C compile	The capabilities of a Software Tool and a Unix utility may not always be exactly the same.		
	lint	C syntax check			

Software Tools Shell (Carousel Microtool's CP/M Implementation)

The shell is a command-line interpreter; it reads lines from the terminal or a file and interprets them as requests to execute other programs.

Commands

In its simplest form, a command is the file name of a program to be run, followed by arguments given to the program. The

command name may specify any file in the system. CP/M enables a user number to be part of the command (file) name. The command may be a Software Tool or any other program. The shell searches for the named file in a series of directories specified by the user in an environment file. When the command is located, it is loaded into memory and executed. When the command

is finished, the shell resumes its own execution. For example, giving the command

```
sort file1 file2
```

causes the shell to locate and execute the command sort. Sort in turn merges and sorts the contents of the two named files and puts the output on the user's terminal.

I/O Redirection

Software Tools programs have three files automatically available to the user:

standard input
standard output
standard error output

All three are assigned to the user's terminal unless specifically redirected to disk files or other devices. Redirection is specified by preceding the desired device or file name with a special character:

```
<file  read standard input from
        "file"
>file  send standard output to
        "file"
?file  send standard error output
        to "file"
>>file append standard output to
        "file"
??file append standard error output
        to "file"
```

In the above example the sorted output could be saved on a file:

```
sort file1 file2 >sorted
```

or sent to the printer:

```
sort file1 file2 >/lst
```

(/lst is the tools form of the name for the printer).

I/O redirection is actually performed by each tool individually, rather than by the shell.

Pipes

A sequence of commands separated by vertical bars (|) causes the shell to execute each command in sequence and arranges to have the standard output of each command delivered as the standard input to the next command in the sequence. The sequence

```
sort list | uniq | crt
```

sorts the contents of file list. The sorted output passes to uniq, which removes extra copies of duplicated lines. This output then goes to crt, which paginates output for viewing on a terminal.

Command Separators

Commands need not be on different lines; instead they may be separated by semicolons:

```
ar -x program rtn ; e rtn
```

extracts the member rtn from the archive file program and then enters the editor.

Background Processes

Unix shells enable processes to be started and have control returned immediately to the shell. The new process continues running in the background, sharing resources with the shell process. This mechanism is impossible to implement on single-process systems such as those using CP/M. However, to simulate the mechanism in some reasonable way, the Carousel shell saves any commands indicated as background processes and executes them at the end of the session, when the user logs out of the shell. For example,

```
format doc >/lst &
```

formats the file doc and sends it to the printer at the end of the session (the ampersand indicates a background process).

Script Files

The real power of the Unix and Software Tools shells comes from the ability to generate new commands by combining existing commands. This feature is possible because the shell not only executes programs, but also treats script files (text files containing yet more commands) as commands. These scripts may participate in pipelines, have their I/O redirected, and appear in any context that a regular command may. Scripts may be nested by referencing scripts that may, in turn, reference other scripts.

Scripts are useful for creating new commands and for grouping commands together for multiple reexecution. For example, you could create a standard procedure by editing file fix to fill it with the following commands for the shell:

```
ar -x book chap1
e chap1
format chap1 | crt
ar -u book chap1
```

Then by typing fix the system would extract chap1 from the archived file book; edit chap1; send chap1 to the formatter and display it page by page on the terminal; and finally update it in the archive file book.

Arguments can also be passed to script files. Character sequences of the form \$n, where n is between 1 and 9, are replaced by the nth argument to the invocation of the script. If book has more than one sec-

tion, the script could be written:

```
ar -x book $1
e $1
format $1 | crt
ar -u book $1
```

Then you could type:

```
fix chap1
or fix chap7
or fix intro
```

to edit, view, and update the respective sections of book.

Script files can include inline explicit data that the tools can read as their standard input. The special input redirection notation << is used to achieve this effect. For example, the editor takes its commands from standard input, normally the terminal. However, within a shell script, commands may also be embedded this way:

```
e file <<!
(editing requests)
!
```

(The ! is arbitrary; any character can be used.) The lines between <<! and ! are called, in Unix terminology, a "here document"; they are read by the shell and made available to the command as its standard input.

Finally, as an indication of the power of script files, listing 1 shows an example of a script file to show changes that have been made to command files of dBASE II, a database-management program.

Environments

Like Unix, The Carousel shell maintains an environment file. This file contains information about the user's system and needs, such as the date, tab settings, and the directories in which to search for user programs or tools. The environment file is available to all tools and is modified by a few. In addition, users are free to adjust the information for their own needs.

Control Structures

Constructs of the nature:

```
if ... then ... else ...
while ... do ...
for ... in ... do ...
```

aid in reiteration and conditional execution within scripts. The Software Tools Users Group is currently standardizing the syntax for these shell control structures.

GOLD FINGERS

All apples taste different, so is our SEKON 64Z



Look at these features:

1. Dual CPU 6502 & Z80, 64K RAM.
2. Numeric Keypad, Upper & Lower Case.
3. Inverse Video, Auto Graphic Test.
4. Auto Repeat, High Speed Cursor.
5. 25 Function Keys, High Quality Case.
6. Build in 40-80 columns capability.
7. All Gold Sockets & Slots.
8. 5A Power Supply, 110V-220V.
9. Compatible with Apple II plus & Iie.
10. Uses all CP/M software.

Made in USA

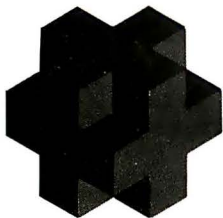
FEATURES	APPLE Iie	SEKON 64Z
6502 CPU	yes	yes
Z80 CPU	no	yes
64K	yes	yes
40-80 column	yes	yes
Numeric Keypad	no	yes
Function Keys	no	yes
Gold Sockets	no	yes
High Speed Cursor	no	yes
Warranty	3 months	1 year
Suggested Retail:	\$1,395	\$795

SEKON introduces another Second-to none product, the SEKON-64Z dual processor computer. It is the world's first mass producing computer, which has all GOLD sockets and slots, to ensure high performance and reliability. In addition to a one full year warranty, we have also a two weeks money back guarantee for mail order customers. Call your dealer or us for our special introductory offer.



"SEKON, Second to None!"

SEKON INT'L. CORP. OF AMERICA (800) 423-4302, in CA (213) 936-1577
2210 Wilshire Blvd., Suite 577, Santa Monica, CA 90403 Telex 181121



Petro-Lewis Yield Plus Fund II

A unique series of limited partnerships formed to purchase oil and gas production payments from a specified portfolio of producing

properties plus engage in development drilling activities on those properties.

\$450 Million Offering

Production Payments for Quarterly Distributions

Plus Long-Term Economic Life

PETRO-LEWIS

Minimum Subscription—\$2,500 in most states (\$2,000 for Qualified IRA Plans)

This announcement is neither an offer to sell nor a solicitation of an offer to buy any of these securities. The offering is made only by the Prospectus, copies of which may be obtained in any State only from dealers or brokers who may lawfully offer these securities in such State.

Please see that I receive the latest information on this offering, including a current prospectus.

