PolyLetter

The Newsletter for PolyMorphic Systems Owners and Users

January/February #8401

NEWS

OSBORN & TI FOLD, POLYMORPHIC HANGS ON: Way to go, guys! It just goes to show that marketing (and big bucks) isn't everything. But please, the suspense about the new system is killing us.

AN 8810 WITH ONE HALF-HEIGHT 5 MB HARD-DISK AND ONE HALF-HEIGHT DSDD. Wouldn't that be something! Seems that it can be done, and the idea has been kicked around by more than one party, including PolyMorphic Systems. *PL* dubs this system the "88-5,000,000". We'll keep our ear to the ground and let you know if this system is about to be born.

NEW POLYLETTER FORMAT: As you can see *PL* ("Polyletter" is the name now used only for formal occasions) has changed its appearance. Frank Stearns Associates, a technical writing firm owned by *PL'S* new editor Frank Stearns, has remote access to digital typesetting facilities. While the number of pages in *PL* has dropped from 10 to 6, the number of words at minimum remains the same, and at maximum can increase by 30%. Later this year *PL* will explore remote typesetting and how it can be done with PolyMorphic equipment.

WHAT HAPPENED TO THE PARROT? The parrot is out for repair and maintenance. ("Poly wanna facelift. Brrrack.") Look for the bird next issue.

Post Office Says: I can't find that mailbox.

OOPS! We goofed. In the last issue of PL, the street number of PL's new home got mashed. PolyLetter's *correct* address is the following:

PolyLetter
14307 NE 16th St
Vancouver, WA 98664
206/892-3970

Poly Utility in 1984

Back in the September/October '82 issue of PolyLetter I'd written an article entitled "Four Years Later". It was subsequently republished in Al Levy's *The Stack*, an excellent newsletter produced by the Long Island Computer Association.

The article was one of loyalty and thanks for a system that had stood the test of time so well. But where are we now? How can a machine based on the quaint 8080 processor (and with a proprietary operating system and disk format) be of continued utility? One of the answers hasn't changed: it still takes time and money (in many instances more than it is worth) to convert from one system to another.

But now there appears to be more to the whole subject, much more than one can glean from listening to specifications shouted from the mainstream, and the internal and external harping about our own skeleton in the closet, incompatibility.

Back in the first days of consumer desk top-computers, Poly and a few others were there all alone. I can't help but think that whether they and their competitors knew it or not, they were there for more than just the money (unlike some of today's manufacturers). While unspoken, there seemed to be an altruistic undercurrent that here at last, the power of computing—and all that it implied—would be put in the hands of the many.

Then, the media—professional and consumer—got into the act. What was once a market consisting of inexperienced but thoughtful buyers got turned into a multi-ringed circus. The professional hucksters from New York and Los Angeles did battle with one another; the ultimate goal being to tell A. Citizen that it was as bad (or worse) to be without a computer as it was to be without the latest deodorant or shampoo.

Many Poly owners (if we venture into today's computer market place at all) draw back in shock. Minimal versions of what we have been using for quite sometime are pitched the same as bathroom tissue, and as if the innovation had been made just this morning. Many mainstream hardware and software offerings are certainly marvelous in their apparent power and scope, but too often the ease of use for daily and repeated chores does not seem to be there. Worse, some ho-hum but vital day-to-day utilities are lacking (such as the ability to list deleted files—Poly's "DLIST" command, and the ability to undelete files—the "ARISE.GO" program). The mainstream products often are designed solely for razzle-dazzle and the first week of user novelty. Little thought appears to go into how command complexities, non-mnemonic syntaxes, and speed bottlenecks can become irksome later.

