PEEK (65)

The Unofficial OSI Users Journal

$1.75

JULY 1982
Vol. 3, No. 7

P.O. Box 347
Owings Mills, Md. 21117
(301) 363-3267

INSIDE:

KeyFamily Computer Systems Announced

PUBLISHERS NOTE

To date, all comment has been left to the province and good care of your Editor, but OSI's invitation to interview their leaders, just prior to NCC, was an opportunity that could not be missed.

Although armed with tape recorder, for a quiet, one on one interview that could nicely be placed directly on our pages I found myself surrounded by up to four people and answers coming from many directions. Consequently, what follows is a synopsis of three hours of conversation with the following: Chuck Bickoff (VP-Engineering), John Wolaver (VP-Marketing), John Werderman (VP-Finance), Bob Welch (North American and International Sales Manager), and Phil Johnson (Corporate Communications Manager).

In general, it seems safe to say that the days of idle promises of things that never happen are over. OSI under the guidance of Bill Chalmers has undergone a most courageous metamorphosis. There is scarcely an area of the computer manufacturing business that has not been touched by his presence. Corporate headquarters has been moved to Bedford, Mass and the Bedford Heights, OH facility closed. A bevy of new, talented and dedicated people have been added to the OSI team, thus bringing an unheard-of aura of professionalism to the OSI image. Hardware has been completely reworked, to take OSI from the early erector set image to a machine that will be comfortable in the most modern office environment. A number of long overdue improvements to the inner workings have been effected. The software team has pressed forward with new operating systems, utilities and associated programs.

PEEK (65) is published monthly by DBMS, Inc.
Owings Mills, MD 21117.
Editor - Al Peabody
Technical Editor - Dickinson H. McGuire
Asst. Technical Editor - Brian Hartson
Circulation & Advertising Mgr. - Karin Q. Gieske
Production Dept. - A. Fusselbaugh, Ginny Mays

Subscription Rates

US (surface)	$15
Canada & Mexico (1st class)	$23
So. & Cen. America (Air)	$35
Europe (Air)	$35
Other Foreign (Air)	$40

All subscriptions are for 1 year and are payable in advance in US Dollars.

For back issues, subscriptions, change of address or other information, write to:

PEEK (65)
P.O. Box 347
Owings Mills, MD 21117

In short, Ohio Scientific is a thing of the past and M/A-COM Office Systems is virtually a new entity, capitalizing on the good aspects of the old OSI aided by the backing of M/A-COM (a Fortune 500 company) and by new direction to catapult OSI back to the forefront and leading edge of the micro world.

All that kind of talk may sound like a sales pitch, but it is in fact the kind of thinking that oozes from anyone you talk to at Bedford. They mean business and they are either going to make it or die trying.

With all this effort on the business front, will OSI support the personal users? It is no secret that the lion share of OSI effort is now directed to the business world, but OSI is quick to point out that although there are no plans at present to advance the Personal line of computers, they will continue to sell and support the existing market. In the same breath, there are hopes that the Hi-Res board will soon be accepted by the FCC for retrofit use in the "P" machines and certainly 65D-V3.3 was a recent giant step forward at a modest $75.

What's this we hear about new names for OSI products? KEY is the word now! The KEYFAMILY is now made up of KEYMATE (the "P" line), MASTERKEY (the business machines), KEYWARE (software), KEYWORD (word processing) and the like. Now you can understand that the C4P-MF with hi-res graphics is a KEYMATE 100.

What will OSI display at the NCC and COMDEX shows? There will be a group of 250J's (74 M-byte) operating in a network, several 230's running, amongst other things, KEYWORD. Also to be announced at NCC is the new KEYMATE 150. Cross between personal and business, this new entry can be either a stand-alone or an intelligent terminal, sports both 6502 and Z-80, 64K of RAM. Mini-floppies that will store 327 K-bytes and a 10 M-byte Winchester hard disk will be on line soon. Two of the five slots are left open for expansion into the time-share world (RAM for two users per board) thus a four user system. All this in a small table-top unit. Just add a terminal, printer, modem, or use the network port to tie into KEYRING (OSI'S network) as a local intelligent terminal or network node.

Admittedly, all the trimmings are not ready yet, but they are planned for fall delivery.

The MASTERKEY 200 series are out. The photo says a lot. On the left is a C 100, in the middle top is a 220C or 230C, below it a 250 with two hard disks, and to the right a 230E with hard disk and up to four users. What you don't see in the picture is that all of the 200's share a common and new CPU drawer; completely enclosed, well ventilated, with either 8 or 16 slots (depending upon how many disks are used) of back plane, new CPU board with both 6502 and Z-80 chips and new disk controller boards.

By Fall, OSI's crown should be in place. Another new machine featuring multi-processing (a CPU for every user) with a choice of either 6502 or Z-80 (OSU or CP/M), KEYRING , all the trimmings and all at the same time. Who else can match that?

Where else will the new OSI equipment be displayed? In addition to NCC, roughly the same display will also be at the Hanover Fair, Comdex/East and West and also at Comdex Europe.

The absence of OSI advertising since the end of last year has many people worried. Will OSI advertise again and if so when and where? An interim plan gets under way during the month of June, with special emphasis on the industry. The budget for the year beginning in October calls for something in excess of a million dollars to be spent among the various forms of advertising: media, public relations publications, direct mail and trade shows. Again, the silence was part of the policy that says, "Better do it right than quick".

Is anything being done about what has, frankly, been poor documentation? The "preliminary" manuals are gone. A crew of both inside and outside talent are completely overhauling every user document produced by OSI. The drafts that I have seen are simpler, easy to read and complete. The energetic plan is to release one new document a week, which will probably be under way by the time you read this. Among the items are user guides for the 220, 230, and 250, a new graphics manual, a complete system guide and a setup guide. Then comes a friendlier and easier to use Planner reference

document and later a tutorial, a complete rewrite of the 65U 1.42 reference manual and lastly a manual for KEYWORD - the new CP/M based super word processor and a maintenance manual in three parts down to the board level with diagnostics. Before the team finishes, everything will be redone.

Is anything being done to help the dealers be better dealers from the technical standpoint? Starting in July, dealers will be attending a series of new professionally structured two and three day seminars on sales, the operating system, and field service. The obvious result will better and more knowledgeable service to OSI end users.

Will OSU be compiled? Most likely not - with good reason. 1. Because of the commingling of BASIC and the OP system, it would be a whale of a job. 2. The current compilers available for OSU are of the single pass type yielding very little shrinkage in program size. 3. There is a better way to achieve speed and maintain the flexibility of BASIC - Multiprocessing.

You mentioned KEYWORD. Can you give us more detail? KEYWORD was produced by Designer Software, Inc. whose people were the key designers of Magic Wand. They assure us that their product will be better and much easier to use than Magic Wand, have all of the features of Word Star, plus a typewriter option, more powerful screen control capabilities, external document merge, a glossary-like function called Lexicon, automatic footnoting and several others. This product runs under CP/M. In the mean time, the search continues for a better WP to run under OS65U, but KEYWORD is needed now and is ready now.

Where does OSI stand with regard to 16 bit processors? They are actively studying 16's for release sometime within the next twelve months once they have selected the architecture. For the moment, there's no rush since there is very little software written for the 16's. In the meantime, time can be better spent cleaning up the 8's besides 8 bit multi-processing will run circles around any 16 bit time-sharer!

When all was said and done, several things became obvious about OSI. The "metamophosis" was worse than trying to start a new business and it has taken longer than was originally anticipated. No one at OSI is going to rush out to the market place with half-baked product - better right than quick. The people calling the shots are part of the new team that was carefully picked and have lengthy and proven track records in the computer industry. In all probability, OSI will move ahead as never before.

PEEK [65] will continue to keep an eye on the goings on in Bedford and keep you informed!

OS-DMS/dBASIC II a comparison

by Al Peabody

We have all seen the ads:

"10 REM ACCOUNTING

20

Boy, is this costing you..."

The clear implication is that there is a much better way to make your computer do its job than by writing programs in BASIC. That way, we are told, is to use a "database manager," specifically Ashton-Tate's dBASE II.

In this article, I hope to discuss just what dBASE II is, and what it can (and cannot) do to improve your computer's performance. Also, I will compare dBASE II with OS-DMS, the "standard" M/A-COM OSI database manager.

What is a DBMS, Anyway?

A DBMS, or Data Base Management System, is a specialized set of programs, perhaps even running in a special machine, which is designed to handle the details of storing information on mass storage devices, specifically floppy or hard disks. If a DBMS works well, the programmer doesn't have to worry about just where and how the information is stored away on the disk. He just knows it's "out there" somewhere, and that he can sort it, look through it, edit it, put it together in reports in various ways and generally use the information.

A major advantage of a DBMS is that it is not necessary to define the data you will be working with in every program you write. The DBMS, in other words, adds another "level" of software between you and the disk system, letting you talk about a piece of information in terms like "balance due" or "part number," rather than "track 24, sector 09," or even "record 125, Field 12 as PARTNUM."

If it works well, a DBMS can be a terrific time saver. If it works poorly for a given application, a DBMS is just one more thing to worry about, one more impediment to useful functioning of your computer.

Fortunately, both OS-DMS and dBASE II work well enough to be a real help. Unfortunately, neither works well enough to be the final answer to the problem. Let's look at how they do work, and what is wrong (and right) with them.

How Does OS-DMS Do it?

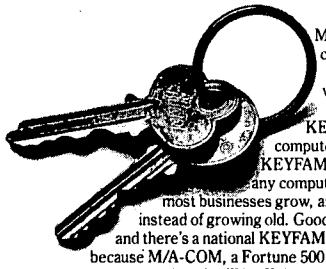
OS-DMS is a set of programs, written in BASIC to run under OS- 65U, which performs the functions of managing a collection of information, a Data Base. This it does using two basic types of data files, a Master File and a Key File.

Each Master File contains information related in some way, such as all the information concerning a list of customers or items in inventory. The Master File is divided into Records, each of which is further divided into Fields. Each Field is the unit of information with which we really must work. For example, "price" might be a Field in each Record of an inventory Master File. The problem is, each program which works with the Master File must know what information is in the file, and where it is. The old way, before DBMS, was to keep a list of any files which programs would share (edit, generate reports from, add records to, etc.) and to encode into each program the lengths and positions of all the fields within the records, in effect redefining the file structure in each program.

Nucleus

OS-DMS solves this problem by storing the information needed to work with a Master File (file name, file type, actual beginning and current ending

New KEYFAMILY small business computers for every size business.



M/A-COM Office Systems, Inc., formerly Ohio Scientific, designed the KEYFAMILY computer line for every size business.

But we didn't stop there. We made KEYFAMILY good for business. In more ways than one.

Good for business because you have the entire KEYFAMILY line to choose from—everything from personal computers to multi-user systems. Good for business because KEYFAMILY gives you the best price/performance ratio of any computer on the market. Good for business because most businesses grow, and KEYFAMILY small business computers grow with you instead of growing old. Good for business because all computers need occasional servicing, and there's a national KEYFAMILY dealer network at your beck and call. Good for business because M/A-COM, a Fortune 500 digital communications and information-processing company, stands behind KEYFAMILY with its technological resources and financial stability. That's not only good for business, it's good for your confidence.

If you are thinking about buying a computer right now, why not get the line that's good for business. KEYFAMILY systems are now available through our national network of KEYFAMILY dealers. For the dealer nearest you, or for more information, write or call M/A-COM Office Systems, Inc., 7 Oak Park, Bedford, MA 01730; 1-800-C·A·L·L·O·S·I.

M/A-COM OSI

KEYFAMILY. Good for business.

position of the data in the file, names and lengths of all the fields in each record in the file, maximum number of records the file can contain and so forth.

Thanks to OS-65U's file structure, which allows every byte in a data file to be addressed individually, this is relatively simple: a position in the file is assigned to each piece of information, and 65U puts it there for you.

What all this means is that a "Nucleus" of utility programs can be written to deal with OS-DMS Master Files: an editor, a report generator, a record inserter, a record remover, a sorting program, and so forth. None of these programs has to be re-coded to deal with different Master Files. Each must simply look into the beginning of a Master File and determine what its name is, what information it contains and where.

In fact, there is just such a nucleus of Utilities, called the "DMS Nucleus." It can create, edit, sort and create reports from DMS Master Files. It can also create what is called Key Files. Key Files are just as simple as Master Files, and equally useful. Suppose, for example, you have an inventory file, which includes part name, part number, supplier, cost, price, number on hand, etc., etc. In order to generate a list of parts on hand, it might be nice to have this file sorted by part name, or maybe part number. To generate a list of suppliers, it would be nice to have the file sorted by supplier name. To generate another report, you might want to have the file sorted by price, or maybe cost, or perhaps number on hand.

One way to do this would be to make 9 copies of the file, and sort each one differently. But then your editor program would have to update all 9 copies whenever you received a new shipment of widgets or sold a couple of frammisses.

Another way to do it would be to sort the file differently each time you wanted to write a report. But then you would spend half your time looking at the message "PLEASE WAIT... SORTING" on your computer screen.

OS-DMS does it by creating Key Files. Listen up, now. A Key File contains ONE FIELD (the same field, such as part number) out of EACH RECORD in a Master File, immediately followed by the location of the start of that record in the Master File.

That means we can create a Key File of part numbers, another of part names, a third of supplier names, and in fact up to seven Key Files for each Master File (why just seven? I don't know. M/A-COM OSI wrote the programs in the Nucleus which create, load, sort and use Key Files to allow just 7, that's why).