Mail to: Kay Grover
Petro-Lewis Securities Corporation
P.O. Box 2250, Dept. MH-2, Denver, CO 80201

Name

Telephone

Firm

Address

City

State

Zip

Listing 1: The alterations to dBASE II command files.

```
# Shell command file to show work done to
# dBASE II command files.
# usage: dbdiff dir
# (where dir is a backup directory)
# "dir" should be specified in tools form,
# e.g. "/2/B"
# dbdiff will print all new dBASE command
# files and will print existing dBASE
# command files with any changes
# marked with a "|" in the right margin.
```

```
# Collect names of .cmd files in both
# directories.
```

```
ls .cmd >1.tmp
```

```
ls $1 .cmd >2.tmp
```

```
# Find and print new dBASE commands.
```

```
# Here comm reports lines in 1.tmp
# which are not present in 2.tmp;
# field changes that report into a series
# of print commands;
# and sh then executes those print
# commands.
```

```
# The "@" signs suppress the following
# newline, effectively continuing the
# shell command across several lines.
```

```
comm -1 1.tmp 2.tmp | @
field "pr >/lst $1" | @
sh
```

```
# Find existing dBASE commands and show
# changes.
```

```
# Here comm reports files listed in both
# 1.tmp and 2.tmp;
# e (the editor) changes each file name
# reported by comm into a series of
# commands to:
# print the file name;
# print the current date & time;
# print the differences between the
# versions in this directory
# and in the other directory;
# and cat puts a few formatter commands
# into 4.tmp, to be called upon
# by each line of 3.tmp.
```

```
comm -3 1.tmp 2.tmp >3.tmp
```

```
e 3.tmp <<!
```

```
1,$s~?*~echo & >/lst ; date >/lst ;
diff -r $1/& & | format 4.tmp - >/lst -
```

```
w
```

```
q
```

```
!
```

```
cat >4.tmp <<!
```

```
.nf
```

```
.in 5
```

```
.rm 70
```

```
!
```

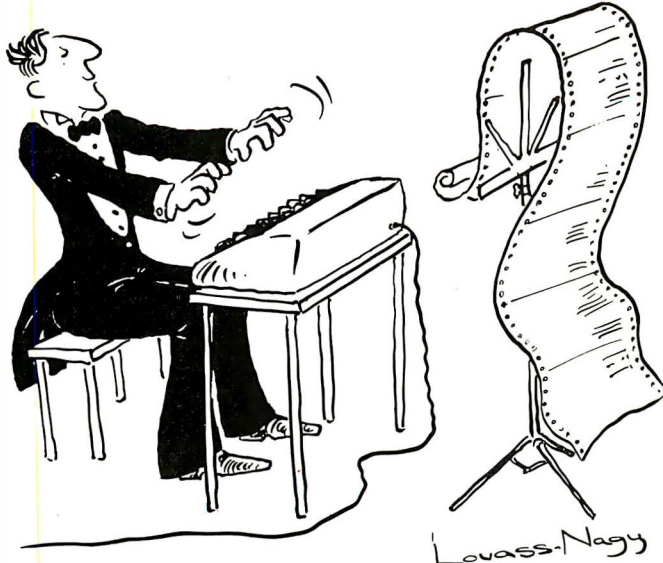
```
# Finally the shell runs the commands
```

```
# that e just prepared and
```

```
# rm removes all three scratch files.
```

```
sh 3.tmp $1
```

```
rm 1.tmp 2.tmp 3.tmp
```

•NEW PRODUCTS•

Before Johann Sebastian Bach developed a new method of tuning, you had to change instruments practically every time you wanted to change keys. Very difficult.

Before Avocet introduced its family of cross-assemblers, developing micro-processor software was much the same. You needed a separate development system for practically every type of processor. Very difficult and very expensive.

But with Avocet's cross-assemblers, a single computer can develop software for virtually any microprocessor! Does that put us in a league with Bach? You decide.

The Well-Tempered Cross-Assembler

Development Tools That Work

Avocet cross-assemblers are fast, reliable and user-proven in over 3 years of actual use. Ask NASA, IBM, XEROX or the hundreds of other organizations that use them. Every time you see a new microprocessor-based product, there's a good chance it was developed with Avocet cross-assemblers.

Avocet cross-assemblers are easy to use. They run on any computer with CP/M* and process assembly language for the most popular microprocessor families.

5¼" disk formats available at no extra cost include Osborne, Xerox, H-P, IBM PC, Kaypro, North Star, Zenith, Televideo, Otrona, DEC.

Turn Your Computer Into A Complete Development System

Of course, there's more. Avocet has the tools you need from start to finish to enter, assemble and test your software and finally cast it in EPROM:

Text Editor VEDIT -- full-screen text editor by CompuView. Makes source code entry a snap. Full-screen text editing, plus TECO-like macro facility for repetitive tasks. Pre-configured for over 40 terminals and personal computers as well as in user-configurable form.

CP/M-80 version \$150
CP/M-86 or MDOS version \$195
(when ordered with any Avocet product)

EPROM Programmer -- Model 7128 EPROM Programmer by Gtek programs most EPROMS without the need for personality modules. Self-contained power supply ... accepts ASCII commands and data from any computer through RS 232 serial interface. Cross-assembler hex object files can be down-loaded directly. Commands include verify and read, as well as partial programming.

PROM types supported: 2508, 2758, 2516, 2716, 2532, 2732, 2732A, 27C32, MCM8766, 2564, 2764, 27C64, 27128, 8748, 8741, 8749, 8742, 8751, 8755, plus Seeq and Xicor EEPROMS.

Avocet Cross-assembler	Target Microprocessor	CP/M-80 Version	•CP/M-86 IBM PC, MSDOS•• Versions•
•XASMZ80	Z-80	\$200.00 each	\$250.00 each
•XASM85	8085		
XASM05	6805		
XASM09	6809		
XASM18	1802		
XASM48	8048/8041		
XASM51	8051		
XASM65	6502		
XASM68	6800/01		
XASMZ8	Z8		
XASMF8	F8/3870		
XASM400	COP400		
XASM75	NEC 7500	\$500.00	
Coming soon: XASM68K...68000			

(Upgrade kits will be available for new PROM types as they are introduced.)

Programmer \$389

Options include:

- Software Driver Package -- enhanced features, no installation required.
- CP/M-80 Version \$ 75
- IBM PC Version \$ 95
- RS 232 Cable \$ 30
- 8748 family socket adaptor ... \$ 98
- 8751 family socket adaptor ... \$174
- 8755 family socket adaptor ... \$135

- **G7228 Programmer by Gtek** -- baud to 2400 ... superfast, adaptive program-ming algorithms ... programs 2764 in one minute.

• Programmer \$499

- Ask us about Gang and PAL programmers.

- **HEXTRAN Universal HEX File Converter** -- Converts to and from Intel, Motorola, MOS Technology, Mostek, RCA, Fairchild, Tektronix, Texas Instruments and Binary formats.

• Converter, each version \$250

Call Us

If you're thinking about development systems, call us for some straight talk. If we don't have what you need, we'll help you find out who does. If you like, we'll even talk about Bach.

CALL TOLL FREE 1-800-448-8500

(In the U.S. except Alaska and Hawaii)

VISA and Mastercard accepted. All popular disc formats now available -- please specify. Prices do not include shipping and handling -- call for exact quotes. OEM INQUIRIES INVITED.

*Trademark of Digital Research

**Trademark of Microsoft



AVOCET SYSTEMS INC.™

DEPT. 1183-B
804 SOUTH STATE STREET
DOVER, DELAWARE 19901
302-734-0151 TELEX 467210

Symbol Definitions (ratdef)

definitions.....standard RATFOR definitions

File Manipulation

*amove.....move (rename) a file
*close.....close (detach) a file
*create.....create a new file (or overwrite an existing one)
*gettyp.....get type of file (character or binary)
*isatty.....determine if a file is a terminal
*mkuniq.....generate unique file name
*open.....open an existing file for reading, writing, or both
*remove.....remove a file from the file system

I/O

fcopy.....copy one file to another
*flush.....flush output buffer for file
getc.....read character from standard input
*getch.....read character from file
*getlin.....read next line from file
*note.....determine current file position
*prompt.....prompt user for input
putc.....write character to standard output
*putch.....write character to file
putdec.....write integer in field
putint.....write integer in field on file
*putlin.....output a line onto file
putstr.....write string in field on file
*readf.....binary read from a file
*remark.....print single-line message
*seek.....move read/write pointer
*setmod.....set character device mode
*writef.....binary write to a file

Process Control

*endst.....close all open files and terminate program execution
*exec.....execute task
*initst.....initialize all standard files and common variables

Directory Manipulation

*closdr.....close directory
*cwwdr.....change working directory
*gdraux.....get auxiliary directory information
*gdrprm.....get next directory entry
*gwwdr.....get name of current working directory
*opendr.....open directory for reading

String Manipulation

addset.....add character to array if it fits, increment pointer
addstr.....add string to array if it fits, increment pointer
concat.....concatenate two strings together
ctoc.....copy string-to-string
equal.....compare str1 to str2
gettok.....parse tokens
getwrd.....get nonblank word from array, increment pointer
index.....find character in string
length.....compute length of string
scopy.....copy string from one array to another
sdrop.....drop characters from a string
skipbl.....skip blanks and tabs in array
sktok.....skip over tokens
sstr.....slice (take) a substring from a string
stake.....take characters from a string
scopy.....copy string, increment pointer
stncmp.....compare first n characters of strings
stncpy.....copy n characters from one array to another
strcmp.....compare two strings

Table 2: *The functions of the Software Tools library.*

Text continued from page 430:

The Tools or Unix?