Confusion for many would-be buyers (knowledgeable or not) is great—the sometimes know-nothing hi-fi salesmen of ten years ago (or their current-generation counterparts) are now manning the sales floors of the local computer supermarket. The promises are great; the bottom line prices match (unless all one wants to do is play video games); and the faceless selection of hardware and software is incomprehensibly huge. At least in the short run, those products that rise to the top do so more on advertising dollars than on ultimate quality of design. It is easy enough to push aside the hype, but when it comes to ascertaining actual performance, one is often unable to do so in a store, or even after several weeks of actual use. Over and over again, we are left with the feeling that the large majority of products are ill-conceived—not that they do not perform as advertised, but that the designs are rushed to market; and that the designs are skewed in favor of the novice, and worse, it seems, *keeping* him a novice.

If your system is doing all it should now, don't change it.

At any level, we were all new to computing at one time—we needed all the help we can get. The problem comes when a system design is so geared toward the perpetual novice that system potential and long-term ergonomics suffer. The irony is that had common sense prevailed (common sense is sometimes defined in marketing circles as "drabness"), there would have been little need to lock systems into a novice mold, and to drape them in vast amounts of help text that can become choking to floppy disk drives. With a little foresight, both advanced and beginner users can be supported comfortably. Take Poly's word-processing system, for example. If you're new to it and would like assistance with editing, formatting, and file management, use the WPS system to act in your behalf. If you know the Poly operating system at all, *all word-processing and file management functions can be done from EXEC, quickly and cleanly.* I know of no major word-processing systems that allow the user to choose between such opposite approaches!

Where does all of this leave the Poly owner? To some extent, if you're of a mind to participate in the outside computing world at all, bewildered. "Uh, Miss, how come this word-processing editor takes two or three non-mnemonic keystrokes to do what I'm used to doing in one mnemonic one?" Shocked, the pert but inexperienced sales person says, "What? All the popular editors work this way." Your query of *why* is met with a shrugged *that's the way it is.*

The trend for hardware manufacturers now is to provide little if any software with the system. Instead, it is bundled with whatever third-party or OEM software is the current rage. There are many good things to be said for this approach. For one, it gets around that devastating Poly bugaboo of incompatibility. For another, certain industry standards begin to appear. But in the process one may also lose a spark of originality. Like so many aspects of our marketed society, software is becoming a homogenized dark gray pudding.

What does the Poly owner do who may be thinking "new system"? First, disregard the marketing hype that has multipled like influenza virus over the past year. Look carefully first at the subtle points of user-interface, then at features. If you're going for specific features, make sure the every day ones are compatible with your Poly temperament. Second, even if you decide to buy a new system, hang on to your '10s and '13s. Several Poly owners have made a happy marriage between their Polys and a new system. Finally, if your system is doing all it should now, don't change it. Hardware-wise (as noted in the "Four Years Later" article), you've got a lot system life left. Wait with me. We can sit on the porch, sip our tall, cool drinks, and watch the "fire" clear out the computer industry underbrush. (As a company, PolyMorphic Systems seems indestructable at this point—witness Texas Instruments and Osborn.) With luck, there will be a "general reintroduction" of many of the basic PolyMorphic design tenets. And, we can have faith that PolyMorphic's new system will emerge as one of the winners. Though there is a catch-22 here: to survive, one has to have an element of mediocrity. Hopefully PolyMorphic's new system, in providing the user a choice of operating systems, will be able to call upon and push aside mediocrity as needed.

PL

As always, PL welcomes new readers, and stands as a forum in which the views and ideas of Poly users can be expressed, and information exchanged.

POLYLETTER

Editor and Publisher: Frank Stearns, PolyServe

Contributing Editor: Bob Bybee

Past Contributing Editors: Russ Nobbs, Bob Schwartz, Bill Davis, Bob Bybee, Ralph Kenyon, Al Levy, Chuck Thompson, and John Warkentin.

Subscriptions: (Domestic) $15.00 per year; (Canada) $15.00 per year, payable in US Dollars; (Overseas) $20.00 per year, payable in US Dollars.

Editorial Contributions: Your contributions to this newsletter are always welcome. Articles, suggestions for articles, or questions you'd like answered are readily accepted. This is *your* newsletter, please help to support it.