Then we can sort each of these Key Files. Since a Key File contains just one field of each record in the Master File, it is much shorter, and sorts much faster.

And THEN, when we want to print a report sorted in order by part number, we just look each time at the next part number in the (sorted) Key File, see where that record is located in the Master File, read out the whole record, print out whatever we need for our report, then go back to the Key File to find where the NEXT record in the Master File is, in part-number order.

So the DMS Nucleus allows us to create Master Files; to load them with data; to edit them (change part of the information in them); to create, load and sort Key Files based on the Master

digital technology

BUS-II LEVEL I BOOKKEEPING & ACCOUNTING SYSTEM

The BUS-II turn-key multi-client accounting package is the leading OSI business software package. BUS-II Version 32 includes five principle modules:

	Inst. Price	List Price
GENERAL LEDGER	$1200	$599
ACCOUNTS RECEIVABLE (a)	1000	599
ACCOUNTS PAYABLE (a)	1000	599
ORDER ENTRY W/ INVENTORY (a) (b)	1000	599
PAYROLL (no extra charge for optional versions)	1200	799

01 - STANDARD MULTI-STATE OPERATION
02 - CPA FIRMS & SERVICE BUREAUS
03 - RESTAURANTS
04 - COMMISSION SALES
05 - CONTRACTOR'S JOB-COST ACCOUNTING

The Accounts Receivable, Accounts Payable, and Order Entry W/ Inventory are completely interactive with the BUS-II General Ledger. Two optional specialized packages (completely interactive) are also available.

CPA EXTENSIONS (see below)
POINT-OF-SALE TERMINAL W/ INVENTORY (see below)

The BUS-II CPA EXTENSIONS Package provides special features for accountants and bookkeepers. The POS-1 Point-of-Sale Terminal package enables the operator to use the computer system's video terminal as an on-line "electronic cash register."

Note: BUS-II operates on floppy-disk or hard disk-based systems running the OS-65U operating system (single-or multi-user). Multi-client use can accommodate any number of client companies on floppy disk systems or hard disk system with H/D/E (required for hard disk use). BUS-II LEVEL I files are limited in size for floppy disk back-up; floppy disk operation continues in case of hard disk failure.

BUS-II "SOFTWARE EXCHANGE" SPECIAL

Users of other business software packages who wish to upgrade to Digital Technology's full-function BUS-II BOOKKEEPING & ACCOUNTING SYSTEM can, in many cases, "trade in" their old, unuseable, or unsupported software for full rebate of the original purchase price (up to $750.00). This "SOFTWARE EXCHANGE" offer includes virtually all of the previously-available OSI business software packages. Contact Digital Technology or your dealer for more information.

BUS-II LEVEL II (EXPANSION TO BUS-II LEVEL I)

BUS-II LEVEL II is designed for much larger businesses. Expanded file size and special operations allow virtually unlimited numbers of accounts and transactions. BUS-II LEVEL II requires BUS-II LEVEL I. Minimal back-up is data cassette (tape) or floppies--although multiple Winchester disk operation is recommended (provides ability to continue computerized bookkeeping functions in case of hard disk failure.) H/D/E Hard Disk Executive is required.

	Inst. Price	List Price
GENERAL LEDGER (c) (d)	$	$399
ACCOUNTS RECEIVABLE (c) (d)	600	399
ACCOUNTS PAYABLE (c) (d)	600	399
ORDER ENTRY W/ INVENTORY (c) (d)	600	399

CPA EXTENSIONS PACKAGE

CPA EXTENSIONS is designed for public accounting firms. A number of special operations are provided: "bankers" Balance Sheet and Profit and Loss statement with summarization and consolidation options, Statement of Changes in Financial Position, Statement of Changes in Components of Working Capital, Cash Flow Analysis, Departmentalized Sales Analysis, Asset Depreciation Schedule (compatible with TAXMAN-1040), and Loan Amortization Schedule. In addition, a pre-processed or "after-the fact" payroll system is provided.

CPA EXTENSIONS is interactive with BUS-II 32 BOOKKEEPING & ACCOUNTING SYSTEM

CPA EXTENSIONS (a) Inst. Price $2400 List Price $1500

POINT-OF-SALE TERMINAL

POS-1 is an on-line multi-store point-of-sale terminal program with integrated inventory designed for cash register emulation. POS-1 controls cash drawer and ticket printer (or system printer). Automates taxable or trantaxable sales, cash transactions, and credit sales (with verification operations). POS-1 also allows the use of industry-standard bar code readers with the point-of-sale terminal system through a "Siamese port-- on the C2 or C3 CPU card. (Extra serial port NOT needed except in multi-user operation.) Configured for industry-standard RS232 bar code "wand" (INTERMEC) or "window" (SPECTRA-PHYSICS).

POS-1 is interactive with the BUS-II V 3.1 BOOKKEEPING & ACCOUNTING SYSTEM.

POS-1 POINT-OF-SALE TERMINAL (a)(b) Inst. Price $1600 List Price $1199

TAXMAN-1040 PERSONAL INCOME TAX PREPARATION

TAXMAN-1040 is designed for tax practitioners and public accountants. TAXMAN-1040 is the leading tax package for OSI micorcomputers--the package has been installed on OSI, Hewlett-Packard, DEC and IBM systems. Designed and supported by CPA tax experts. This package automatically prepares FORM 1040 and 32 schedules. Support includes annual forms, tax tables, and computational revisions in accordance with Federal Tax Law changes.

TAXMAN-1040 Inst. Price $3600 List Price $2399

H/D/E HARD DISK EXECUTIVE

Digital Technology's implementation of H/D/E is the answer to AMCAP's HDM. Digital Technology's H/D/E provides user functions not found on HDM of similar products: ability to copy from any user "system" to another; automatic recovery in case of "back-up to floppy" or "restore from floppy" utility failures, allowing the user 3 options: (1) ignore error, (2) abort to menu, (3) try again; use of both "A" and "B" floppy drives to back-up hard disck files; and automatic back-up diskette initialization. H/D/E operates on any OSI Winchester disk system from 7 - 80 megabytes. Re-use of hard disk space is provided. Superior to AMCAP's hard disk manager in every respect (and Digital Technology software does not self-destruct).

NOTE: H/D/E is required when installing any Digital Technology business applications packages on OSI hard disk systems.

H/D/E HARD DISK EXECUTIVE List Price $399

OS-DMX DATABASE MANAGEMENT SYSTEM

Command-oriented OS-DMS compatible database management system. OS-DMX operates under the OS-65U V1.2 operating system (single- or multi-user). Features such as control files, extensive operating commands and the innovative HELP function, make this one of the most usable--as well as powerful--systems available for microcomputers. OS-DMX may be used instead of, or in addition to, OS-DMS Nucleus, Query, Sort; OS-DMX will replace virtually all of the specialized OS-DMS modules-- and in most applications will provide greatly improved performance.

OS-DMX Database Management System buyers will receive (no extra charge) a number of "extras" previously sold separately:

DMX-MAIL	Mailing List Management (FEB 82)
DMX-STAT	Comprehensive Statistical Analysis package (JULY 82)
DMX-COPY	Edit Database Structure after the fact (FEB 82)
DMX-MERGE	File Merge Operation (FEB 82)
DMX-TUTOR	450-Pg Tutorial w/ Demo Data Diskette (AVAILABLE)

In addition, DMX-SORT operations will be upgraded to machine-code sorting for faster operation. There will be no need to purchase high-speed sort programs separately.

OS-DMX DATABASE MANAGEMENT SYSTEM Inst. Price $1600 List Price $1199

BISYNC-80/HASP

BISYNC-80/HASP is a full-function Multileaving Workstation package which allows communication with any remote CPU that supports a HASP Multileaving Workstation, and, as such, is ideally suited to Remote Job Entry applications.

OS-BISYNC-80/HASP (e)(f) List Price $1195

BISYNC-80/3270

BISYNC-80/3270 is a full-function IBM 3270 terminal emulator which allows the microcomputer to communicate over point-to-point telephone lines with any IBM S/360, S/370, or S30xx CPU that provides standard IBM support for one of the following:

IBM 3275 Model 2
IBM 3271 Model 2 or control unit w/ attached 3277 Model 2
IBM 3284 or 3286 printer

OS-BISYNC-80/3270 (e)(f) List Price $895

BISYNC-80/3780

BISYN-80/3780 is a full-function IBM 2780/3780 emulator allowing the microcomputer to communicate over point-to-point telephone lines with any CPU or device that provides standard IBM support for:

IBM 2780 Models 1, 2, 3 or 4
IBM 3780 w/ or w/o 3781 card punch
IBM CPU to CPU BSC communications

OS-BISYNC-80/3780 (e)(f) List Price $895

BISYNC-80/ASYNC

BISYNC-80/ASYNC is a full-function asynchronous communications package which allows microcomputers to communicate asynchronously with a mainframe or other microcomputers. This package is an ASYNC adaptation of BISYNC-80/3780 terminal emulation program, providing asynchronous communications at 75 to 9600 baud, using full IBM BISYNC protocol.

OS-BISYNC-80/ASYNC (e)(f) List Price $195

OS-BISYNC-80 SYNCHRONOUS INTERFACE ASSY List Price $395

NOTE: The prices shown in this catalog are estimates only; contact your OSI dealer for quotations. The "suggested installed price" reflects a typical business installation and includes reasonable allowance for software installation, minor program adaptation or customization, operator training, dealer support, back-up, etc. The "reference" or "list" price reflects a base price for the software for comparison purposes, exclusive of dealer installation and support.

Digital Technology, Inc., is the largest independent supplier of OSI software with hundreds of business packages in use around the world. Digital Technology software is sold by a growing number of conscientious OSI dealers and OEMs. Every package is backed by the finest support program in the microcomputer industry. All "bugs" are fixed free of charge. Updates (fixes to bugs, minor enhancements, new product announcements) are provided to all dealers and licensed users free of charge. And upgrades to new versions are encouraged (at nominal charge).

Digital Technology's software is user-obtained. In fact, no one else provides such expensive features as on-line documentation, idiot-proof prompting, and operator's manuals that are comprehensive, detailed, and accurate.

All Digital Technology software systems allow the operator to "set" the programs to the type of video terminal and printer used. The operator selects the terminal and printer types from the list provided in the "TERMINAL & PRINTER OPTIONS" program. Screen formatting and printer control are provided automatically yet may be redefined through user subroutines.

digital technology
inc.
P.O. BOX 178590
SAN DIEGO, CA 92117
(714) 270-2000

REQUIREMENTS

(a) BUS-II LEVEL I or LEVEL II G/L req'd
(b) BUS-II LEVEL I or LEVEL II A/R req'd
(c) Corresponding BUS-II Level I module(s) req'd
(d) H/D/E req'd
(e) C3 CPU W/ 56K RAM & OS-CP/M or Lifeboat Associates CP/M req'd
(f) SYNCHRONOUS INTERFACE ASSY req'd

Files; to write reports from Master Files, either directly or through "Key File access"; and to do such other functions as removing old records no longer needed or inserting blank records into the middle of a Master File.

All of which makes the writing of an inventory system (or name and address list or accounting system) much easier. But does NOT, in itself, represent any of these systems.

Modules

Here is where the DMS modules come in. Using the DMS file structure, using Master Files and Key Files, M/A-COM OSI has written several sets of programs with predefined files, predefined editing procedures, and predefined report formats, designed to do such jobs as keeping track of Accounts Receivable, Inventory, Accounts Payable and so forth.

They all look somewhat alike, since they all use the same basic file editors and file structures, with modified report writers and a great deal in common. They are all rather simple, almost simplistic, and while some would get the job done, none of them could be called sophisticated or high powered. The payroll system, for example, automatically deducted Ohio state income tax from every employee; the inventory system, A/R and A/P systems were little more than automated card file systems, keeping a record for each card you might have put in a card file, and allowing you little more choice than to change the information on the card or throw it away.

Expanding the System

Many people recognized the limitations of OS-DMS, but also recognized the potential power of a standard Master File structure, which could be read by the very programs which would work with it.

Before long, improved versions of Accounts Receivable systems, of Utilities (such as KYUTIL, capable of loading more than one field from each record in a Master File), machine language sorting routines, entire accounting systems began to appear, all based on OS-DMS. Now, many people just buy the Nucleus, and get their modules either from second sources such as DBMS,INC., or write them themselves, using the Nucleus Utilities to avoid having to reinvent the most basic wheels which make the system run.

Documentation

Another problem with OS-DMS has been its documentation. A first reading of an OS-DMS manual will never let you run it efficiently. In the words of Wallace Kendall, the manuals are "PCIPU -- Perfectly Clear If Previously Understood." I won't say too much about this aspect, since M/A-COM OSI is working very hard at the moment to bring all their manuals up to snuff.

Advantages

OS-DMS has many advantages, particularly for the programmer who wants to produce a new application.

All the utility programs are written in BASIC, and therefore, easily modified. Modifications to STAT03, the statistical report generator, have been published in PEEK(65).

The file structure is rigidly standardized, making it easy to write compatible applications packages, which will work with what others have written.

Perhaps best of all, OS-DMS runs under OS-65U, meaning the very powerful FIND command and byte-addressable file structure are there for the using.

Limitations

Many of OS-DMS's limitations are the flip side of its advantages. It is written in BASIC. Everything happens rather slowly for that reason.