Although the Software Tools provide many of the features of Unix, they are not an exact copy of Unix. They exist alongside the local operating system and provide many of the desirable aspects of Unix in situations where using Unix is impossible or inappropriate. For instance, if you don't want to pay Unix's high price, if you want to use software packages that aren't available in Unix versions, or if a Unix implementation is not available for your hardware, the Software Tools can provide the power and elegance of the Unix interface.

Let's look at the Software tools movement and considerations that have made the tools successful.

The Software Tools Movement

In 1976 Kernighan and Plauger wrote *Software Tools* (see reference 3). Their goal was to teach good programming style based on their experiences with Unix at Bell Laboratories.

They used pared-down versions of Unix utilities rewritten in RATFOR (Rational FORTRAN), a C-like preprocessor language (see text box, "What Is RATFOR?"). The programs and the RATFOR preprocessor were made available on magnetic tape. The book and tape were the seeds from which the tools movement developed. The movement arose independently at several major research laboratories and universities.

The tools were of immediate interest to researchers and users, and the programs were implemented on numerous computers. As users began to experiment with and enhance the programs, they began to realize that the tools offered more than a useful set of utility programs. Researchers, primarily at Lawrence Berkeley Laboratory (LBL), expanded the original package to include a powerful subroutine library, a Unix-like shell, and many more of the Unix utilities. By providing all three levels

(shell, utilities, and library) the tools now offered a portable, uniform interface with the functionality of Unix. The package was implemented on the diverse assortment of LBL machines and on many machines to which the researchers had network access. The result was Unix functionality on non-Unix systems and a consistent user interface across many different systems (see reference 1).

One reason the Software Tools have been so widely accepted is their portability. The tools can be implemented on virtually any machine. This portability was achieved by using a programming language that was available on all machines and by isolating system dependencies into "primitive" function calls that must be implemented separately for each different system.

With certain data-type manipulation conventions and other programming details, this portability has enabled the package to be imple-

strim.....trim trailing blanks and tabs from a string
 type.....determine type of character

Character Conversion

clower.....convert character to lower case
 ctoi.....convert string to integer, increment pointer
 ctmn.....translate ASCII control character to mnemonic
 cupper.....convert character to upper case
 esc.....check for escaped character
 fold.....convert string to lower case
 gctoi.....generalized character-to-integer conversion
 gitoc.....generalized integer-to-character conversion
 itoc.....convert integer to character string
 lower.....convert string to lower case
 mntoc.....convert ASCII mnemonic to character
 upper.....convert string to upper case

Pattern Matching

amatch.....look for pattern matching regular expression
 getpat.....encode regular expression for pattern matching
 makpat.....encode regular expression for pattern matching
 match.....match pattern anywhere on line

Command Line Handling

*delarg.....delete a command-line argument
 *getarg.....get command-line arguments
 gfnarg.....get next filename argument
 query.....print command usage information

Dynamic Storage Allocation

*dsfree.....free a block of dynamic storage
 *dsget.....obtain a block of dynamic storage
 *dsinit.....initialize dynamic storage

Symbol Table Manipulation

delete.....remove a symbol from symbol table
 enter.....place symbol in symbol table
 lookup.....get string associated with symbol from hash table
 mktabl.....make a symbol table
 rmtabl.....remove a symbol table
 sctabl.....scan all symbols in a symbol table

Linked List / Stack Handling

maklst.....create and initialize linked list
 frelst.....remove a linked list and free allocated memory
 push.....push an item onto the top of the list/stack
 pop.....pop an item from the top of the list/stack
 inject.....inject a new item into a linked list
 xtract.....read an item from a linked list
 prvnod.....get previous node pointer
 nxtnod.....get next node pointer
 remod.....remove a node from a linked list

Date Manipulation

atodat.....convert ASCII characters to integer date
 fmdat.....convert date to character string
 *getnow.....get current date and time
 wkday.....get day-of-week corresponding to month-day-year

Error Handling

cant.....print "name: can't open" and terminate execution
 error.....print single-line message and terminate execution

* indicates that the routine is system dependent and has been implemented by Carousel Microtools for CP/M and MS-DOS.

mented on more than 50 operating systems. Table 3 provides a partial list of manufacturers offering computers on which the tools have been implemented.

Which Language Is Best?

Computer languages are judged on their ability to solve specific problems; therefore, the best language for the Software Tools package was the one that could most adequately fill the following requirements:

- availability—the language had to be available on almost every machine
- suitability—the language had to be appropriate for textual (as opposed to numerical) applications; it had to be powerful enough to handle the support libraries that provide the necessary file access, I/O, process control, and other system-support services
- quality—the language had to be high-level, easy to read and understand, easy to learn, and powerful

enough to solve applications problems

FORTRAN filled the first requirement, fell down a bit on the second, and provided little of the third. C met the second and third requirements but was not usually available on both microcomputers and larger machines. Pascal met the third requirement but was no more commonly available than C and was not appropriate to the support of large libraries and moderately complex bodies of code (see reference 2). Several other state-of-the-art languages were appealing but not generally available. Thus, no single language met all the requirements, and a compromise was necessary. The RATFOR language preprocessor was chosen because it provided the control structures, readability, and elegance of C and was translatable into FORTRAN (the language available on most systems). A C-like support library was developed

to supplant FORTRAN's incomplete textual, file manipulation, and I/O capabilities. Even though FORTRAN is used at the RATFOR base level, the user is insulated from FORTRAN just as the user of any high-level language is insulated from the machine language.

The choice of language was not critical to the approach. In fact, for the person using the tools the implementation language is unimportant. Only the tools implementer and people developing new tools with the library ever need to use the language. Had the tools been designed solely for the microcomputer environment, C might have been a more appropriate choice. With the computer industry rapidly developing new machines and more elegant languages, the Software Tools community is now reevaluating the original choice of language and considering mechanisms for making the tools available in other languages as well.

What is RATFOR?

RATFOR (Rational FORTRAN) is the implementation language for the Software Tools. It is closely patterned after C in its control structures, but it is compiled into FORTRAN by the RATFOR preprocessor. The availability of FORTRAN allows RATFOR to be easily installed on a wide variety of systems. In addition to being a portable language suitable for implementing the Software Tools, RATFOR is a convenient language for program development. The control constructs of RATFOR are those of C, and the data structures are those of FORTRAN.

RATFOR's nature can most easily be described with examples of some actual code. A file of standard definitions is automatically processed by the RATFOR compiler to define new symbolic constants. A section of this file is:

```
define (EOF, -1)
define (EOS, 0)
define (MAXLINE, 128)
define (STDIN, 1)
define (STDOUT, 2)
define (character, integer)
```

Using these definitions, the following code is an example of a program in RATFOR that finds the length of the longest line read from standard input:

```
DRIVER
character line(MAXLINE)
integer getlin, length, len, size
size = 0
while (getlin(line, STDIN) != EOF)
{
    len = length (line)
    if (len > size)
        size = len
}
call putint (size, 5, STDOUT)
call putch (NEWLINE, STDOUT)
DRETURN
end
```

The macros DRIVER and DRETURN are also defined in the standard definition file and are used to start and end all RATFOR programs.