Advertising: On February 1st, commercial advertising rates will be announced for PL. If you're planning ads for the March/April issue, please contact us. (Personal ads will still be free of charge.)

PolyLetter is published bi-monthly by PolyServe. PolyLetter and PolyServe are not affiliated with PolyMorphic Systems.

CALLING ALL POLYMORPHIC DEALERS!

PL readers want to know who the dealers are. We'd like to publish a revised list for 1984. If you are a dealer and would like to be listed, please drop us a line or call.

Your wish is Poly's Command (File)

Perhaps one of the most under-used features of Poly's operating system (and often one of the most powerful) is the command file structure. Other systems offer command files, but few are as helpful as Poly's. Ultimately, the flexibility of Poly's command file structure comes down to this: if you can enter it on the keyboard (including control characters), you can put it in a command file. In fact, when command file mode is activated, the keyboard buffer is fed from a file instead of the keyboard. If you use your imagination, you can set Poly up to be a batch processing system. A command file can do a ton of work while you are warm and snug in your bed—it can even enter the editor or BASIC and perform various tasks!

The following questions are most frequently asked about the command file feature of Exec:

What is a command file? It is any file containing characters that you would ordinarily type manually. This includes all the alphanumeric characters, plus all the control characters. Those characters, of course, can form all the Exec commands, editor or BASIC commands, or data or menu selections within applications programs.

How long can the command file be? You are limited only by disk space.

Can one command file call another? No, unfortunately not. This is the one failing of Poly's command file structure. But it is not insurmountable. You can use the editor to merge command files or duplicate portions of other command files.

Command files can be used in place of the keyboard any time you wish.

How is the command file invoked? This is the best part: from Exec, simply enter the name of the command file and press return. (The extension is not important, though ".CM" seems appropriate.) Then sit back and relax, or go to lunch, or retire for the evening—whichever is appropriate.

When should command files be used? Again, command files can be used in place of the keyboard any time you wish. But they are most useful when you are repeatedly typing a sequence of Exec or other commands. Consider the following examples:

- If you are a machine language programmer, you know that during program development there can be many, many cycles of edit-assembly-test. A post-edit command file can be written to (a), delete the old object code file; (b), pack the disk; (c), call the assembler and provide the input and output filenames; (d), answer the assembler's set-up dialog; and (e), call the new object code file after it is assembled to see if it works. If you are entering data to be used with each test of the program, that too can be provided by the command file. You save tremendous amounts of typing, and are assured of unfailing consistency in the commands and data entered.
- Did you ever wish that the editor had a "super global" search and replace mode—a mode that once initiated, could make the same changes to several files? This can be done with two command files, eventually merged to form an "executable" command file (recall that one command file cannot directly call another). The first file (File 1) would contain the names of the files to edit and the commands required to enter the editor. A gap would be left for the second file (described next), and the last part of File 1 would contain the commands to exit the editor. This block is repeated for each filename.

  The second command file (File 2) would contain the commands and data needed to perform the search(es)/replace(s). To build the executable command file, alternate the elements of file 1 with repetitions of file 2.

  What about the control characters, such as control-B, ESC-control-E, and so on? You can't generate them in the editor—or can you? While in the editor, do a control-F (the find function). Enter the control characters you want; they'll appear as Greek characters. Press ESC. Press control-U to "undelete" the characters into active editor memory—there they are! Delete the second "dead" cursor and use the block markers to duplicate and move the characters as needed.

  What about ESC? That's easy too. Make an end-block marker (ESC left-pointing arrow). Press ESC-delete. Press control-U. There is your ESC character!
- Back before the editor featured the key-definition library, I'd designed a command file to define the keys. The editor was entered, then interrupted, and the command filename was typed manually. The first line in the command file reentered the editor (back in the days of Edit 81 "CONTINUE" was used, not "REENTER"), and the other lines of the command file went through the key definition process (ESC = key,definition). This idea might be applied to some of your own applications programming.