Only 7 Key Files can be used, and inventing an editor which would update all the Key Files used with any given Master File at the same time the Master File is edited makes my head hurt. This means we have to reload and sort the Key Files each time we want to run a report which, while MUCH faster than resorting the Master File, is painfully slow.

Each step in each process involves running a BASIC program then, usually, returning to a menu, then running another BASIC program. Too simplistic, too slow.

★

```
10:      SINGLE DISK COPY UTILITY LISTING
20:
30:      RE ARTICLE PUBLISHED LAST MONTH
40:
50:           For OS-65D Version 3.X
60:           Written By:J.T.Dripps
70:
80:
90:            Dripps Electronics
100:
110:           Box 118 DeHaven Rd.
120:
130:           Beaver Falls,Pa.
140:
150:                    15010
160:
170:     Phone:(412)846-5311
180:
190:
200:-------------------------------------------------
210:
220             ERROR=$0000
230           SRTTKN=$0002
240           ENDTKN=$0003
250           RDINDX=$0004
260           WRINDX=$0005
270           TKZLVH=$0006
280           TKZLVL=$0007
290           TKZPGC=$0008
300              TS1=$0009
310              TS2=$000A
320            TABLE=$000B
330           SCTBYP=$00F9
340           SCTLEN=$00FA
350           SCTNUM=$00FB
360           STKADR=$00FC
370            MEMLO=$00FE
380            MEMHI=$00FF
390            STACK=$0100
400           SECTNM=$265E
410            PGCNT=$265F
420             HOME=$2663
430            TENMS=$267A   ;(TENMS+2)   $2678
440            SETTK=$26BC
450           WAITIH=$271D
460           RSACIA=$272E   ;(RSACIA+3)  $272B
470           LDHEAD=$2754
480           UNLOAD=$2761
490           INITTK=$277D
500            DKWTX=$27C2
510           DSKBYT=$27CD
520           DSKWRT=$27E1
530           SETSCT=$28C4
540           READDK=$2967
550           BPSECT=$2998
```

continued

AD$

Superboard II, C1P: TAC-MAN- Similar to PAC-MAN, avoid the ghosts while eating dots, cherries, and power pills. All this plus fast, smooth action, for only $9.95. (8K only). STAR FIRE: Similar to Defender. 4K-$7.95, 8K-$9.95. Send $1.00 for a complete catalog. Cassette only. Swany's OSI Software, 2652 37th, West, Seattle, WA. 98199. (206) 282-7376.

FOR SALE OR LEASE: Ohio Scientific C3OEM with 56K memory and 2 - 8" drives. Soroc IQ120 Terminal, NEC Spinwriter Printer, CPM, COBOL, FORTRAN BASIC OSI Data Base Management and General Ledger Packages. A/R and A/P. Lease of $302.86 per month can be assumed or Purchase for $6000.00. Contact Joe Myers 214-596-4507 or 3001 Bandolino, Plano, TX 75075.

FOR SALE: OSI C2-8P Challenger with 48K memory; OSI CD2 Dual Disk Drive, Koyo TMC-9M 7inch monitor; Integral Data Systems DMTP-G Matrix Printer. Asking $2,000. V. Baus, 151 E. Merrill, Fond du Lac, WI 54935, (414) 922-0970 before 5 p.m.

"Scientific Calculator" OSI 8K program with easy entry, input work sheet display, totals in Hex and Dec, Hex and Dollar modes selectable. 8K Tape $15. Send SSAE for Data Sheet and free utility program listing. Harry Hawkins, Box 4432, Burton, SC. 29902.

****

```
560            OSINP=$2C9B
570           BUFBYT=$2CE4
580           STROUT=$2D73
590           PRT2HX=$2D92
600           FLOPIN=$C000
610           FLOPOT=$C002
620            CLEAR=$FCD6
630;=====================================
640;=====================================
650;
660;
670;
680                *=$0200
690;
700;
710        LDA #$2E
720        STA $E2      ;SET HI BYTE OF INPUT BUFFER
730        LDA #$1E
740        STA $E1      ;SET LO BYTE OF INPUT BUFFER
750        JSR CLEAR    ;CLEAR SCREEN
760        JSR DSPLY1   ;DO FIRST VIDEO DISPLAY
770 START  JSR STARTK   ;GO DO START TRACK # ROUTINE
780        JSR GENDTK   ;GO DO END TRACK # ROUTINE
790        JSR CKERR    ;CHECK FOR START>END ERROR
800        BCC START    ;ON ERROR START OVER
810        JSR CLEAR    ;CLEAR SCREEN
820        JSR DSPLY2   ;DO SECOND VIDEO DISPLAY
830        JSR KEYINP   ;GET INPUT FROM KEYBOARD
840        BCC DBEND    ;IF CARRY CLR,THEN EXIT
850        JSR INTTBL    ;INITIALIZE TABLE INDEX AND
860                      ;SET MEMLO/MEMHI FOR 1st TK
870 RNXTTK JSR WTABLE    ;POSITION HEAD TO TK # IN
880                      ;SRTTKN,READS SCTLEN/SECTNUM
890                      ;/TRACK # AND WRITES THEM TO
900                      ;TABLE-READS TRACK TO MEMORY
910                      ;BUMPS MEMLO/MEMHI TO REFLCT
920                      ;PAGES READ AND WRITES THEM
930                      ;
940                      ;TO TABLE FOR READ ROUTINE
950                      ;
960 INCTKN LDA SRTTKN    ;GET THE TRACK NUMBER
970        SED           ;TRACK # STORED AS BCD
980        CLC           ;GET SET TO INC. TRACK #
990        ADC #01       ;ADD 1 TO TRACK #
1000       CLD
1010       STA SRTTKN    ;SAVE THE NEXT TRACK NUMBER
1020       CMP ENDTKN    ;HAVE WE READ ALL THE TRACKS
1030       BCC RNXTTK    ;NO,GO READ NEXT TRACK
1040       BEQ RNXTTK
1050       DEC WRINDX    ;DROP WRITE TABLE INDEX BY
1060       DEC WRINDX    ;TWO (WRINDX=RDINDX=END)
1070       LDA #$00
1080       STA SRTTKN    ;CLEAR START TRACK HOLD
1090       JSR DSPLY3    ;DO THIRD VIDEO DISPLAY
1100       JSR KEYINP    ;GET KEYBOARD INPUT
1110 DBEND BCC END       ;CHECK EXIT FLAG
1120       JSR CLEAR     ;CLEAR SCREEN
1130       LDY #$00      ;SET'Y'TO INDEX THROUGH TABLE
1140       STY RDINDX    ;AND SAVE IN READ INDEX HOLD
1150 WNXTTK JSR RTABLE   ;READ LOCATIONS FROM TABLE
1160       LDY RDINDX    ;GET READ TABLE INDEX
1170       CPY WRINDX    ;HAVE WE WRITTEN ENOUGH TKS
1180       BNE WNXTTK    ;NO,THEN WRITE NEXT TRACK
1190       JSR STROUT    ;PRINT MESSAGE
1200       .BYTE $0D,$0A,$0A,$0A
1210       .BYTE 'AGAIN(Y/N):',0
1220       JSR KEYINP    ;GET KEYBOARD CHARACTER
1230       BCS START     ;IF 'Y' THEN RUN AGAIN
1240       JSR CLEAR     ;IF 'N' THEN...
1250       JSR STROUT    ;PRINT REBOOT MESSAGE AND
1260       .BYTE $0D,$0A,$0A,$0A,$0A,$0A
1270       .BYTE              'DISK COPY COMPLETE'
1280       .BYTE $0D,$0A
1290       .BYTE $0A
1300       .BYTE                '  RE-BOOT SYSTEM  ',0
1310  END  RTS           ;AND EXIT THE COPY UTILITY
1320 KEYINP JSR OSINP    ;PUT KEYIN CHAR.IN BUFFER
1330       LDA #$00
1340       STA BUFBYT+1  ;SET BUFFER OFSET TO 0
1350       JSR BUFBYT    ;GET KEYBOARD CHAR. IN 'A'
1360       CMP #$59      ;IS IT A "Y"
1370       BNE RTN       ;NO,CLR.CARRY AS FAULT FLAG
1380       RTS           ;YES,RETURN WITH CARRY SET
1390  RTN  CLC           ;CLEAR CARRY
1400       RTS           ;AND RETURN
1410 STARTK LDX #$00     ;SET INDEX X FOR START TK #
1420       STX TS2       ;SAVE X INDEX AT TEMP.2
1430       LDA #$B8      ;SET ERROR JUMP TO STARTK
1440       STA ERROR     ;SET ERROR JMP FOR STARTING
```

```
1450          LDA #$02
1460          STA ERROR+1   ;TRACK NUMBER ROUTINE
1470          JSR STROUT    ;PRINT FOLLOWING CHAR.$
1480          .BYTE 'Specify Starting Track:',0
1490 INPUT    JSR OSINP     ;GET KEYBOARD CHARACTER
1500          LDA #$00
1510          STA BUFBYT+1  ;SET BUFFER OFFSET TO 0
1520          JSR BUFBYT    ;LOAD 'A' WITH FIRST CHAR.
1530          CMP #$0D      ;IS THE FIRST CHAR.= CR.
1540          BEQ BADTK     ;YES,THEN ERROR ROUTINE
1550          PHA           ;NO,THEN SAVE THE FIRST BYTE
1560          JSR BUFBYT    ;GET SECOND BYTE
1570          CMP #$0D      ;IS IT A CARRIGE RETURN
1580          BEQ CK1BYT    ;YES,THEN CHECK FIRST BYTE
1590          AND #$0F      ;NO,THEN MASK UPPER 4 BITS
1600          STA TS1       ;AND SAVE IT IN TEMP.
1610          PLA           ;RECOVER FIRST BYTE
1620          CMP #$30      ;IS TRACK # < 0
1630          BMI BADTK     ;YES,GO DO ERROR ROUTINE
1640          CMP #$34      ;IS TRACK # > 39
1650          BPL BADTK     ;YES,GO DO ERROR ROUTINE
1660          ASL A
1670          ASL A         ;MOVE LSB TO MSB (4 BITS)
1680          ASL A
1690          ASL A
1700          ORA TS1       ;COMBINE WITH SECOND BYTE
1710          LDX TS2       ;GET INDEX FOR TRACK # SAVE
1720          STA $02,X     ;SAVE TRACK NUMBER
1730          JSR BUFBYT    ;GET THIRD BYTE
1740          CMP #$0D      ;IS IT A CARRIGE RETURN
1750          BNE BADTK     ;NO,THEN ERROR
1760          RTS
1770 GENDTK   LDX #$01      ;SET UP FOR END TK # SAVE
1780          STX TS2       ;SAVE INDEX X IN TEMP.2
1790          LDA #$15      ;SET ERROR JUMP TO GENDTK
1800          STA ERROR     ;SET ERROR JUMP TO ENDING
1810          LDA #$03
1820          STA ERROR+1   ;TRACK # ROUTINE
1830          JSR STROUT    ;PRINT FOLLOWING STRING
1840          .BYTE $0A,$0A
1850          .BYTE '  Specify Ending Track:',0
1860          JMP INPUT
1870 CK1BYT   PLA           ;RECOVER FIRST BYTE
1880          CMP #$30      ;IS TRACK # < 0
1890          BMI BADTK     ;YES,GO DO ERROR ROUTINE
1900          CMP #$3A      ;IS TRACK # > 9
1910          BPL BADTK     ;YES,GO DO ERROR ROUTINE
1920          AND #$0F
1930          LDX TS2       ;INDEX FOR TRACK # SAVE
1940          STA $02,X     ;SAVE TRACK #
1950          RTS
1960 BADTK    JSR STROUT    ;PRINT FOLLOWING STRING
1970          .BYTE $0A
1980          .BYTE '* * BAD TRACK NUMBER * *',$0D,$0A
1990          .BYTE $0A
2000          .BYTE '   PLEASE TRY AGAIN   ',$0D,$0A,$0A,0
2010          JMP (ERROR) ;ON ERROR GET ADDRESS AND JUMP
2020  ERR     JSR STROUT    ;PRINT FOLLOWING CHAR. $
2030          .BYTE $0A,$0A
2040          .BYTE '* START TRACK # > END TRACK # *'
2050          .BYTE $0A,$0A,$0A,$0D,0
2060          RTS           ;ERROR,RETURN WITH CARRY CLR
2070 CKERR    LDA $03       ;GET END TRACK NUMBER
2080          CMP $02       ;IS START TK # > END TK #
2090          BCC ERR       ;YES,GO DO ERROR ROUTINE
2100          RTS
2110 DSPLY1   JSR STROUT    ;PRINT OUT FOLLOWING MESSAGE
2120          .BYTE '                '
2130          .BYTE 'OS-65D V3.X'
2140          .BYTE $0D,$0A,$0A
2150          .BYTE '                '
2160          .BYTE 'Single Disk Copy Utility'
2170          .BYTE $0D,$0A,$0A,$0A,0
2180          RTS
2190 DSPLY2   JSR STROUT    ;PRINT FOLLOWING MESSAGE
2200          .BYTE $0D,$0A,$0A,$0A
2210          .BYTE '                '
2220          .BYTE '**********************'
2230          .BYTE $0D,$0A
2240          .BYTE '                '
2250          .BYTE 'INSERT MASTER DISK!'
2260          .BYTE $0D,$0A
2270          .BYTE '                '
2280          .BYTE '**********************'
2290          .BYTE $0D,$0A,$0A,$0A
2300          .BYTE '                '
2310          .BYTE 'Are You Ready(Y/N):',0
2320          RTS
2330 DSPLY3   JSR STROUT    ;PRINT OUT FOLLOWING MESSAGE
```

```
2340          .BYTE $0D,$0A,$0A
2350          .BYTE '                          '
2360          .BYTE '**************************'
2370          .BYTE $0D,$0A
2380          .BYTE '                          '
2390          .BYTE 'INSERT DESTINATION DISK!'
2400          .BYTE $0D,$0A
2410          .BYTE '                          '
2420          .BYTE '**************************'
2430          .BYTE $0D,$0A,$0A,$0A
2440          .BYTE '                          '
2450          .BYTE 'Are You Ready(Y/N):',0
2460          RTS
2470 DSPTBL   JSR STROUT   ;DISPLAY TRACK TABLE ON R/W
2480          .BYTE $0D,$0A,$0A
2490          .BYTE ' R/W Track#:',0
2500          LDA SRTTKN   ;GET TRACK #
2510          JSR PRT2HX   ;PRINT AS TWO HEX DIGETS
2520          JSR STROUT
2530          .BYTE ' Sector#:',0
2540          LDA SECTNM   ;GET SECTOR NUMBER
2550          JSR PRT2HX   ;AND PRINT IT
2560          JSR STROUT
2570          .BYTE ' To:$',0
2580          LDA MEMHI    ;GET HI BYTE FOR R/W
2590          JSR PRT2HX   ;PRINT IT
2600          LDA MEMLO    ;GET LO BYTE
2610          JSR PRT2HX   ;AND PRINT IT
2620          JSR STROUT
2630          .BYTE ' FOR:',0
2640          LDA PGCNT    ;GET THE PAGE COUNT
2650          JSR PRT2HX   ;PRINT IT
2660          JSR STROUT
2670          .BYTE ' PAGE(S)',0
2680          RTS
2690 INTTBL   LDY ##$00     ;SET INDEX FOR SAVE TABLE
2700          TYA          ;SET LO BYTE R/W ADDRESS
2710          STA TABLE,Y  ;WRITE MEMLO TO TABLE
2720          STA MEMLO    ;INITIALIZE MEMLO
2730          INY          ;BUMP SAVE TABLE INDEX
2740          LDA ##$09    ;SET HI BYTE R/W ADDRESS
2750          STA TABLE,Y  ;WRITE MEMHI TO TABLE
2760          STA MEMHI    ;INITIALIZE MEMHI
2770          INY          ;BUMP INDEX
2780          STY WRINDX   ;AND SAVE WRITE INDEX
2790          RTS
2800 WTABLE   LDA SRTTKN   ;GET STARTING TK # FOR READ
2810          BEQ RDTRKZ   ;IF TRACK 0 THEN BRANCH
2820          JSR SETTK    ;POSITION HEAD TO TK # IN A
2830          TSX          ;MOVE STACK POINTER TO REG X
2840          STX STKADR   ;AND SAVE IT
2850          JSR LDHEAD   ;LOAD HEAD TO DISK
2860          INX          ;SET X TO 1
2870          STX SECTNM   ;SET SECTOR NUMBER TO 1
2880          JSR SETSCT   ;POSITION FOR SECTOR/SECTNM
2890          LDA ##$00
2900          STA SCTBYP   ;CLEAR SECTORS BYPASSED CNT
2910 NXTSCT   JSR BPSECT   ;BYPASS A SECTOR
2920          LDA SRTTKN   ;GET THIS TRACK NUMBER
2930          PHA          ;SAVE IT ON STACK
2940          LDA SCTNUM   ;GET SECTOR NUMBE
2950          PHA          ;SAVE IT ON STACK
2960          LDA SCTLEN   ;GET PAGE COUNT
2970          PHA          ;SAVE IT ON STACK
2980          BCS NXTSCT   ;MORE SECTORS?-CONTINUE
2990          JSR UNLOAD   ;UNLOAD HEAD FROM DISK
3000          LDX STKADR   ;GET STACK ADDR.FOR INDEX
3010        LDA SCTBYP   ;GET SECTORS BYPASSED CNT.
3020        BEQ NODATE   ;IF 0 THEN NO DATA THIS TK.
3030          BCC DROP     ;IF > 0 THEN CONTINUE
3040 SAVE     JSR STORE    ;STORE SCTLEN/SCTNUM/TK#
3050 DROP     DEC SCTBYP   ;DROP SECTORS BYPASSED CNT
3060          BPL SAVE     ;IF MORE SECTORS,CONTINUE
3070          LDX STKADR   ;RESET STACK ADDRESS
3080          TXS
3090          RTS
3100 NODATA   JSR STROUT   ;PRINT NO DATA MESSAGE
3110          .BYTE $0D,$0A,$0A
3120          .BYTE '      Track#:',0
3130          LDA SRTTKN    ;GET TRACK NUMBER
3140          JSR PRT2HX    ;PRINT IT
3150          JSR STROUT
3160          .BYTE ' Contains No Data',0
3170          JMP DROP
3180 RDTRKZ   STA TABLE,Y  ;WRITE TK# TO TABLE
3190          INY          ;BUMP THE SAVE TABLE INDEX
3200          STY WRINDX   ;AND SAVE IT
3210          JSR HOME     ;HOME HEAD TO TRACK ZERO
```