The following code is the same program written in C:

```
#include <stdio.h>
#define(MAXLINE,128)

main()
{
    char line[MAXLINE];
    int fgets(), strlen(), size=0, len;
    while (fgets(line, MAXLINE, stdin))
    {
        len = strlen(line);
        if (len > size)
            size = len;
    }
```

```
}
fprintf (stdout, "%5d \n", size);
}
```

The similarity between the RATFOR and C versions is obvious. Notice that the RATFOR example consists almost entirely of standard FORTRAN statements, especially assignment statements and subroutine calls. The RATFOR compiler passes these statements through to the FORTRAN version almost unchanged. What RATFOR adds to FORTRAN are file inclusion, token substitution, macros for text replacement, and the following control constructs:

*if-else for conditional execution,
while, for, and repeat-until for
looping,
break and next for controlling loop
exits,
switch-case-default for selection of
alternatives,
braces ({}) for statement grouping.*

RATFOR's syntax was intended to liberalize FORTRAN's syntax restrictions as much as possible. As a result, RATFOR source code is naturally concise and reasonably pleasing to the eye. RATFOR features are as follows:

- free-form page layout
- unobtrusive comments

Primitives Isolate Machine Dependencies

In the Software Tools package, system dependencies are isolated in the primitives, a set of routines that make up the tools' interface to the operating system. The primitives provide standardized system services such as file manipulation, I/O, process control, and dynamic memory allocation. The tools and their subroutines access system services through these primitives. Tool source code can be moved from system to system without change. When the tools package is moved to a new system, only the primitives must be changed or rewritten.

The original implementers of the tools issued two prime directives to assure compatibility among a wide variety of operating systems. First,

they decided to use the file types of the operating system. Internal file formats specific to the machine are hidden from the user by the primitive functions, allowing both local utilities and Software Tools programs to read and write the same files and providing a standardized way to access files on all systems. Second, changes to the local system, or interference with it to implement the package, are discouraged. Such changes, combined with the local system's idiosyncrasies, would make the package unstable in new system releases.

The primitives address the issue of machine efficiency; they minimize the demands of the software upon scarce system resources like memory or central processor time. For example, the utilities of the Software Tools package are oriented toward text pro-

cessing and program development (writing source code, documentation, data preparation, etc.). These utilities are characteristically limited by I/O rates. Because the I/O capabilities are isolated in the primitives, the effect of this problem can be reduced through efficient implementation of the I/O primitives. Because all utilities access resources through the primitives, they automatically benefit from such optimization.

The Software Tools Users Group

The need for cooperation among implementers and users of the tools led to the formation of the Software Tools Users Group at Menlo Park, California. It originated at the Lawrence Berkeley Laboratory and was initially funded by the Depart-

- use of <, <=, >, >=, ==, !=, etc. for comparison expressions
- string data type
- quoted character strings and character constants
- define statement for symbolic constants
- include statement for source-file inclusion
- macro preprocessor for textual manipulation

RATFOR code is often easier to read and understand than the corresponding section of code as normally written in C. For example, the two following fragments of code each copy a string from one buffer to another:

RATFOR version

```
for (i=1; from(i) != EOS; i=i+1)
  to(i) = from(i)
to(i) = EOS
```

/* C version */

```
char *t=to, *f=from;
while (*t++ = *f++);
```

One could argue that a good C compiler sometimes produces faster code, but in large programs the readability of the RATFOR style is often an advantage over the more terse C style.

ACOS
Amdahl
Apollo
AN/UYK
Burroughs
CDC
Cray
Data General
DEC
FACOM
GEC
HP
HITAC
Honeywell
IBM
Intel
Interdata
Modcomp
Multics
NCR
Perkin-Elmer
Prime
Rolm
SEL
Tandem
Univac
Wang
Xerox
Machines running CP/M
Machines running MS-DOS
Machines running Unix

Table 3: A partial list of manufacturers on whose machines the Software Tools package has been implemented to varying degrees of sophistication.

ment of Energy. Since its inception in 1978, the group has become an international body performing the following functions:

- establishing and publishing standards for the primitives and tools and supporting an ongoing standards committee
- collecting and distributing information on current developments to avoid duplication of effort
- collecting and evaluating new utilities, extensions, and variants
- holding semiannual meetings in conjunction with the Usenix Unix users group
- publishing a newsletter and software catalog
- distributing tapes containing collections of utilities from different organizations

Much of the tools' source code is now in the public domain and freely distributed. The primitives, however, are generally developed, licensed, and maintained by vendors.

The standardization procedure used by the tools group is unusual. New utilities are collected and distributed early in their development phase, allowing users to experiment with new ideas and reject those that prove unportable or functionally undesirable. Code sharing also allows users and developers to glean ideas from new offerings and incorporate them into their own developments. As ideas are distilled and utilities enhanced or extended, the utilities are redistributed, and those receiving popular support are eventually returned to the tools group. There they pass to the Implementers Committee,

MAINFRAME SOFTWARE FOR MICROS

Assembler programs that bring a set of functions available in IBM mainframes to small computers.

MenuWiz \$99

MENU DRIVER WITH PASSWORD CONTROL

Create easily menus to drive all your applications and optionally restrict access to them selectively with password protection.

No programming skills are required. Just write a few simple specifications to design a menu and tell MenuWiz for each menu option.

- Name of the program or of another menu to load (IBM DOS 2.0 users can also execute operating system commands)
- What level of security you want (if zero no security clearance is required).

To access a protected menu option the user has to have entered a password with a security clearance equal or greater than the security level for that option.

A password maintenance program included, let the master security officer maintain passwords, set its security level (from zero to 99), and define a password scrambling code.

— **MenuWiz Logger to record \$49**
accesses on a history file

ScreenWiz \$99

FULL SCREEN CONTROL

Define screen format specifications outside your programs and save them as a disk file. At execution time, call ScreenWiz to retrieve the screen formats and do all the I/O for you.

- One command input/displays a full screen
- Use all of your CRT video attributes
- Define up to 24 function keys
- Validate input while keying. Insert and delete characters to correct fields.
- Tab fields forward and backward.
- Draw figures using vertical field displays.
- Save memory using format overlays and virtual constants.
- Build help screens. Call them from any field with a function key then return to where you were.
- Cut your program code by as much as 50%.

— **ScreenWiz Tester to display \$49**
screen samples

ReportWiz \$99

FULL PRINTER CONTROL WITH DISK BACKUP

Define report format specifications outside your programs and save them as a disk file. At execution time, call ReportWiz to print the report for you.

- Handles automatically printing of heading, detail and total lines, on page overflow.
- Edits numeric fields (commas, zero suppress, date, sign, CR/DB, etc.).
- Automatic print of report page number.
- Uses ALL fancy printer controls your printer supports. For example you can print headings in double width, totals underlined, messages in italic, etc.
- Optionally redirects printer output to disk (like a spooler) to avoid printer jam problems. A print program included lets you print this disk file selecting the page range and the no. of copies.
- Cut your program code by as much as 50%.

— **ReportWiz Tester to print \$49**
report samples

Available for CP/M-80 and IBM DOS operating systems. ScreenWiz and ReportWiz can be called from most popular Basic, Cobol, Pascal, Fortran, Assembler and C programs. All prices are in U.S. dollars and include a comprehensive Reference Manual, extensive examples and tutorial demonstration programs. Call for CP/M-86 availability.

Shipping costs per program (air parcel)

U.S. and Canada \$ 6

Overseas \$18

INTERDATA SYSTEMS INC.

15 Toronto St. Suite 700 Visa, M/C Am. Exp.
Toronto, Ontario Canada M5C 2E7 C.O.D.

CALL COLLECT TO ORDER

(416) 366-2136

which makes final decisions on acceptance and standardization. Thus, standards are always based on ideas or utilities tested and proven by the community rather than on newly designed products or untested ideas.

The sharing of code and feedback from users enables developers of new tools to build on each other's work, creating an environment in which new ideas can be quickly and thoroughly tested. The sharing results in natural selection of useful tools that have been tried and accepted by a large number of users with varying needs on many different systems.

The Present and the Future

Development of the Software Tools package is proceeding on two fronts: the basic package is being implemented on new systems, and user interfaces are being extended. The original package provided an environment for effective development of programs and manipulation of textual data and materials. However, the tools approach is applicable to most software

projects, including those involving networks, database management, graphics, and word processing. Among the portable packages being developed are experimental shells, statistical analysis systems, electronic-mail systems, screen editors, data-management packages, data-analysis packages, and source-code-maintenance systems. The tools group is actively evaluating suggested enhancements and extending the primitive set to provide as dynamic and creative an environment as possible.