You can also use command file segments. The command file segment can do repetitive dirty work, leaving the final command or two for you to type manually. I use this concept frequently. I have several different but standardized WordMaster formatting sequences that require one or more environment files. To make sure that none of the environment files are missed, a command file is used to invoke the formatter and specify the environment files. For each different set of formatter/environment file calls, there is a separate command file segment. These segments go up to the point where a filename is needed—that is left to me to type.

There are, of course, some cautions and limitations. If you're doing a large scale search/replace for instance, make backups of the files before batch-processing them! Should you decide later that the replacement text was wrong, you have a starting place to return to.

Poly's command file structure, while basically "deaf and blind", does have some protective features. If the system encounters an error during command file execution, the command file will immediately abort. This assures that the command file will not continue when it may be disastrous to do so. GOTO 5

The Rescue

As the years tick by, those of us using older systems must stick together. The following true story can be an inspiration to us all, and can serve as an example of the kind of cooperation and helpfulness that prevents heart-attacks.

Thursday, December 15, 1983, 11:00 pm: Things had certainly gone well on this technical-manual writing project: the clients were pleased with the product—it had been honed to a fine edge and the readers were sure to enjoy it. Tomorrow began the final series of touch-up edits, followed by batch conversion of Poly's word-processing commands into typesetting commands. On Monday, Poly would be carted to the typesetting shop and all 350,000 characters would uploaded to the CCI mainframe.

I began to feel smugly confident in this very old system, and how, through a head-start in system evolution and a non-knee-jerk approach to design, it had served my needs so well these many years.

Then—trouble.

I returned from a phone call; Poly was in front panel. Pressing "G" did not restore Exec. The remaining front panel display had an odd, slow roll to it—not at all the usual "snap" of a new display if Exec could not be recovered. Slapping the load button, Exec came up cleanly. Then, within seconds, as I watched, the system slipped into front panel again.

By midnight I was in serious trouble. There was something very wrong; the system would not boot, the video display was filled with garbage most of the time.

It had to be the ancient ROM sockets acting up again. I had been having trouble lately. Strange things happened when the ROMs weren't making good contact. But after all, the system was due for its internal dusting, edge-connector cleaning, and chip reseating. That's all it was, I thought, relieved.

It was now 11:05 pm, but it wouldn't take more than 30 or 40 minutes to strip the system down, clean it, and reassemble it.

By midnight I was in serious trouble. There was something very wrong; the system would not boot, the video display was filled with garbage most of the time. And when the system finally did boot, it would not hold for more than a few seconds. Fighting a rising panic when considering the production schedule of the next few days, I managed to leave the problems for now—the freshness of early morning would put a new light on things.

Friday, December 16, 10:00 am. I'd been up since 4:00 am working on the system. The ROM sockets had been swapped, I'd even tried soldering the ROMs in place. Nothing seemed to help. If I didn't have a system on line by noon, I'd start falling way behind schedule. I could either attempt to continue fiddling with the system, and do so without the proper tools, or I could see if a Poly was available for loan.

The question to any Poly owner, especially to those living in metro areas with populations of less than 2 million, is where can I possibly borrow a system?

Enter Bill Davis, partner in Church-Davis, one of Portland's more progressive architectural firms, and owner of two Polys (and all sorts of pieces). Bill, like myself (and most Poly owners) had seen what a small system could do years before the trend had hit the nation. More importantly, Bill was (and is) one of those legendary nice guys, always willing to help out, even while putting his own staff in an awkward position.

A single call to Bill turned around my personal panic. It is difficult to describe the sensation of utter relief that washed over me during that phone call. Bill, croaking on the phone through a horrendous-sounding cold, said, "Take my second system for the weekend and the following Monday."

From there on out, everything went without a hitch. Charles Dickens could not have invented a better set of circumstances to express the meaning of Christmas.

The moral to this story (if there is one) is two-fold. First, get to know the other Poly users in your area. You may want to set up some kind of informal help arrangement so that if someone goes down, someone else can supply first aid. Second, you may want to investigate a second system, or at least a stack of circuit cards, so that you can swap things in and out should you have trouble. (An upcoming PL article will look into building a spares kit.)