THE WORKINGS OF ROM BASIC

By Stephen P. Hendrix
Route 8 Box 81E
New Braunfels, TX 78130

This is the first in a series of monthly articles about the code in the ROMs (read-only memories) in the C1P and other OSI systems. Some of it will be of interest to those of you who program only in BASIC, but it will be aimed mainly at those who want to better understand how BASIC interpreter works in order to better blend machine language with BASIC to have the best of both worlds - the speed of machine language with the special functions and input-output capabilities of BASIC. Along the way, you should also pick up some clever programming tricks for machine code and gain a better understanding of some of the tradeoffs involved in the design of this interpreter.

This month's routine is one which I'll call "GETBYTE". Its main function is to get the next byte of a BASIC program or immediate line, and it also does some of the work of determining what type of byte it is passing back to the BASIC interpreter. Nearly every other routine in BASIC calls this routine sooner or later. The fact that it is

```
3220            JSR LDHEAD  ;LOAD HEAD TO DISK
3230            JSR WAITIH  ;WAIT FOR THE INDEX HOLE
3240            JSR RSACIA  ;RESET ACIA & WAIT FOR IH
3250            JSR DSKBYT  ;GET HI BYTE LOAD VECTOR
3260            STA TKZLVH  ;SAVE AT TK 0 LOAD VECTOR HI
3270            JSR DSKBYT  ;GET NEXT BYTE
3280            STA TKZLVL  ;SAVE AT TK 0 LOAD VECTOR LO
3290            JSR DSKBYT  ;GET THIRD BYTE
3300            STA TKZPGC  ;SAVE AT TK 0 PAGE COUNT
3310            STA PGCNT   ;SET UP PAGE COUNT FOR READ
3320            LDY #$00    ;SET INDEX TO 0
3330 GETBYT     JSR DSKBYT  ;READ TRACK TO MEMORY AT
3340            STA (MEMLO),Y ;MEMLO,MEMHI,Y
3350            INY         ;BUMP INDEX
3360            BNE GETBYT  ;SAME PAGE? YES,CONTINUE
3370            INC MEMHI   ;NEXT PAGE  BUMP MEMHI
3380            DEC PGCNT   ;DROP PAGE COUNT INDEX
3390            BNE GETBYT  ;MORE PAGES? YES,CONTINUE
3400            JSR UNLOAD  ;NO,UNLOAD HEAD
3410            LDY WRINDX  ;GET WRITE TABLE INDEX
3420            LDA MEMLO   ;GET NEXT FREE MEM.LOCATION
3430            STA TABLE,Y ;WRITE IT TO SAVE TABLE
3440            INY         ;BUMP THE INDEX
3450            LDA MEMHI   ;GET HI BYTE FREE MEM.
3460            STA TABLE,Y ;WRITE IT
3470            INY         ;BUMP THE INDEX
3480            STY WRINDX  ;AND SAVE IT FOR LATER USE
3490            JSR CLEAR   ;CLEAR SCREEN
3500            JSR STROUT  ;PRINT TRACK 0 MESSAGE
3510            .BYTE '         '
3520            .BYTE 'Track ZERO Read To:$',0
3530 TKZPRT     LDA TABLE+1 ;GET MEMHI
3540            JSR PRT2HX  ;PRINT IT
3550            LDA TABLE   ;GET MEMLO
3560            JSR PRT2HX  ;PRINT IT
3570            JSR STROUT
3580            .BYTE ' For:',0
3590            LDA TKZPGC  ;GET PAGE COUNT
3600            JSR PRT2HX  ;PRINT IT
3610            JSR STROUT
3620            .BYTE ' Pages',$0D,$0A,$0A
3630            .BYTE '         '
3640            .BYTE '    With:$',0
3650            LDA TKZLVH  ;GET HI BYTE LOAD VECTOR
3660            JSR PRT2HX  ;PRINT IT
3670            LDA TKZLVL  ;GET LO BYTE LOAD VECTOR
3680            JSR PRT2HX  ;PRINT IT
3690            JSR STROUT
3700            .BYTE ' As The Load Vector',0
3710            RTS         ;AND RETURN
3720 STORE      LDY WRINDX  ;GET WRITE TABLE INDEX
3730            LDA STACK,X ;GET STARTING TRACK #
3740            STA TABLE,Y ;WRITE IT TO TABLE
3750            INY         ;BUMP SAVE INDEX
3760            DEX         ;DROP STACK INDEX
3770            LDA STACK,X ;GET SECTOR NUMBER
3780            STA TABLE,Y ;WRITE IT TO TABLE
3790            STA SECTNM  ;SET SECTOR # FOR READ
3800            INY         ;BUMP SAVE INDEX
3810            DEX         ;DROP STACK INDEX
3820            LDA STACK,X ;GET SECTOR LENGTH
3830            STA TABLE,Y ;WRITE IT TO TABLE
3840            STA PGCNT   ;SAVE FOR DSPTBL ROUTINE
3850            LDA MEMHI   ;GET MEMORY ADDR.FOR READ
3860            CMP #$32    ;HAVE WE JUMPED OVER THE OS
3870            BPL BACK    ;YES,THEN CONTINUE
3880            LDA STACK,X ;GET SECTOR LENGTH BACK
3890            ADC MEMHI   ;ADD SECTOR LENGTH TO MEMHI
3900            CMP #$22    ;WILL IT OVERWRITE THE OS
3910            BPL INCMEM  ;YES;BUMP MEMHI ABOVE OS
3920  BACK      INY         ;BUMP SAVE INDEX
3930            DEX         ;DROP STACK INDEX
3940            TXA         ;MOVE STACK INDEX TO 'A'
3950            PHA         ;SAVE STACK INDEX
3960            TYA         ;MOVE SAVE INDEX TO 'A'
3970            PHA         ;SAVE SAVE TABLE INDEX
3980            LDA SCTBYP  ;GET SECTORS BYPASSED COUNT
3990            PHA         ;SAVE IT
4000            JSR DSPTBL  ;VIDEO DISPLY TK#'S,ETC
4010            JSR READDK  ;READ THIS SECTOR TO MEMORY
4020            PLA         ;PULL SCTBYP OFF STACK
4030            STA SCTBYP  ;RESTORE SECTORS BYPASSED CNT
4040            PLA         ;PULL TABLE INDEX
4050            TAY         ;RESTORE SAVE TABLE INDEX
4060            PLA         ;PULL STACK INDEX
4070            TAX         ;AND RESTORE IT
4080            LDA MEMLO   ;GET LO BYTE NEXT FREE MEM.
4090            STA TABLE,Y ;WRITE IT TO TABLE
4100            INY         ;BUMP SAVE INDEX
```

copied into RAM in cold start and called there makes it vulnerable to some tinkering, permitting you to change the way almost any part of BASIC works. More on this later.

```
00BC GETBYTE    INC ADDRLO
00BE            BNE REGETBYTE
00C0            INC ADDRHI
00C2 REGETBYTE LDA $XXXX
 ; I refer to xxxx as ADDR;
 ; the low byte as ADDRLO,
 ; and the high byte as
   ADDRHI.
00C5            CMP #$3A
00C7            BCS RETURN
00C9            CMP #$20
00CB            BEQ GETBYTE
00CD            SEC
00CE            SBC #$30
00D0            SEC
00D1            SBC #$D0
00D3 RETURN     RTS
```

ADDR; composed of the two bytes I'm calling ADDRHI and ADDRLO, always points to the character currently being processed. When BASIC calls GETBYTE, the three instructions from $00BC to $00C1 do a two-byte increment of the value in ADDR. This is an example of self-modifying code, which is generally bad practice, but which runs faster in this case than would equivalent code which was not self-modifying. Since the 6502 has no simple indirect addressing mode for data, it would be necessary to save the contents of one of the index registers, load it with zero, load the desired byte using an indexed indirect mode, and then restore the original value of the index register. Since this routine is used so heavily by BASIC, what appears to be a slight speed advantage turns out to be quite significant.