Some hardware manufacturers avoid the Software Tools package because easy portability to a competitor's hardware is obviously bad for business. Increasingly, however, independent companies are marketing specific system implementations of the tools. These firms typically implement the primitives and provide maintenance and upgrade support. The high-level source code (utilities and portable sections of the library) is left unlicensed, so the Software Tools Users Group handles variations, extensions, and standards (a

compromise between the need for vendor support and the desire for user control).

The Software Tools package is already running on most minicomputer and mainframe systems, and extensions into the microcomputer world have begun.

Implementing the Tools

Writing programs in a language that is available on many systems is insufficient; you must also define an interface layer that isolates an application program from the details of any particular system. The primitives form the tools' interface layer and are the key to their success. They are the only allowed connection between the tools and the underlying operating system. Porting, or adapting, the tools to a new operating system involves writing the code for the primitives for that new system.

The primitives are more than just a collection of subroutines; they provide a complete environment for the tools. In a sense, they coordinate the "world view" of the tools with the

COMPUTER DEALERS

THIS AD WRITTEN FOR YOU ...

BYTE INDUSTRIES WANTS TO BE YOUR COMPUTER CONNECTION

Byte Industries is your one STOP shopping place for over 50 lines of microcomputers, terminals, printers and various other computer peripherals and accessories. Byte won't leave you at a STOP waiting for equipment to complete a sale. If you call in your order by 1:00 PST, we'll ship it out THE SAME DAY!

Byte Industries is searching for qualified Computer Dealers in the business of reselling the product we stock. If you're such a Dealer, you may have just found the best wholesale "Computer Connection" available.

BYTE INDUSTRIES IS THE AUTHORIZED COMPUPRO DISTRIBUTOR

I am in the business of reselling computers and related products. I would like a sales representative to contact me and tell me how I can receive my free Wholesale Pricelist and open a Dealer Account.

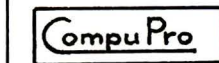
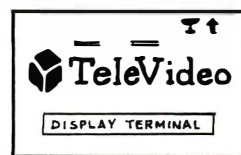
Name: _____

Store Name: _____

Address: _____

City: _____ State: _____ Zip: _____

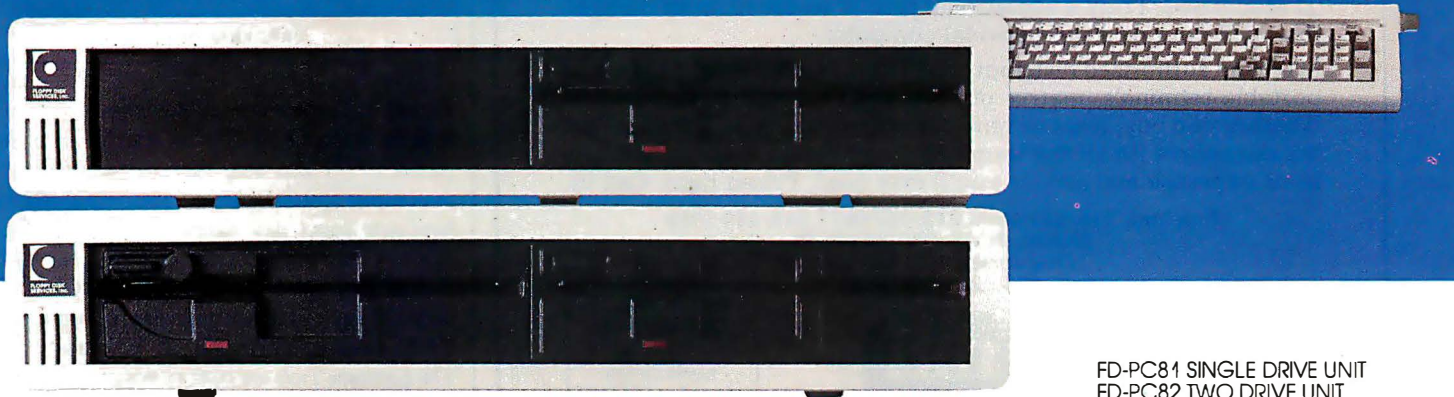
Return to: Byte Industries, Inc. (415) 783-8272
21130 Cabot Blvd., Hayward, CA 94545



NEC Information Systems



Plug into 2.5 Mb of additional online storage.



FD-PC81 SINGLE DRIVE UNIT
FD-PC82 TWO DRIVE UNIT

The FD-PC8 is a new 8" double sided disk drive sub-system that adds an impressive 2.5 megabytes of online storage capacity to your IBM PC. Coupled with the 640 kilobyte capacity of the two 5¼" internal drives in your IBM PC, the system gives you enormous data base capability from the moment you plug it in!

IBM PC styling.

The FD-PC8 looks like your other IBM PC components. Same styling. Same color. Same dimensions. It stacks neatly under, on top of, or next to your IBM PC, and better still, is only one-half the height of standard 8" drives.

IBM PC and mainframe compatible.

Total interface capability with both IBM mainframes and PC affords new and unique data base flexibility and computing opportunities to the serious PC user.

Long life design features.

The FD-PC8 utilizes Shugart drives; the standard for quality and reliability in the computer industry. Unlike most disk drives, the motor runs only when selected, greatly increasing the life expectancy of the FD-PC8.



FLOPPY DISK
SERVICES, INC.
741 Alexander Road
Princeton, NJ 08540

Circle 190 on inquiry card.

IBM PC is a registered trademark of International Business Machines Corporation.
Maynard Controller is a registered trademark of Maynard Electronics.
JFormat is a registered trademark of Tall Tree Systems.

SPECIFICATIONS

- Full one year warranty on materials and workmanship.
- Two fully assembled and tested Shugart double-sided 8" drives also available in one drive configuration.
- Direct drive DC motor.
- Exactly ½ the height of standard 8" drives.
- IBM PC styled and painted cabinet.
- All cabling included.
- IBM 3740 format compatible.
- No-mar rubber feet.
- Sturdy construction easily supports PC or monitor.
- Power supply designed for long life, trouble-free operation.

PRICING

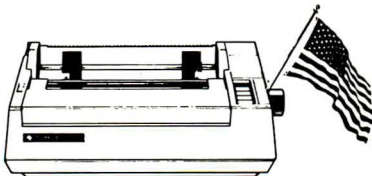
FD-PC8... dual system/\$1495.00... single system/\$1095.00
*Maynard Controller/\$245.00...JFormat/\$50.00

*Use of the B" system requires the Maynard B" controller and Tall Tree Systems® JFormat™ software, allowing the use of two internal 5¼" drives and the two outboard 8" drives.

Plug into the perfect IBM PC add-on!

For more information about the FD-PC*, call (609) 799-4440.
Or call our toll free order line at (800) 223-0306.
Dealer Inquiries Invited.

Texas Instruments Model 850...
the new American-made printer
that beats the imports on price,
performance and compatibility.



Buy, lease or rent the new TI 850 desktop printer from MTI.

150 cps, 9x9 dot matrix, both bit image and mosaic graphics with better resolution, easier font-changing, both parallel and RS232 interfaces standard. These are just a few of the features of the new TI Model 850. Truly a sensational printer designed to be compatible with your desktop personal computer.

MTI is an authorized distributor of Texas Instruments' full line of portable terminals, matrix printers and Professional Computers. Whether you buy, lease or rent our equipment, you'll find MTI is the one source for all the terminals, peripherals, systems, applications expertise and service you'll ever need. Priced right. Call us.

New York: 516/621-6200, 212/767-0677, 518/449-5959
Outside N.Y.S.: 800/645-6530
New Jersey: 201/227-5552
Ohio: 216/464-6688

"QED" Discounts
VISA & MasterCard



AUTHORIZED
DISTRIBUTOR
TEXAS
INSTRUMENTS
Data Systems Group

Applications Specialists & Distributors, New York, New Jersey and Ohio.
DEC, Intel, Lear Siegler, Texas Instruments, Dataproducts, Diablo, 3Com,
Hazeltine, Racal-Vadic, Digital Engineering, MICOM, Cipher, U.S. Design,
Protocol Computers, MicroPro, Microsoft, Polygon and Select.