EPILOG

What was the problem with my system? Basically, low voltage on the negative 18 volt supply. (If the following seems too technical, just think of how a flashlight performs when the batteries are weak or dead...) The low-voltage was due to an intermittent connector that ran from the power supply bridge to the filter capacitor. With the capacitor effectively out of the picture, there was nothing to "kick" the voltage back to 18-20 volts when the 0 volts portion of the pulsating DC wave was reached. Under load, the voltage dropped to a point where the on-board regulators could not work properly. The ROMs, CPU, and baud-rate generator could not function well, or if they did, it was on at best on a "ragged edge". With the connector reseated and cleaned, all was well.

Special thanks to Bob Bybee and Mark Maclin for their help and advice.

If YOU have problems and are under the gun, give us a call. We or someone we know will try to help. It is now generally recognized that the four most terrifying words in any language are *"the computer is down."*

PL

DISKS OF THE MONTH

All *Disks of the Month* released since April of 1980 are listed below. There are some wonderful items here, now is your chance to catch up on things you might have missed. New disks will be offered in future issues of *PL* .

Disk of the Month for July 83

DIS80.GO *Bob Bybee's fast machine language disassembler. The source code is included!*

ERROR.GO *Enter an error number and ERROR.GO will print the text.*

BERR.GO *Just like* ERROR.GO, *but BASIC error messages are displayed.*

MKDIR.GO *Creates one or more blank subdirectories at the time of disk initialization. Having all the subs "up front" improves system speed.*

This disk also includes some tutorial information and suggestions for novice Assembly programmers. A handy disk for the budding systems programmer.

Disk of the Month for March 83

DATABASE *Art Norton's impressive DBMS (data-base management system) for Poly. Includes several program and data files and complete documentation. A "typical" DBMS application is demonstrated both in the text and on disk.*

BASES.GO *Don Barrett's universal radix converter. Enter a number in one base (hex, decimal, ASCII, octal, or split-octal), and see the number in the other bases.*

SCAN.GO *Bob Bybee's disk-global search routine. Enter a string and* SCAN *will search the entire disk for it. If found, it'll tell you the file, and display the surrounding text.*

SNIFFALL.GO *Like the system command* Sniff, *but* SNIFFALL *checks the entire disk, even those sectors currently not used by active files. A must for a thorough surface check without INITing the disk.*

Disk of the Month for July 82

FIND.GO *Locates the specified filename—very handy if you have many subdirectories.*

SDIR.GO *A super listing program—* SDIR *can list all files in all directories of a disk, both to the printer and the screen, and it will list system and deleted files if you like.*

FPL.GO *John McGaw's front-panel lockout program. Stops the system from jumping into front panel unexpectedly.*

DX.GO *Captures all the files in a disk's main directory and writes them to a file (like Futil, only faster).*

MASTERMIND.BS *John Warkentin's board game for Poly. Like all good games, this one is simple but intriguing.*

CHANGE.GO *A utility by John Warkentin that allows you to swap system diskettes without blowing them (and Poly) out of the water. A must if you're frequently changing system disks.*

BOWLING.BS *Chuck Thompson's bowling game. Lets you guide the ball with your arrow keys. Up to four can play, and Poly can be a player.*

TEXT-TRAN.BS *Russ Nobbs' program to convert a text file to the equivalent BASIC "PRINT" statements. Takes the drudgery out of putting text into your BASIC programs.*

Disk of the Month for March 82

VM.VM *An adventure game; something of a mystery trip through logic and what we sense as real, and that which is not so real.*

Bchr.OV *Prints* ***BIG*** *characters on the screen. Set up a command file that loads* Bchr.OV *on boot, and surprise your employees.*

FETCH.GO *Loads any file in at any address. Primarily a utility for machine-language programmers.*

PUNCH.GO *Sends a file out the serial port in Intel Hexadecimal format to a ROM burner. Truly a tool for the machine-language hacker!*