The instruction at REGETBYTE loads the A register with the actual value of the next byte to be interpreted. REGETBYTE is an alternate entry point used, if anything, more heavily than GETBYTE. REGETBYTE picks up the current character without stepping ADDR, but still sets the flags for the type of byte which it returns. The flags indicate conditions which you would expect the interpreter to need to know about: an end-of-statement mark or ASCII digits. The processor's Z flag indicates end-of-statement if it is set, while the C flag is used to indicate an ASCII digit if it is cleared. In order to fully understand this routine, you may need to walk through it with various values, but here's a play-by-play covering the operation of the routine.

```
4110          LDA MEMHI    ;GET HI BYTE NEXT FREE MEM.
4120          STA TABLE,Y  ;WRITE IT TO TABLE
4130          INY          ;BUMP SAVE INDEX
4140          STY WRINDX   ;AND SAVE IN WRITE INDEX HOLD
4150          RTS          ;RETURN TO READ NEXT SECTOR
4160 INCMEM   STY WRINDX   ;SAVE THE WRITE TABLE INDEX
4170          DEY          ;DROP TABLE INDEX
4180          DEY
4190          DEY
4200          LDA #$33     ;FIRST FREE PAGE ABOVE OS
4210          STA MEMHI    ;SET MEMHI ABOVE OS
4220          STA TABLE,Y  ;SAVE NEW ADDR. IN TABLE
4230          LDY WRINDX   ;GET WRITE TABLE INDEX BACK
4240          JMP BACK
4250 RTABLE   LDA TABLE,Y  ;GET LO BYTE ADDR.FOR WRITE
4260          STA MEMLO    ;SET UP MEMLO FOR DISK WRITE
4270          INY          ;BUMP WRITE INDEX
4280          LDA TABLE,Y  ;GET HI BYTE ADDR.FOR WRITE
4290          STA MEMHI    ;SET UP MEMHI FOR DISK WRITE
4300          INY          ;BUMP WRITE INDEX
4310          LDX TABLE,Y  ;GET TRACK# IN X
4320          BEQ TRKZWR   ;IF TRACK 0 THEN BRANCH
4330          STX TS1      ;SAVE TK# IN TEMP. STORE
4340          LDA SRTTKN   ;GET LAST TRACK# IN 'A'
4350          CMP TS1      ;ARE WE ON THE SAME TRACK
4360          TXA          ;NO,SET CARRY FOR STRACK
4370          STA SRTTKN   ;AND SAVE THIS TRACK #
4380          INY          ;BUMP WRITE INDEX
4390          LDA TABLE,Y  ;GET SECTOR # FOR WRITE
4400          STA SECTNM   ;SET UP SECTOR # FOR WRITE
4410          INY          ;BUMP WRITE INDEX
4420          LDA TABLE,Y  ;GET SECTOR LENGTH(PAGE CNT)
4430          STA PGCNT    ;SET UP PAGE COUNT FOR WRITE
4440          INY          ;BUMP WRITE INDEX
4450          STY RDINDX   ;AND SAVE IN READ INDEX HOLD
4460          LDA SRTTKN   ;GET TRACK # IN 'A'
4470          BCC STRACK   ;CARRY CLR.? YES,INZ&SET TK.
4480 WRTSCT   JSR DSPTBL   ;DISPLAY TRACKS AS WRITTEN
4490          JSR DSKWRT   ;WRITE SECTOR TO DISK
4500          RTS          ;AND RETURN
4510 STRACK   JSR SETTK    ;POSITION HEAD TO TK# IN A
4520          JSR INITTK   ;INITIALIZE THIS TRACK
4530          JMP WRTSCT   ;WRITE THIS TRACK TO DISK
4540 TRKZWR   INY          ;BUMP READ INDEX
4550          STY RDINDX   ;AND SAVE IT
4560          LDA TKZPGC   ;GET TRACK ZERO PAGE COUNT
4570          STA PGCNT    ;SET PAGE COUNT INDEX
4580          JSR HOME     ;HOME HEAD TO TRACK 0
4590          JSR LDHEAD   ;LOAD HEAD TO DISK
4600 FINDXH   LDA FLOPIN   ;GET THE DISK STATUS
4610          BPL FINDXH   ;MISSED INDEX HOLE-TRY AGAIN
4620 PASSIH   LDA FLOPIN   ;FOUND THE INDEX HOLE
4630          BMI PASSIH   ;WAIT FOR INDX.HOLE TO PASS
4640          LDA #$FC
4650          AND FLOPOT   ;MASK UPPER SIX BITS
4660          STA FLOPOT   ;TURN ON WRITE&ERASE ENABLE
4670 AGAIN    LDA FLOPIN   ;GET THE DISK STATUS AGAIN
4680          BPL AGAIN    ;WAIT FOR INDEX HOLE AGAIN
4690          LDX #$0A
4700          JSR TENMS    ;WAIT 1 MILLISECOND,THEN
4710          LDX TKZLUH   ;GET THE HI BYTE LOAD VECTOR
4720          JSR DKWTX    ;WRITE IT TO DISK
4730          LDX TKZLUL   ;GET THE LO BYTE LOAD VECTOR
4740          JSR DKWTX    ;WRITE IT
4750          LDX TKZPGC   ;GET THE PAGE COUNT OF TK0
4760          JSR DKWTX    ;WRITE IT
4770          LDY #$00     ;SET WRITE INDEX TO 0
4780 NXTBYT   LDA (MEMLO),Y ;GET DATA BYTE FROM MEMORY
4790          TAX           ;PUT DATA BYTE IN X REG.
4800          JSR DKWTX     ;WRITE X (DATA BYTE)TO DISK
4810          INY           ;BUMP WRITE INDEX
4820          BNE NXTBYT    ;END OF PAGE? NO,CONTINUE
4830          INC MEMHI     ;BUMP HI BYTE PAGE INDEX
4840          DEC PGCNT     ;DROP PAGE COUNT
4850          BNE NXTBYT    ;MORE PAGES? YES,CONTINUE
4860 WAITXH   LDA FLOPIN    ;GET THE DISK STATUS
4870          BMI WAITXH    ;WAIT FOR INDEX HOLE TO PASS
4880          LDA #$83      ;TURN OFF WRITE&ERASE ENABLE
4890          JSR UNLOAD+2 ;UNLOAD HEAD
4900          JSR STROUT    ;PRINT TRACK 0 WRITE MESSAGE
4910          .BYTE '      Track ZERO Written To:$',0
4920          JMP TKZPRT    ;JMP TO PRINT REST OF MESSAGE
4930          ZZZZ=*
!IO ,02
```

★ ★ ★

With the byte in the accumulator, the CMP at $00C5 tests against the ASCII character for a colon (:). If the actual byte is a colon or "higher" (in the ASCII sequence), it jumps immediately to the RTS at $00D3. Note that the C (carry) flag is set, indicating that the byte is not a digit. The ASCII characters for the digits are $30 thru $39, just "below" the $3A for the colon. Also, if the byte is exactly a colon, the Z (zero or equal) flag will be set, indicating the end of a statement.

Next the CMP at $00C9 tests to see if the byte is the ASCII character for a space. This routine will not return a space but rather discards it and gets the next character. The BEQ at $00CB implements this by branching back to GETBYTE, which increments the pointer and falls back into REGETBYTE. This is why BASIC totally ignores spaces, even spaces imbedded within numbers. (The actual characters in a literal string are picked up differently, allowing spaces to appear there).

Now the plot thickens. Before, it was a simple matter of a comparison to set the carry flag if the byte was too high to be a digit. Now, we must also set the carry flag if the byte is too low to be any of the digits. A simple comparison with the value of an ASCII "0" ($30) will leave the flag in the opposite state from the way it was defined above - that is, set will indicate a digit, and cleared will indicate a non-digit. While there are many ways to reverse this, most obvious ways would disturb some registers' contents, requiring some tricky programming to preserve them. The code from $00CD thru $00D2 is a very elegant way of making the flag mean what it is supposed to.

The SEC at $00CD and $00D0 simply prevent the processor from doing a borrow during the subtractions, which would obviously change the results. Notice that subtracting $30 and then subtracting $D0 is equivalent to subtracting $0100, and since we are discarding the borrow from this operation, the net result is that of subtracting $00, except for the flags. Starting with a byte greater than or equal to $30 in the accumulator, the first subtraction leaves a value from $00 thru $CF. The second

subtraction must then necessarily involve a borrow, which leaves the C flag cleared. Starting with a byte less than $30, the first subtraction leaves a byte from $D0 thru $FF, with a borrow which is ignored by the following SEC. Then the second subtraction ($D0 from a value in the range $D0 - $FF) cannot involve a borrow, leaving the C flag set. Ta - dah!! Notice that if the final result is a zero, the Z flag will be set, indicating end-of-statement like the colon above. (BASIC uses a colon to separate statements on the same line but uses a null [$00] to indicate the end of a line, which must also be the end of a statement.)

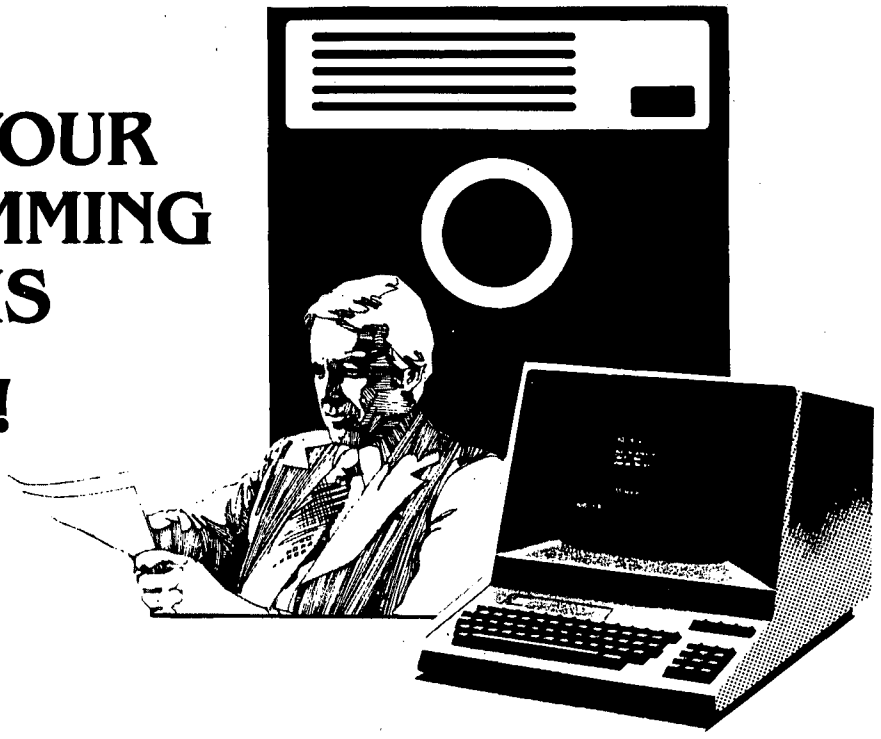
Notice that throughout this routine, great pains are taken not to disturb any registers. Also, the sequence of the tests is optimized for speed. The large majority of the bytes in a BASIC program are greater than the colon, so most of the time the first branch will be taken, saving the time required by the other tests.

Because this routine is called so often by BASIC and because it is run in read-write memory rather than in ROM, it makes BASIC susceptible to some tinkering. For instance, suppose you want to implement an additional command. A very simple way to do this is to make the new command consist of a single character which is not normally used by BASIC, such as #, $, %, &, or '. You could insert the instruction JMP PATCH in place of the CMP #$3A and the first byte of the CMP instruction at $00C5. PATCH could look something like this:

```
PATCH   CMP #<your selected
            character>
        BEQ <to machine code
            executing the
            command>
        CMP #$3A ; to substi-
            tute for the de-
            stroyed instruc-
            tions in REGETBYTE
        BCC NORMAL
        RTS
NORMAL  JMP $00C9 ; to finish
            the other tests
```

Obviously, this would impose a slight speed penalty, since this patched routine is called for every byte of your BASIC program, but this is a tradeoff for the convenience of having the extra command or commands. On the other hand, by implementing the added command in machine language,

TACKLE YOUR PROGRAMMING PROBLEMS HEAD ON!



If operating bugs and "unsolvable" coding problems have got you dodging a big project, here's help . . . Quality software products that'll let you tackle those problems and move on to something easier.

At Software Consultants, we know the kinds of problems you often face with OSI gear, so we designed our software accordingly. Most of our products are modifications and/or extensions of OSI packages, but some are complete replacements. All are easily extensible for your own custom applications. We feel they're the finest OSI software products around—and the most reasonably priced.

PRODUCT ROUNDUP

OS-65D V3.2 DISASSEMBLY MANUAL

A super-complete manual that has it all. 50 pages of disassembly listings, complete and clear comments on all lines, 10 pages of computer generated cross reference listings, and more! Praised by many OSIers who couldn't believe it til they bought one. A deal at $25.95.

REF COMMAND UNDER BASIC

A complete, cross reference utility that'll find and list any BASIC line number, variable, or numeric constant. It's available under 65D or 65U and comes on 8" floppies. This one will save your sanity, and cut out hours of wasted time. Yours for $31.95.

SPOOLER/DESPOOLER UTILITY

A useful utility that feeds backed-up data to your printer for normal output, and leaves your screen free for other work by intercepting data bound for the printer and temporarily storing it on hard disk. Written in super fast machine language. Interfaces with serial and parallel printers. Only $69.95.

FIG FORTH UNDER OS-65U

The only other language available for your 65U system. You get lots of extras, like terminal oriented editor, printer and terminal tools, and much more. Under multi-user , runs BASIC simultaneously too. Unbelievable at $89.95.