HIERARCHY OF PRIMITIVES

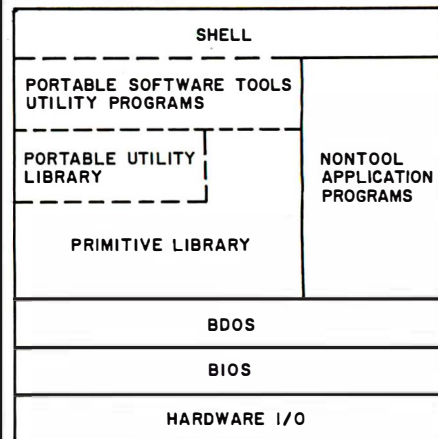


Figure 1: The hierarchical dependence of interfaces in the CP/M-80 version of the tools. At the CP/M level only the BIOS (basic input/output system) knows how to do direct hardware input and output, and only the BDOS (basic disk operating system) knows how to talk to the BIOS. These clean divisions were the key to the early success in moving CP/M to many different types of hardware. The Software Tools are built in isolated layers in the same way. Note that only the primitive functions know how to talk to the BDOS. The primitives are the communication channel between the portable tools and a specific operating system, such as CP/M or MS-DOS. The tools themselves can use the primitives or the library of utility routines that are also part of the tools package. The clean boundaries between the various interface layers in a system such as this are very important for maintaining clean portable programs. Any time these separations are violated, the resulting program may prove expensive to maintain and difficult to move to new machines.

TOTAL CONTROL: PC/FORTH™

GRAPHICS • GAMES • COMMUNICATIONS • ROBOTICS
DATA ACQUISITION • PROCESS CONTROL

- **PC/FORTH™**: interactive and conversational, but 20 times faster than BASIC.
- **PC/FORTH™** programs: highly structured, modular, easy to maintain.
- **PC/FORTH™**: direct control over all interrupts, memory locations, and i/o ports.
- **PC/FORTH™**: full access to DOS files and functions.
- **PC/FORTH™** application programs can be compiled into turnkey COM files and distributed with no license fee.
- **PC/FORTH™** Cross Compilers available for ROM'ed or disk based applications on most microprocessors.
- **PC/FORTH™**: compatible with PC/XT, Eagle, COMPAQ, and all hard disks.
- **PC/FORTH™**: for PC-DOS 1.1 and 2.0, CP/M-86®, Concurrent CP/M.

PC/FORTH™ \$100.00

Includes interpreter/compiler with virtual memory and background multi-tasking, full screen editor, assembler, many demonstration programs, and 170-page manual. Complies with 83-Standard. Upgrade to PC/FORTH+ available.

PC/FORTH+™ \$250.00

Allows creation of FORTH programs up to 1 megabyte in size.

DEMO DISK \$5.00

Requires graphics card.

Extension Packages

Forth Cross Compiler	\$300.00
Advanced Color Graphics	\$100.00
Intel 8087 Support	\$100.00
Software Floating Point	\$100.00
Interactive Symbolic Debugger	\$100.00
PC/GEN™ Custom Character Sets	\$ 50.00
PC/TERM for Smartmodem	\$ 60.00
QTF+ Editor/Text Formatter	\$100.00
Curry FORTH Programming Aids	\$150.00
Cross Reference Utility	\$ 25.00
Hierarchical File Manager	\$ 50.00
B+ Tree Index Manager	\$125.00
B+ Tree File and Index Manager	\$200.00
Intel 8087 Processor	\$250.00
"Starting FORTH" Tutorial	\$ 16.00

Trademarks: IBM, International Business Machines Corp.; CP/M, Digital Research Inc.; PC/FORTH+ and PC/GEN, Laboratory Microsystems Inc.



Laboratory Microsystems Incorporated
4147 Beethoven Street, Los Angeles, CA 90066
Phone credit card orders to (213) 306-7412



world view of the host operating system. The task is simple if the tools and the new system have similar views of the programmer's environment; the task is difficult if the new system has a different view. For example, it took less than a week to write and test the tools' primitives for Unix because Unix's view of the programmer's environment is similar to that of the tools. But implementing the tools' primitives on CP/M and MS-DOS (which are based on very different views) took more than a year.

When implementing the primitives, it is essential to keep in mind the two prime directives: maintain correspondence of file types and

There's a side to the Wang PC that IBM would like you to ignore.

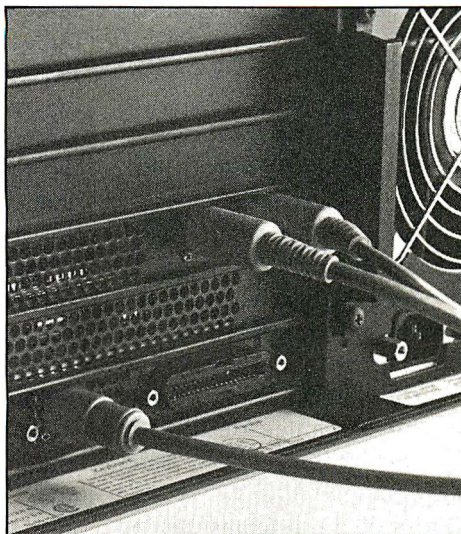
Maybe what puts the Wang Professional Computer ahead of the IBM PC is what's behind it.

Because at the back of the Wang PC is a series of system interfaces that can tie you into total office automation now. Integral links that let you access not only other Wang equipment, but even most popular databases and mainframes, including IBM.

The Wang Professional Computer also offers a variety of telecommunications options that can access everything from the Dow Jones News/Retrieval® service to electronic mail. And by simply adding word processing software, the Wang PC turns into the easiest-to-use word processor on the market.

Face to face, the Wang PC is superior to the IBM PC in speed, compatibility and ease-of-use.

Back to back, there's no comparison.

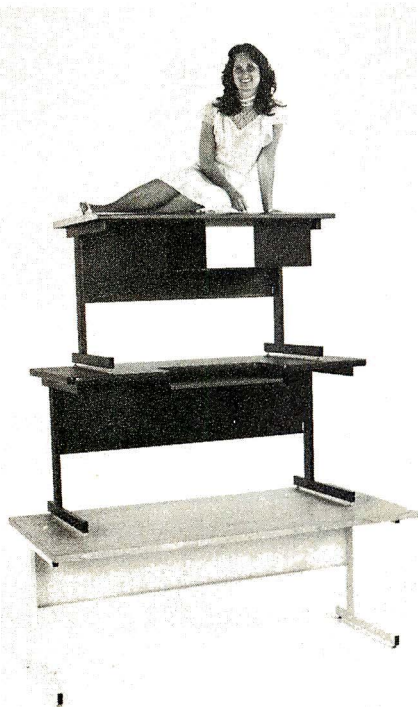


For a demonstration of the Wang Professional Computer, call 1-800-225-9264. Or send this coupon to: Wang Laboratories, Inc., Business Executive Center, One Industrial Avenue, Lowell, MA 01851.

Name _____
Title _____
Company _____
Address _____
City _____ State _____ Zip _____
()
Telephone _____

WANG

**The Office Automation
Computer People.**



TOP OF THE LINE

When it comes to computer furniture, CF&A is head and shoulders above the rest. How did we get there? By offering a top line of desks, workstations, enclosures, and printer stands priced to make your bottom line look good. With attractive styling and durable construction plus personal service and prompt delivery. It all adds up to a top line of products from a top of the line company. Computer Furniture and Accessories... our name says it all.

CF&A

**Computer Furniture and
Accessories, Inc.**
515 West 132nd Street
Gardena, CA 90248
(213) 327-7710

See us at Comdex booth 517.

avoid interfering with or changing the host system. An example of the relationship between the tools and the host system is illustrated in the implementation of the Carousel Tool-kits on CP/M (see figure 1).