VIDEO ROUTINE

This convenient extension for OS65D V1.2 really spices up your video system with little niceties like 24 separate control codes, horizontal and vertical plotting, and many variable screen parameters. Software extensions are available to connect this with the graphics resolution booster. The routine alone is $25.95, with extensions, $29.95.

GRAPHICS RESOLUTION BOOSTER

An ingenious piece of hardware that increases the graphic resolution per character by 8 times. With this device, your circles will be rounder and your curves smoother. A slick addition to your C4/8P system. Priced at $49.95, or, with the video routine and extensions, $79.95.

Unlike the majority of other software vendors, we offer our customers copies of source code (on floppies) for any of our products they've purchased. For a nominal fee of $12, covering postage & handling, we'll send you the source code you choose.

Foreign orders please add $10 postage fee for air mail. Dealer inquiries welcome. Tennessee residents add 6% sales tax. COD orders will incur a $1.50 handling charge. We ship via UPS, so please include street address with your order.

Write or call us today with your order, or ask for our free product catalog and get all the details.

SOFTWARE CONSULTANTS

HARDWORKING *software*

6435 Summer Avenue
Memphis,TN 38134
901/377-3503

you may gain far more speed than you lose by adding this patch.

SUMMARY

GETBYTE -- increments BASIC's current-byte pointer and then falls into REGETBYTE.

REGETBYTE -- loads the current byte into the A register, does not disturb any other registers and returns flags to indicate the type of byte:

End of statement ($00 or $3A)- Z flag and C flag both set.

ASCII digit ($30 - $39) - Z flag and C flag cleared.

ASCII space - gets the next character and sets the flags accordingly.

Any other character - Z flags cleared and C flag set.

The N and V flags are both affected by this routine but have no particular significance to BASIC.

The original copy of this routine is in ROM starting at $BCEE.

★ ★

OSI/MPI DATA SELECTOR

By: Kenneth D. Koonsman
2325 Bel Air
Abilene, TX 79603

You seem to have a lot of computerists writing that their non-OSI purchased MPI-B51 drives have no data separator, and they are unable to buy one anywhere!

Well, Micro Peripherals Inc., 9754 Deering Ave., Chatsworth, CA 91311 sells data separator, P/N 29002-001 for $25.00, and answers their mail promptly.

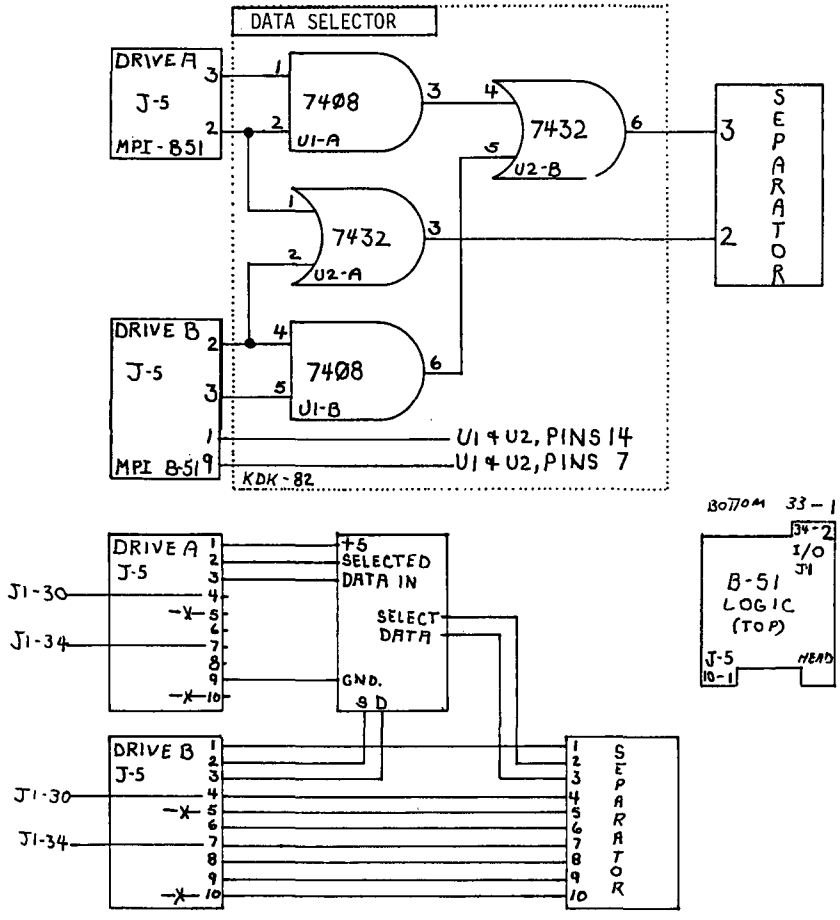
However, there is another way! In order to aid my fellow computerists, and satisfy my money grubbing greed, (I hear you pay for construction articles) I enclose schematics and information for a 'data selector', to couple two MPI-B51/52 drives to a single MPI data separator. The perf board, two chips and two sockets cost about a dollar, skip the sockets if the budget is really tight! This data selector has been built, tested, and is operating nicely in my system (yes, tis true, I couldn't find a second separator either). I have no plans to market this original design in any form, except in this offer to PEEK (65). Publish if you like, pay what you will!

In the pages of PEEK, there continues the question of 'why OSI didn't write software to load heads on the mini-floppies! Well, they did write it, they just didn't use it hardware wise! The 'Superscreen Headload Kit' advertised in the February PEEK (65), makes use of this software and applies the required logic to load heads on the selected drive. Motor control may also be possible!

★

OSI/MPI DATA SELECTOR

DUAL MPI B-51 DRIVES WITH A SINGLE DATA SEPARATOR



NOTES:

1. Layout is not critical, 1¼ sq. 1/10" perf board works nicely, with soldertail sockets, point to point wiring and 8" wires. Tape with Mylar or masking tape to insulate, or mount to the back of the separator with double-back foam rubber.
2. Open the trace to J5 pin 5, on both drives.
3. Open the trace to J5 pin 10, on both drives.
4. J5 pin 1 is +5 volts, to U1 & U2 pin 14.
5. J5 pin 9 is ground, to U1 & U2 pin 7.
6. Separator can be installed in either drive.
7. Open the trace to J1 pin 30, connect J1 pin 30 to J5 pin 4.
8. Open the trace to J1 pin 34, connect J1 pin 34 to J5 pin 7.

★

LETTERS

ED:

The telephone calls in response to my February article convinced me that some people really don't like LEVEL3's automatic top-of-form printer paging. PRINT#5! seems to move top-of-form by one line, so I use POKE 2683,0:PRINT #5!:POKE 2683,10. To disable the automatic top-of-form:

1. LOAD"LEVEL3

2. Verify that:
   PEEK(25811) = 173
   PEEK(25812) = 36
   PEEK(25813) = 62

3. POKE 25811,76:POKE 25812,229:POKE 25813,220

4. SAVE

One reader was interested in keyboard editing of a string which has already been entered, possibly for a word processing application. The listing below, which should be self-explanatory, fools the 02/80 version of EDITOR into doing just that.

```
1 REM SAMPLE ROUTINE FOR
  EDITING A PRE-EXISTING
  STRING
2 REM
3 REM NOTE: The 02/80 OSI
  EDITOR must be enabled.
5 REM
10 CLEAR:PRINT:PRINT
20 POKE 2888,0: REM allow null
   INPUT
30 POKE 2972,13: REM allow :
   in INPUT
40 POKE 2976,13: REM allow ,
   in INPUT
50 POKE 8778,140:POKE 8779,93:
   REM point USR to 23948
60 REM
70 S1$="   1 2 3 4"
80 S1$=S1$+"5 6 7"
90 REM
100 S2$="....:....0....:....0
    ....:....0....:....0"
110 S2$=S2$+"....:....0....:
    ....0....:....0."
120 REM
130 Z$="THIS IS THE STRING OF
    UP TO 71 CHARACTERS TO BE
    EDITED."
140 REM
150 LZ=LEN(Z$): REM length of
    string
160 POKE 23717,LZ: REM tell
    EDITOR the length
170 PRINT S1$:PRINT S2$:PRINT
    Z$; : REM display string
    with scale
180 B=27: REM INPUT buffer
    address
190 REM
200 REM POKE string into
    buffer
210 IF LZ>0 THEN FOR Z=1 TO
    LZ:POKE B,ASC(MID$(Z$,Z,1)
    ):B=B+1:NEXTZ
220 REM
230 POKE B,13: REM <CR> in
    buffer after string
240 REM
250  X=USR(X): REM let them
EDIT
    string
260 REM
270 Z$="":LZ=PEEK(23717): REM
    recover EDITed string
280 IF LZ>0 THEN FOR Z=27 TO
    26+LZ:Z$=Z$+CHR$(PEEK(Z)):
    NEXTZ
290 REM
300 PRINT:PRINT:PRINT Z$: REM
    PRINT edited string
```

Ron Mosley
Englewood, CO 80110

* * * * *

ED:

This letter is in response to your call for information about software we have tried and liked. The specific item is the terminal driver program written by Larry Hinsley of Software Consultants. I bought a copy from PEEK (65) a year or so ago, and liked it so much that I have started selling it and require any potential clients to purchase a copy before I will do any programming for them. Since the program easily pays for itself in savings of programming time, that isn't much of a problem.

The program consists primarily of a modified BEXEC*, which pokes all of the program's goodies into the operating system. These consist of mnemonics for the terminal functions (clear screen, erase line, etc.) and a routine which allows direct cursor addressing by means of a PRINT @ statement. The program also does away with BASIC's habit of removing leading spaces from data retrieved from a disk. That can be disconcerting at first, but can also be very valuable.

Since the changes are made in BEXEC*, all programs written using this driver become terminal independent. The terminal dependence is all in BEXEC*. I have several of these rigged up, one for each of my terminals, and once I have run the proper one, the programs don't care which terminal I use.

Another feature of the program is a way of limiting the length of an entry typed from the terminal. If the operator tries to exceed that length, the bell sounds and the computer refuses to accept the extra characters. There is a subroutine in a sample program which allows the acceptance of a single character entry from the keyboard without hitting return, and another which fills the function of a PRINT USING command.

As a sample of the way the program works, take the following line:

```
PRINT'CF'@(20,12)'SB';"Which
do you wish?";'SF':GOSUB 61000
```

The mnemonic 'CF' erases all full intensity characters on the screen, the @(20,12) moves the cursor to the 20th horizontal column in the 12th row down. 'SB' is the mnemonic for start background, all following characters will be printed in half intensity. The text is printed and then the mnemonic 'SF' switches the terminal back to printing in full intensity. The subroutine reference is to that subroutine which allows single character entry without a carriage return. It should be noted that the question mark in the text is necessary if it is to appear at all. One of the other features is to remove the prompt for input statements, and I was overjoyed to have that available.

The line above also demonstrates the only trouble that I have had with the program: the 'CF'. For a while, if I used that mnemonic just prior to writing a screen full of data from the disk, the first few fields would be missing. It turned out that clearing foreground characters is a rather slow function. Now, whenever I use that mnemonic, I immediately call a subroutine which sends 75 nulls to the terminal. That has completely cured the problem.

To sum up, I think that this is one of the best tools I have come across. It makes error checking for length unnecessary and has eliminated scrolling from all of my programs which speeds things up a lot. There are no messy leftovers on the screen, and the programs are a good bit shorter in the data entry sections. I can think of some worthwhile extensions, and have written some, but I have absolutely no complaints about the program. Anyone writing in 65-U should have something like this.

Loren Weaver
Elgin, IL 60120

* * * * *

Business System Users

Get Mainframe Software features with
"Computer Business Software"
"CBS"

the **INSTALLABLE** Accounting Package
for Floppy Disk and Hard Disk Systems -
Single or Multi User

Featuring: Accounts Receivable with Sales Analysis
Inventory with Product Analysis
Order Entry with Invoicing
Accounts Payable with Checkwriting
General Ledger with Financial Statements
Payroll with Labor Distribution

- Thoroughly Documented
- Already Installed
- Fully Supported
- Ready Now

Call **1-800-843-9838** for your free documentation kit, price schedule and sales brochure.

MICROSOFTWARE INTERNATIONAL

3300 South Madelyn, Sioux Falls, South Dakota 57106
"Mainframe Software for your Micro"

ED:

SooperSpooler is a piece of hardware that is interfaced between your CPU (550 or 555 board) and your parallel or serial printer, or between your Centronics parallel interface and your serial or parallel printers.

I cannot report on all aspects of this unit yet, since I haven't tested all configurations yet, but the way it is currently hooked up and configured it's well worth its price to the medium - heavy user. It is hooked up between my 550 board (soon to be replaced by a 555 board) on the one hand and a serial C.Itoh word processing printer (Model FP-1500, 25 cps) on the other.

SooperSpooler is an independent unit with its own power supply and ROM/RAM memory. The basic unit costs $349.00 for parallel throughput, with 16K bytes of RAM; I also have a serial board in it for an additional $95.00. An additional board is available to increase the memory to 62K bytes, which I may well need, eventually. 2K bytes of ROM contains the operating system. It uses a Z80 CPU. Data input/output, handshaking and the "intelligent" features are contained in a 2K x 8 bit (2048 byte) type 2716 EPROM.

I have my SooperSpooler currently configured for RS232C, and of course, I had to make some changes to OS65U and OS DMS to output to DV8 or DV5 via pokes to the 550. This was no problem with one or two exceptions. I've tried to configure it directly out of the terminal (Televideo 910 with Hazeltine 1410 emulation) but that did not work well with the software, however, you can't beat this hookup for debugging terminal and other problems. I am mainly using SooperSpooler for word processing and in that aspect it works admirably with the Itoh.

How nice it is to hear the printer happily clattering away, while the CPU is released from the time consuming wait (until the printer is done) because now SooperSpooler is storing the data to be printed, instead.

The intelligent features are numerous, not the least of which is the ability by means of CHR$ instructions to reset the spooler to accept parallel input into serial output (although I haven't fully tried this yet). Unfortunately, there is a problem doing it the other way, i.e. from serial input to Centronics Parallel output. As far as I can determine with my Centronics 737, it is not a handshaking problem, but instead, the output from the spooler is 7-bit and the Centronics 737 is 8-bit, but even that I am not sure of. The printed text garbles consistently, i.e. 50 lines of "The quick brown fox" will garble at the same spots in all of the lines.

Other intelligent features include a hard and soft reset, space compression (great with columnar material), pagination, headers, page numbering, ability to have printer accept single sheets of paper, page formatting, line formatting (margins), change configuration, re-defining the lead-in character, etc. The documentation is very good and there is even a set-up program included in Microsoft Basic that needed only a few changes to match OSI's basic, and that now runs AOK. All pinouts are described and all timing patterns. It is easy to by-pass the spooler, should this become necessary, because all inputs are one gender and the output connectors are the opposite gender (so they mate). Additional cables are not included (I put my own together). There are a series of configuration switches on the back of the unit (some items are software controllable as well) for handshaking, output, input, baud rate, parity, character length, etc. The handshaking can be either ETX/ACK serial or XON/XOFF and hardware serial handshaking. The front panel has a 1 inch LED which shows the amount of Kbytes in use (VERY handy), a power switch, a soft reset, space suppression and pagination test switches. A self test can be done with those switches.

F.S. Schaeffer
Jamaica, NY 11435

* * * * *

ED:

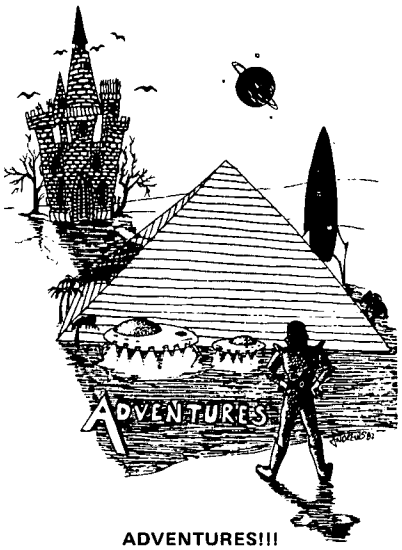
The comments by Richard L. Trethewey in the May 1982 issue of PEEK (65) concerning OS-65D V3.3 are mostly palatable, and I agree that it is a very good operating system. I personally prefer the output speed of the Software Consultants video routines as patched into V3.2. That scrolls very fast, with none of the V3.3 jerkiness, gives screen clear, home, etc. Along with a keyboard routine from Arnie Ames of the Rockford, Illinois OSI User Group which behaves as it should with none of the stupid problems of the OSI polled keyboard ROM which are almost corrected in V3.3. This gives me a terrific operating system.

However, Mr. Trethewey, in his second paragraph on page 16 gives some corrections to locations in the Polled Keyboard Input Routine which, if implemented will hang your system. His efforts point out the GREAT need for all of us who work with V3.2 or V3.3 to have a copy of the BIBLE, namely Software Consultants' V3.2 Disassembly Manual. I have disassembled all of V3.3 and notated it where it is different from their disassembly. In particular, Mr. Trethewey is discussing the Routine named above which is on page 15 of the BIBLE. He is correct in wanting to avoid JSR-ing to the SWAP4 routine at $2644, but replacing the JSR with a JMP will take you back to the DO I/O routine (BIBLE p.8) (returning at $2359) without even getting to the V3.3 Polled Keyboard Routine, called at $2531. What must be done is to fill locations $252B, $252C and $252D with NOP, ie. with $EA or D234. The RTS at $2539 with a $60 is correct, and things will work. He is correct in the two changes at $2532 and $2533 because it makes no sense to JSR to a JMP, but don't bother with the change at $363C - it only serves to make the blinking cursor distracting.

The end result of his suggestions, as corrected, is that the 23 bytes from $2644 thru $265A are now free for re-use, unless BASIC calls this routine. If it does, then those JSR's should similarly be filled with $EA.

Lastly, on page 17 of the April issue, the editor asks for someone to write about the DAC I & II. I have worked with those extensively and will soon write my findings concerning the wave-shape tables, the bad POKE in the OSI program which prevents anything other than a square wave, etc. The programs can be put on V3.3, and I want to change the file that stores tune names to a Random instead of Sequential so that one can take advantage of the reduced number of disk accesses in V3.3, or one could use the disk checking subroutine in V3.2. In either case I will

QUALITY SOFTWARE FOR TRS-80 COLOR AND OSI



ADVENTURES!!!
For TRS-80 COLOR and OSI. These Adventures are written in BASIC, are full featured, fast action, full plotted adventures that take 30-50 hours to play. (Adventures are inter-active fantasies. It's like reading a book except that you are the main character as you give the computer commands like "Look in the Coffin" and "Light the torch.")
Adventures require 16K on COLOR-80 and TRS-80. They sell for $14.95 each.

ESCAPE FROM MARS
(by Rodger Olsen)
This ADVENTURE takes place on the RED PLANET. You'll have to explore a Martian city and deal with possibly hostile aliens to survive this one. A good first adventure.

PYRAMID (by Rodger Olsen)
This is our most challenging ADVENTURE. It is a treasure hunt in a pyramid full of problems. Exciting and tough!

TREK ADVENTURE (by Bob Retelle)
This one takes place aboard a familiar starship. The crew has left for good reasons – but they forgot to take you, and now you are in deep trouble.

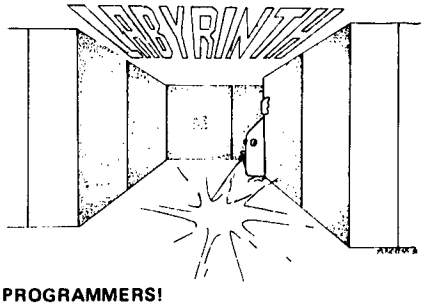
NEW!!
CIRCLE WORLD – We got Kzinti and puppeteers and problems. Our newest and biggest adventure. Requires 12k on OSI and 16K on TRS-80 Color.

NUCLEAR SUB – You are trapped in a nuclear sub at the bottom of the ocean. Escape and even survival is in doubt. Plotted by three of the most sadistic – I mean "creative" minds in adventure programming.

VENTURER!–A fast action all machine code Arcade game that feels like an adventure. Go berserk as you sneak past the DREADED HALL MONSTERS to gather treasure in room after room, killing the NASTIES as you go. Great color, high res graphics, sound and Joystick game for the TRS-80 Color or OSI machines. (black and white and silent on OSI.) Tape only. $19.95.

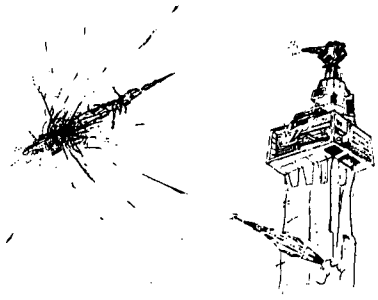
BASIC THAT ZOOOMMS!!
AT LAST AN AFFORDABLE COMPILER FOR OSI AND TRS-80 COLOR MACHINES!!! The compiler allows you to write your programs in easy BASIC and then automatically generates a machine code equivalent that runs 50 to 150 times faster.
It does have some limitations. It takes at least 8K of RAM to run the compiler and it does only support a subset of BASIC–about 20 commands including FOR, NEXT, END, GOSUB, GOTO, RETURN, END, PRINT, STOP, USR(X), PEEK, POKE, *, /, +, -, X, X, =, VARIABLE NAMES A-Z, A SUBSCRIPTED VARIABLE, and INTEGER NUMBERS FROM 0 - 64K.
TINY COMPILER is written in BASIC. It generates native, relocatable 6502 or 6809 code. It comes with a 20 page manual and can be modified or augmented by the user. $24.95 on tape or disk for OSI or TRS-80 Color.

LABYRINTH – 16K EXTENDED COLOR BASIC – With amazing 3D graphics, you fight your way through a maze facing real time monsters. The graphics are real enough to cause claustrophobia. The most realistic game that I have ever seen on either system. $14.95. (8K on OSI)



PROGRAMMERS!
SEE YOUR PROGRAM IN THIS SPACE!! Aardvark traditionally pays the highest commissions in the industry and gives programs the widest possible coverage. Quality is the keyword. If your program is good and you want it presented by the best, send it to Aardvark.

QUEST – A NEW IDEA IN ADVENTURE GAMES! Different from all the others. Quest is played on a computer generated map of Alesia. Your job is to gather men and supplies by combat, bargaining, exploration of ruins and temples and outright banditry. When your force is strong enough, you attack the Citadel of Moorlock in a life or death battle to the finish. Playable in 2 to 5 hours, this one is different every time. 16K COLOR-80 OR TRS-80. ONLY $14.95.



SPACE ZAPPER – Protect your central Star Base from ships that attack from all four sides. Fast reflexes are required as the action speeds up. Great for kids or Dads. This game has high speed high resolution graphics and looks as if it just stepped out of the arcades. – 16K extended or 32K disk. BASIC TRS-80. Color only. $14.95.

Please specify system on all orders

This is only a partial listing of what we have to offer. We have arcade and thinking games, utilities and business programs for the OS1 and TRS-80 Color. We add new programs every week. Send $1.00 for our complete catalog.

AARDVARK - 80
TRS 80 COLOR **2352 S. Commerce, Walled Lake, MI 48088** **OSI**
(313) 669-3110

OSI COMPATIBLE PRODUCTS

56K 2-MHz Ultra Low Power CMOS Static Memory Board **MEM-56K $850**

Partially Populated Boards (Specify address locations required) ... **MEM-48K $750**
MEM Board uses the new 2K-Byte Wide Static RAM chips which are **MEM-32K $550**
2716 EPROM compatible. Any 2K byte memory segment can be **MEM-24K $450**
populated with RAM or EPROM (or left empty for use of Address Space **MEM-16K $350**
by another board). Fully expandable to any memory size you will ever **MEM- 8K $250**
need. No special addressing requirements, just solder in extra sockets **MEM- 4K $200**
Extra 2K RAM Memory Chip **$24**
Optional Parallel Printer Port **-P $120**
Optional Calendar/Clock Software available in EPROM) **-T $ 25**
Both options (Disk software mods provided for use of 6522 VIA on printer). **-PT $125**

EXAMPLE USES:

C4P & C8P: Expansion to **4K RAM** of Basic workspace.
Parallel Printer Port — Reserve Serial Port for MODEM
Calendar/Clock Displaying on unused portion of screen.
Space for 5.75K of **Enhanced System Monitor EPROMS.**

All of this on 1 Board, using only one of your precious slots. Software for Enhanced System Monitor capabilities is continuously being developed and improved. As new EPROM Monitors are available, you may upgrade to them for any price differential plus a nominal $10 exchange fee. Another possibility is to fill any portion of the memory with Basic Programs in EPROM for **Power-on Instant Action.** This custom EPROM programming service is available at $25 per 2716 (Includes EPROM). Extra copies at $15 for each EPROM.

C4P-MF & C8P-DF:
Memory expansion to **48K.**
Add 6K **Memory** above BASIC for special software requirements.
Parallel Printer Interface and/or **Displaying Calendar/Clock.**
Add 1.75 K **Enhanced System Monitor ROM.**

C3: Up to **56K of Memory** Expansion — can be addressed for **Multiuser.**
(Optionally, each user can have his own **Dedicated Printer Port).**

C1P,C4P & C8P FLOPPY DISC CONVERSIONS:
Memory/Floppy Board (Includes MI48P1 ROM) ..... **MEM F-16K $450**
C1P-600 Board Adapter & Cable ..... **A600/48 $ 50**
Additional Memory/Printer/Times ..... **(See MEM Board Prices)**
5¼" Drive/Case/Power Supply & Cable to MEMF Board **FD5 $399**

IEE-488 INTERFACES AND SOFTWARE:

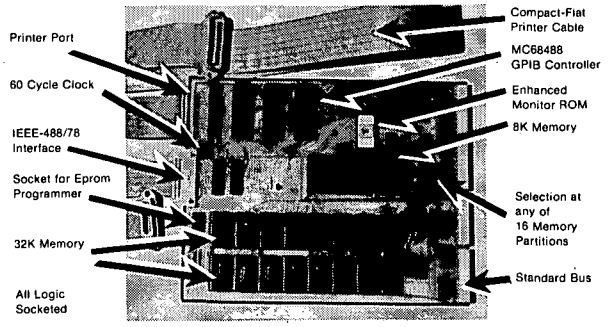
The General Purpose Instrumentation Bus (GPIB Controller interface is available for all OSI Computers. Machine code GPIB Drivers are linked to Basic to provide easy control of IEEE-488 instruments which is equal to the best of Hewlett-Packard Controllers and far superior to most others. Basic Commands for Serial Poll, Parallel Poll, IFC Clear, full Local/Remote Control, Respond to SRQ Interrupts, Send Trigger, do Formatted Input/Output, Direct Memory Input/Output and MORE. Interface includes IEEE-488 Ribbon Cable/Connector.