File and Directory Names

The Software Tools view all I/O operations as actions on named files. As in Unix, use of files from within programs must be as device independent as possible because the program does not know whether the I/O is being done with a terminal, file, or another program. The file to be used is specified when the program is run instead of when it is compiled. When the host provides some sort of directory structure, it should appear to the user as the Unix model of a hierarchical directory structure does. These requirements have effects at both the RATFOR library level and at the tools execution level. For example, some allowed file names with the tools on CP/M are:

data	the file "data" on the current directory
/b/data	the file "data" on drive B in the current user area
/2/a/data	the file "data" in user area 2 on drive A
/tty	the programmer's terminal
/nul	the "bit bucket," a place for unwanted output
/lst	the printer

File names of these forms can be used anywhere a file name is needed. For example, in the tools *open* primitive, the statement

```
fd = open
      ("/0/c/foobar.dat", READWRITE)
```

results in the file /0/c/foobar.dat being opened in a mode allowing random reads and writes. The command

```
diff /1/b/prog.bas prog.bas
```

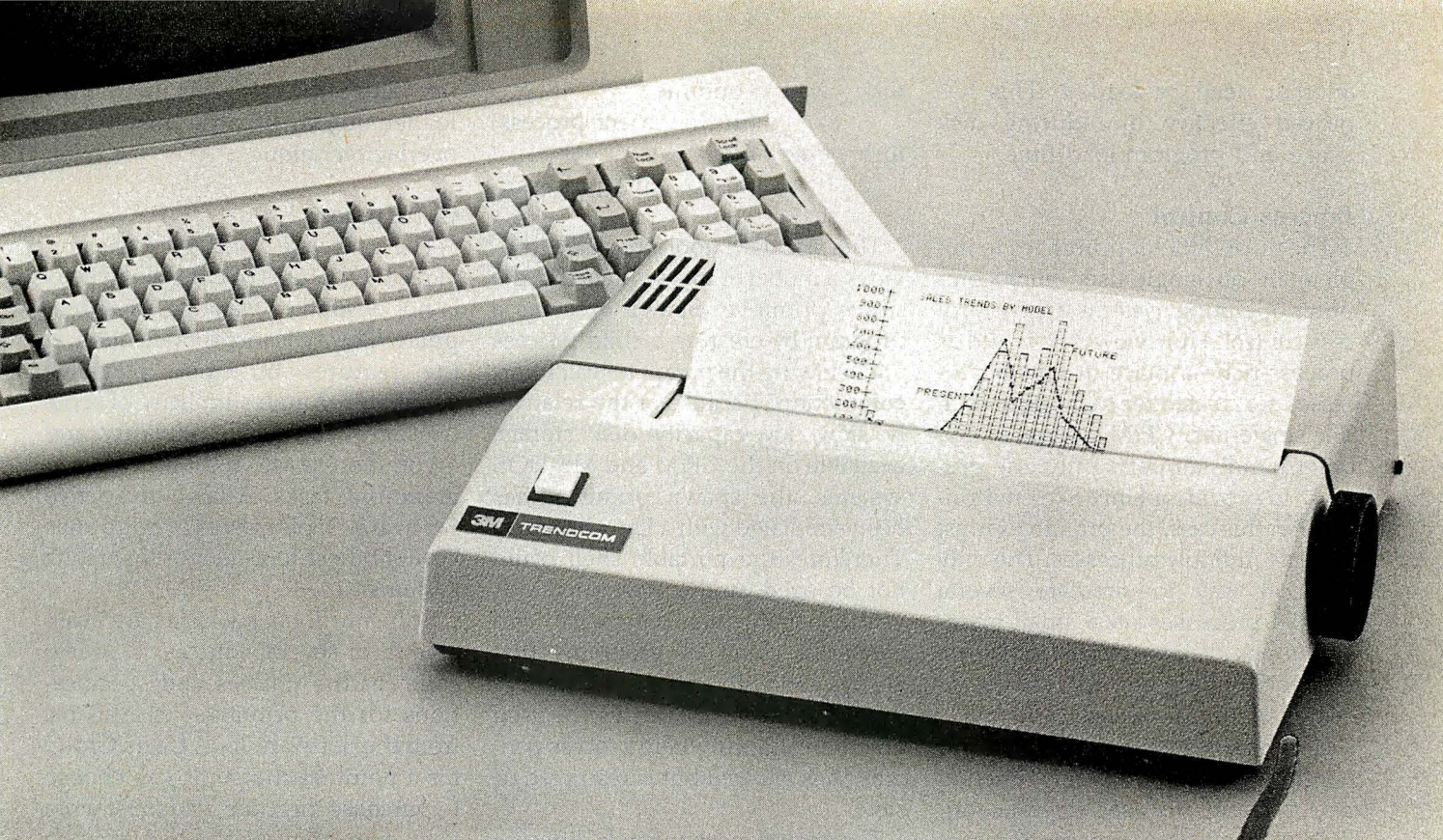
displays the differences between the version of prog.bas on drive B in user area 1 and the version in the current directory. By putting CP/M's user-

area number at the higher level in the hierarchy, a programmer can operate within a given area on several drives without specifying the user area. In accordance with the prime directive, a CP/M style of directory naming is also recognized (e.g., 1b:prog.bas). In addition, the temptation to further follow the Unix style and allow user-named subdirectories, as opposed to the hard-wired CP/M user/disk names, was tempered by the prime directive's requirement that all tools files be available on the host system with recognizably similar names.

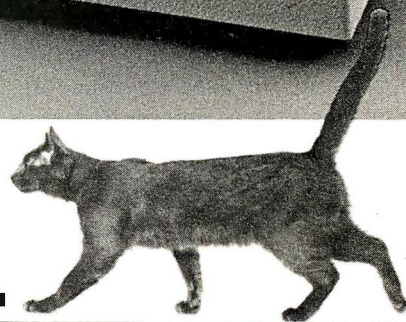
Memory Allocation and Disk

The tools package includes primitives to dynamically allocate memory areas for temporary use within a program. This feature has proven easy to provide on single-user systems such as CP/M and MS-DOS, where the programmer has access to all memory not occupied by the program or operating system. However, bulk-storage I/O devices, usually floppy disks, are so slow that it is desirable to use as much high-speed memory as possible for a cache of recently used or soon-to-be-used data. These two requirements force the dynamic-storage primitives for CP/M to share the memory with the I/O primitives. This provides the tools with dynamically available storage while using all remaining memory to speed up disk operations.

The Software Tools package also enables a user to quickly access the large collection of the tools' utilities on a small system. Sixty nontrivial tools could easily occupy a large amount of disk space. Unlike integrated programs in which all functions are available to the user within one large complex program, the tools are a collection of single-purpose programs, each of which must be loaded into memory when needed. To provide both fast program load times and small disk-space usage on CP/M, the tools were stored on disk as overlays of each other. Because they all share the common primitives, the primitives need be loaded into memory only once. When a tool program is run, only the part of the program that is different from one tool to



3M introduces the Purrrrrrrrrsonal Printers.



Quiet, high-quality printers at prices that won't scratch holes in your pocketbook.

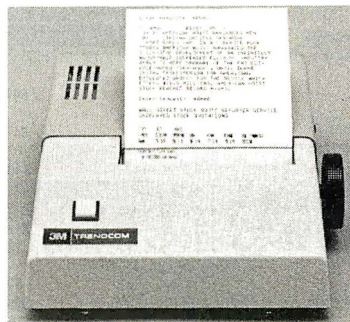
\$299, manufacturer's suggested retail price, for a printer that puts 80 columns of full-size text, 136 columns of compressed print, and computer-generated graphics on crisp white paper. Or \$249 for the 40/80-column model.

Price is about the only way that our new 3M Silent PC Printers draw attention to themselves. At home or in the office, they're virtually inaudible since their non-impact electronic printing elements skim across the paper with none of the monkeylike chatter that you hear from ordinary matrix printers. Result: Less distraction for you, and less disturbance to others.

Our Silent PC Printers differ from most other sensibly-priced printers in another way, too. They print sharp, black characters from one end of the paper roll to the other, since there's no ribbon to wear out.