GPIB Controller Interface for C2, C3, C4 and C8 Systems ..... **GPIB 4-488 $395**
GPIB Software for **OS-65D** (Add -8 for 8" or -5 for 5") ..... **GPIB 488-D $ 70**
GPIB Software for **OS-65U** ..... **GPIB 488-U $100**
GPIB Software on two 2716 EPROMS for **ROM** based systems ... **GPIB 488-R $100**
Add Optional **Parallel Printer Interface** to GPIB 4-488 ..... **-P $120**
Add Optional **Calendar/Clock** to GPIB 4-488 ..... **-T $ 25**
Add **2K RAM** to GPIB 4-488 (Specify location, $4000-$BFFF & $D000-$EFFF available) **-M $25**
GPIB Controller for C1P, Includes Software, Clock, All Features of ROMTERMS, & space for 6K EPROM ..... **GPIB 6-488R $395**
Add Optional Parallel Printer Interface to GPIB 6-488R ..... **-P $120**

EPROMS:
C1P ROM with 24/48 Col Display for Series II, Smart Terminal, Line Editing, Corrected Keyboard Screen Clear and More ..... **ROM-TERM II $59.95**
C1P ROM with 24 Col Display. Other ROM-TERM II Features, Disk Boot, and ROM/Disc Basic Interchange ..... **ROM-TERM $59.95**
C4P-MF/C8P-DF Disk warm start, changed IRQ Vector and just flip switch for Serial or Video System with Corrected Keyboard ..... **SYNKEY $39.95**

ENHANCED MONITOR ROMS FOR USE ON GPIB 4-488 & MEM BOARDS:
Expanded Support for **C4P & C8P** Featuring Calendar/Clock, Line Edit, Smart Terminal, Memory Files, Parallel Printer Control, Corrected Keyboard, All Features of ROMTERMS, **Disk Support** with Warm Start and More **MI48P1 $59.95**
Expanded C2 Monitor with Calendar/Clock Software, Hard Disk Boot, Warm Start and Multi-User Control for C2 Systems **MIC2-1 $59.95**

IEEE-488 CONTROLLER INTERFACE



THE GPIB 4-448 INTERFACE BOARD CONVERTS ANY OSI COMPUTER INTO AN IEEE-488 INSTRUMENT BUS CONTROLLER!

BENEFITS — Provides a Sophisticated Instrumentation Controller at very low cost (often saving thousands of Dollars). The combination of IEEE-488 Instrumentation Controller and High Capacity Hard Disk file storage available on OSI Computer systems is available at a fraction of the cost required by the nearest competitor. The IEEE-488 Bus, also known as the GPIB,HP-IB or IEC-625 is the most popular International Standard for connecting instrumentation systems. This 16-line bus is designed to interconnect and control up to 15 instruments at a time. Currently, over 2000 different instruments are available to work on this bus. They include: Plotters, Digitizers, Printers, Graphic Displays, Recorders and a multitude of specialized Test/MeasurementControl Equipment.

EPROM-ABLE — Can be used with a C4-P to create a dedicated IEEE-488 controller.

C2-D MULTIPLE USER SYSTEMS

SAVE — 2 and 3 user Time Sharing Systems are available on the C2-D Winchester Disk Computer at a considerable cost savings from C3 Multiple User Systems. The 3 user C2-D System can be expanded to include a word processing printer, 4 other parallel printers and 3 serial printer interfaces.

COMPATABLE — The special C2-D Multi-User Executive Program is 100% compatable with OS-65U V1.4. The Multi-User Real Time Clock, Memory Partition Control and IRQ Interrupt Management are done on the Micro Interface Memory Board. Thus, the CPU board is not modified and remains in factory condition.

CONVERSIONS — The Up-Grade of your existing C2-D Computer to Multiple User Configuration is also available. Call for details.

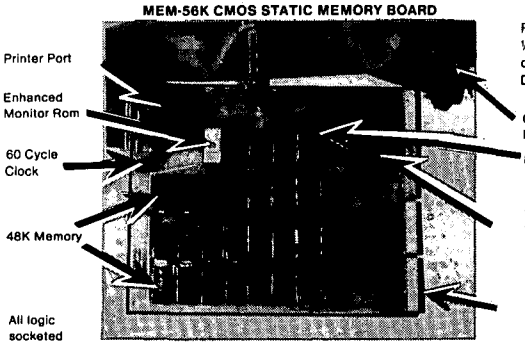
FLOPPY DISK UPGRADES FOR C1P, C4P & C8P

Our Memory/Floppy Board provides easy conversion of 502 and 600 CPU Computers to Floppy Disk Operation. The **MEMF** Board has a floppy disk interface which includes a data separator and the ability to automatically lift the disk drive heads — your floppy disk lifetime will be extended many times. You will retain the cassette interface for your existing software; which can easily be converted to Disk.

This **MEMF-16K** Board is populated with **16K RAM** (50K possible) and has features of the MEM CMOS Static Memory Board with an added floppy interface. The low power memory means extra power supply not required. **ROM Basic** is retained even when Board is populated for 48K Disc Basic. An optional Parallel Printer Port and Real Time Calendar/Clock is on board.

Complete Ready to Run conversion kits with 5¼" or 8" Disk Drives are available.

MEM-56K CMOS STATIC MEMORY BOARD



ULTRA-LOW POWER — By using CMOS Static RAM Memory, the total power consumption is about ½ Amp at 5 Volts when populated for 48K. In fact, most of power is used by the Address Line Buffers and the Data Transceivers.

MULTI-USER — Can be addressed for any of the 16 multi-user memory partitions. The low power and single memory board/partition simplify installation and provide a typical $1400 saving for a 3-user system.

MICRO-INTERFACE
3111 SO. VALLEY VIEW BLVD., SUITE I-101
LAS VEGAS, NEVADA 89102
Telephone: (702) 871-3263

Check with your local Dealer or Order Direct.
Phone orders accepted.
TERMS: Check/Money Order/Master Charge/VISA
Sent POSTPAID ON PREPAID ORDERS.
Foreign Orders: Prepaid only.
Add 5% for handling/shipping.

do that soon, then write about DAC I & II.

Paul Rainy
Villa Park, IL.

* * * * *

ED:

Re: Letter to the Editor from F. Schaeffer, PEEK (65) April '82.

Thanks! While you gave us a rather back-handed compliment, it was a compliment nonetheless.

I recall talking to Mr. Schaeffer re: problems with WP6502 V1.3 getting along with the Hazeltine 1420. Seeing the letter and reading about the whole situation, I now know the problem and solution.

WP6502 V1.3 comes with the default "line feed indicator" set to the tilda or character #126. This is, of course, the usual command lead-in for the 1420. Problems are usually circumvented by moving the Hazeltine's "ESC/~" switch to the "ESC" position. However, then enters the Centronics printer and Catch - 22.

Centronics commands usually use the ESC as a lead-in. Therefore, no matter where the Hazeltine switch is set, if you send commands to the Centronics, you will eventually get the Hazeltine to go crazy. The solution would have been to change the "line feed indicator" to #126 in the WP6502 Install program and to set the switch to the "~" position.

As to the proportional Centronics 737 justification, the Operations Manual states that we do not support proportional justification on this class of printer.

It is true that if you select NEC printer and then switch devices to the Centronics device number that you can get justification. This is because if proportional justification sees a printer device that it knows is not a NEC or Parallel-Diablo type, it simply justifies via TTY-logic.

We do intend to support 737 proportional logic (and all others we know of) by Fall '82.

Meanwhile, we have released V1.3a which gets around the "line feed indicator" problem entirely. Version 1.3a also adds a File Clerk to 65U and works with 65U Version 1.42 as well as all earlier versions. 1.3a also allows character translation between terminal and printer to facilitate foreign language word processing among other things. Attacked are also other requests from users such as three digit margins and character-by-number commands. Page numbers can be at the top or bottom and centered or at the right or left. Hopefully, you will see a full review of V1.3a in a forthcoming PEEK (65).

In the meantime, you will be receiving a WP6502 V1.3a with our compliments.

Fred Beyer
DQFLS, New York, NY.

* * * * *

ED:

Thanks for publishing my letter in the May issue. I noticed that I made a mistake in it though. The POKE to $252B should have been a $2C and not a $4C. I apologize for any trouble this may have caused.

The May issue also contained a letter from a Mr. David L. Kuhn. He was asking about the NULL command under OS-65D V3.3. While the keyword "NULL" was indeed replaced by the keyword "EDIT" under 3.3, the NULL function remains intact. To execute nulls, simply POKE location 21 ($15) with the number of nulls required for the application. This value is preset to 0 upon cold starting BASIC and the POKE is valid for all versions of OS-65D. I hope this helps. Thanks again.

Richard L. Trethewey
Minneapolis, MN

* * * * *

ED:

Let us share with you some of the 'quirks' of OS-65U, V1.3 (September 1981 release):

1. The largest number is now slightly less than 2^32 (4. 294 967 29 E09) which makes it almost useless for scientists, mathematicians and engineers. It was bad enough when 65U-V1.2 only went to 2^127. And you have to set Flag 30 to get an overflow error message.

2. Several Utility programs have errors introduced when someone tried to avoid using exponents because exponents are overlayed with INP$. They tried to do mathematically incorrect powers of 10.

3. RENAME, as in 65U-V1.2, allows one to insert a duplicate file name.

4. The File Protect attributes in both versions do not seem to follow the reported scheme.

5. Be aware that numbers are truncated by PRINT L$,A or PRINT R$,A. That is, the numbers are NOT rounded.

6. Reportedly, SWAP and PACK (which is more appropriately called FILL) are still supported by 65U-V1.3 if COMKILL is not activated. Fill may be used to add leading left spaces to strings being PRINTED to file.

7. It is not possible to PRINT#DV when DV=17, or 129, etc-that is to both screen and printer. You may POKE 11686,17 however.

8. In WP2, when a file is deleted, then that track is reused, be aware that GARBAGE, i.e., the old file info, is still there!

9. WP3.2 is oriented strictly to terminals that OSI sells! This cost us 10% restocking to learn!

We hope this will save someone some of the bewilderment we have experienced.

We would like someone to explain the significance of File TYPE. Just what limits are on a DATA file as opposed to a BASIC file? Or is this just to help a user identify his general file usage?

Charles E. Muhleman
Marion, IN 46952

Charles:

Your assumption is correct as to File TYPE, 65U cares not. Re paragraph #9. WP3.2 can be configured for other terminals.

Dick McGuire
Tech. Ed.

* * * * *

Editors Note

In response to letters to PEEK concerning the lack of response from Modular Systems' DiskDoubler, we talked to Rich Edwards who advises that problems were encountered when using certain versions of OSI op systems which have now been fixed. Data sheets are being printed and should have reached you by this time. Product shipment should begin in mid July.

PEEK (65)
The Unofficial OSI Users Journal

P.O. Box 347
Owings Mills, Md. 21117

BULK RATE
U.S. POSTAGE
PAID
Owings Mills, MD
PERMIT NO. 18

DELIVER TO:

PEEK (65)
P.O. BOX 347
OWINGS MILLS, MD. 21117

SUBSCRIPTION RATES FOR 12 ISSUES (ONE YEAR), EFFECTIVE WITH THE JULY, 1981 ISSUE. ALL RATES QUOTED IN U.S. DOLLARS.

PLEASE FILL OUT AND RETURN WITH CHECK OR MONEY ORDER.

( ) $15.00 ENCLOSED. U.S. (MARYLAND RESIDENTS ADD 5% SALES TAX.)
( ) $23.00 ENCLOSED. CANADA AND MEXICO, 1ST CLASS, SURFACE.
( ) $35.00 ENCLOSED. SOUTH AND CENTRAL AMERICA. AIR MAIL.
( ) $35.00 ENCLOSED. EUROPE. AIR MAIL.
( ) $40.00 ENCLOSED. ALL OTHER. AIR MAIL.

NAME _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ STREET _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

CITY _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ STATE _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

ZIP CODE _ _ _ _ _ _ _ _ _ _ _ _ _ COUNTRY _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

PLEASE SEND THE FOLLOWING BACK ISSUES. I ENCLOSE:

( ) $2.00 EA. U.S. SURFACE. (MARYLAND RESIDENTS ADD 5% SALES TAX.)
( ) $2.50 EA. CANADA AND MEXICO, SURFACE.
( ) $3.00 EA. SOUTH AND CENTRAL AMERICA. SURFACE.
( ) $3.00 EA. EUROPE. SURFACE.
( ) $3.50 EA. ALL OTHER. SURFACE.

VOL 1. 1980

( ) JAN #1	( ) FEB #2	( ) MAR #3	( ) APR #4
( ) MAY #5	( ) JUN #6	( ) JUL #7	( ) AUG #8
( ) SEP #9	( ) OCT #10	( ) NOV #11	( ) DEC #12

VOL 2, 1981

( ) JAN #1	( ) FEB #2	( ) MAR #3	( ) APR #4
( ) MAY #5	( ) JUN #6	( ) JUL #7	( ) AUG #8
( ) SEP #9	( ) OCT #10	( ) NOV #11	( ) DEC #12

VOL 3, 1982

( ) JAN #1	( ) FEB #2	( ) MAR #3	( ) APR #4
( ) MAY #5	( ) JUNE #6		

INDEXES ARE INCLUDED IN THE JANUARY AND DECEMBER 1981 ISSUES.