Maintenance is kept to a minimum since 3M Silent PC Printers have only two moving parts: the pinless printhead and the DC stepper-motor paper drive. They come with a one-year warranty (90 days on Printhead), and they're made in the U.S.A. by 3M, a company whose name is synonymous with quality.

Either serial or parallel interfaces are available. For literature, and for the name



40/80-column Silent PC Printer

of a dealer who can give you a demonstration, call **800-538-8157 Ext. 928** toll-free. (In California, call **800-672-3470 Ext. 928.**) In Canada, call **1-800-268-9055** and ask for operator #11. Or, if the cat's got your tongue, mail the coupon instead.

Mail to:
3M Business Communication
Products Division
Attn.: G. Collins
3M Center—Building 216-2N
St. Paul, MN 55144

- ☐ Please send literature on the new 3M Silent PC Printers and tell me where I can buy them.
☐ I'd like a demonstration soon.

Name _____

Title _____

Phone (____) _____

Company _____

Address _____

City _____

State _____ Zip _____

3M Business Communication Products Division

3M hears you ...

3M

another need be loaded. This has proved effective in reducing disk usage and program load time.

Process Control

The most difficult primitives to implement on single-user microcomputer operating systems are for process control. Unix views the world as process-rich—a place in which processes are created for each command. The single-user CP/M system, on the other hand, supports only one process. To provide a Unix-like environment in this case, the primitives must emulate multiple processes. The only practical way to simulate several parallel processes on a small-memory, floppy-disk-based system is by a sequence of programs that are not executed simultaneously.

Unix enables process creation and program execution by the function pair *fork* and *exec* (see reference 4). *Fork* creates a clone process and *exec* overlays the current process with a new program. The most common sequence in Unix is

fork - wait - continue
(in the parent process)
fork - exec - die
(in the child process)

The standard tools package provides a model of this sequence in the *spawn* primitive. *Spawn* executes a program by creating a child process and allowing the parent to wait for its completion. Because of the relatively slow, low-capacity disk storage available on the CP/M and MS-DOS systems, the *spawn* primitive has been simulated with a Unix-like *exec*. Therefore, the portable shell could not be used, and a new shell was written that uses only *exec* and creates a chain of programs that always end with a new invocation of itself. This new shell can also be used on other systems where process generation is allowed but is restricted or slow.

The *spawn* mechanism is different from those used by other command-interpreter replacements for CP/M that always expect to reside in mem-

ory. The Software Tools utilities are loaded quickly because they use the overlay technique.

Conclusion

The Software Tools package provides the features of Unix when Unix is not desirable, available, or appropriate. The tools incorporate many of the features of Unix: elegance achieved through simplicity of style, consistency of use, modularity, and a common-sense approach to programming tasks. A large and active Software Tools Users Group has brought these tools to most operating systems.

Software Tools packages are available from several sources. A source code for the utilities and specifications for the primitives is available from the Software Tools Users Group for a nominal charge. If you choose to purchase this code, you must write your own primitives, which may be difficult.

You may be able to obtain a complete tools implementation for your system from someone who has already done it for a similar system. The tools group distributes versions for a few minicomputers and mainframe systems. These are provided without support.

You may also purchase specific implementations of the Software Tools from a vendor. If you do so, you should expect a version of the primitives optimized for your system, with continuing support and contact with the Software Tools Users Group. ■

References

1. Hall, Dennis, Deborah Scherrer, and Joe Sveteck. "A Virtual Operating System," Communication of the Association of Computing Machinery, volume 23, number 9, pages 495-502.
2. Kernighan, Brian W. "Why Pascal Is Not My Favorite Language," Bell Laboratories Technical Note #100, Murray Hill, NJ.
3. Kernighan, Brian W., and P. J. Plauger, Software Tools. North Reading, MA: Addison-Wesley, 1976.
4. The Bell System Technical Journal, volume 57, number 6, part 2 (July-August 1978).

Deborah K. Scherrer, Philip H. Scherrer, Thomas H. Strong, and Samuel J. Penny can be reached at Carousel Microtools, Inc., 609 Kearney St., El Cerrito, CA 94530.

Novation

C. ITOH
Prowriter 8510 AP, 10"
Carriage, 120 CPS, 80 col.
Parallel **CALL**
Prowriter II, 1550P, 15"
Carriage 120 CPS, 132 col.
Parallel \$645
Starwriter F-10, 40CPS,
Daisywheel, Parallel \$1145
Printmaster F-10, 55 CPS,
Daisywheel, Parallel . \$1550

STAR
Gemini-10X, 10", 120 CP **CALL**
Gemini-15, 15", 100 CPS \$465
Delta-10, 10", 160 CPS **CALL**

EPSON FX-80 . . **CALL**
FX-100 . . **CALL**

Multi-Function
Cards for the
IBM-PC & XT
SuperDrive &
SuperSpool

Standard card with 1 Serial port and Clock/calendar . . \$125
Additional Parallel, Game, or Serial port add \$ 35

SIXPACKPLUS CARDS
Offers maximum of 384K RAM, Parallel and Serial ports,
clock/calendar, optional Game port, SuperDrive and SuperSpool
Amount of memory . . . 64K 128K 192K 256K 320K 384K
Parallel, Serial ports clock \$275 \$345 \$425 \$485 \$535 \$595
Optional Game port add \$ 35

COMBOPLUS II CARDS
Offers a maximum of 1 256K, 1 Parallel and Serial ports,
clock/calendar, SuperDrive and SuperSpool Software
Amount of memory 64K 128K 192K 256K
all options Parallel, Serial ports clock \$285 \$335 \$385 \$435

MEGAPLUS II CARDS
Offers a maximum of 512K RAM, SuperDrive and SuperSpool
Amount of memory 64K 128K 192K 256K
w/1 Serial port & clock \$287 \$337 \$387 \$439
w/ either 2nd Serial port or Parallel port \$320 \$370 \$420 \$470
w/ Parallel and additional Serial port . . \$355 \$400 \$455 \$509
Megapak, expands MegaPlus II from 256K-512K . . \$330

μ-SCI

RGB COLOR
PRINCETON HX-12, 12" (690/
line), Hi/Res, Made for
IBM \$475
AMDEK Color II, 13"
Md/Res \$475
BMC AU-919MU (680/line),
13" Made for IBM . . \$455

GREEN SCREEN
AMDEK Video 300, 12" Hi/Res,
Green \$145
BML BM-12A, 12" Green \$89
GORILLA, 12" Green,
Hi/Res \$90
USI P11, 9" Green . . . \$135
P12, 12" Green . . . \$155

AMBER SCREEN
Video 310-A, 12" Amber,
Made for IBM-PC . . . \$175
TAXAN KG-12N UY, 12" AMBER
800 Dot \$150
USI P13, 12" Amber . . \$175

ORDERS ONLY
TOLL FREE
1-800-231-0616
INFORMATION
713/692-7100

TO ORDER:
Please send money order or cashiers check Personal
or company checks require 2-3 weeks to clear. Prices
reflect a 3% cash discount. Mastercard/Visa accepted.
Shipping, insurance & handling charges are 3% of total
order value by UPS Ground. 5% UPS Blue Label Prices
subject to change. All brands are registered trademarks

DISK DRIVES
Micro SCI A2 . . . \$245
Rana Elite I . . . \$260

DISKETTES
Elephant SS/SD . . \$18
Elephant SS/DD . . \$22

5 1/4" TANDUM DRIVES
TM100-1, SS/DD . . \$195
160K \$245
TM100-2, DS/DD
320K \$245

APPLE COMPATIBLE
Disk Drive . . . \$225
Controller Card . . \$55

Hayes
Micromodem II \$258
Smartmodem 300 \$205
Smartmodem 1200 \$519

Quadram
Quadboard-64K . . \$295
Quadboard-128K . \$345
Quadboard-192K . \$385
Quadboard-256K . \$445
Each board has a parallel
and serial port.
Quadram drive, OSpool
clock 256K ability and
Quadmaster software

EAGLE SOFTWARE TM 6121 Stuebner Airline
Houston, Texas 77091
713/692-7100

We'll Match Any Advertised Price in this Magazine!