Disk of the Month for December 81 (Games-2)

SLOT.BS *Turn Poly into a one-armed bandit.*

BACKGAMMON.BS *Poly Backgammon—no board or pieces to lose.*

ARTIL.BS *Turn Poly into a cannon and practice your artillery skills.*

MOON-LANDER.BS *Turn Poly into a LEM and practice your space-craft landing skills.*

Sex-Appeal.BS *Are you sexy? Poly will ask you some interesting questions and then you can check your scoring.*

Disk of the Month for July 81

READ.GO *Just like the Poly Exec command* TYPE, *except that you have cursor and bi-directional paging controls. Very handy.*

COUNT.GO *Count the words in your text files (disregards WordMaster commands).*

FLIES.BS *Poly swats electronic flies. The misses are infrequent. This program is somewhat comforting when you have trouble getting rid of the real things!*

INPUT.BS *Bill Sullivan's generic input routine: select a cursor; determine numeric or string input (with length limits); and test for faulty data.*

Disk of the Month for May 81 (GAMES-1)

BIORHYTHM.BS *Predict your Physical, Emotional and Intellectual well-being. Not a horoscope but a scientifically-proven method of deciding whether to get out of bed in the morning.*

BATTLESHIP.BS *Battle on the high seas—with Poly trying to sink you. Comes complete with a smart-mouthed ensign.*

CHESS.GO *Poly the chessmaster—watch out, she's a fair player.*

HANGMAN.BS *The Hangman has his itchy finger on the gallows rope. You'd better bee a gud spellr.*

SPIRAL.BS & ART.BS *Do pretty things on your Poly screen.*

Disk of the Month for March 81

TABBER.BS *Print a grid that can be used as a programming guide in lining up fields on a form, columns of numbers, etc...*

FNTIMER.BS *Two for one: Russ Nobbs' routine to place the cursor anywhere you wish; and find out where your programs are burning time.*

PEEK-DUMP.BS *Russ Nobbs' Utility to send the memory contents of the specified address range to the printer, in numerical and ASCII form (similar to SZAP's display).*

Tran.OV *Jon Wolfert's overlay for BASIC programs that converts numerical information to its textual equivalent ("1" becomes "one", "30" to "thirty", and so on).*

Disk of the Month for January 81

SORT-DEMO.BS *G. R. Gamble's program for comparing six of the more popular sorting algorithms. You'll be able see which sort (or combination of sorts) may work best for your application.*

MAZE.GO *Poly builds a maze, and runs a rat through it. Good graphics.*

READABILITY.BS *Donald J. Goodman & Sandra Schwab's program for running a sample of your text through the "Flesch Readability Scale". Determine the level of your writing. Program adapted from* Creative Computing, April 1980.

GENE.BS *Jim Garson's Genealogy program—chart your family tree.*

HOME-INVENTORY.BS *L.E. Sparks' program to track all that stuff around the house, and compute its replacement value at today's prices. Program adapted by Chuck Gross from a program appearing in the May 1980 edition of* Creative Computing.

Disk of the Month for August 80

POP.GO *Toggle the "system"/"not system" bit for any file. Make system files "unsystem", and unsystem files system files.*

Cursor.GO *Change the cursor to any character, any character at all...*

COPYALL.BS *Moves all non-system files to another disk.*

COPY-SUB-DIR.BS *Works the same as* COPYALL, *but copies files that are in a subdirectory.*

MOVE.BS *Don Moe's program to help recover data from a disk with a blown directory. Much easier than trying to rebuild the directory.*

ROOM.GO *How many bytes are left in the specified directory? Run this one from time to time and avoid that annoying "Directory Full" error.*

POKE.BS *A strange little ditty to let you dawdle and doodle on the screen. Appropriately enough, the character you doodle with is a question mark. No extra charge.*

Disk of the Month for April 80

CONTROL-U.GO *Once loaded, pressing control-U causes a screen dump to the printer. Many of today's system have a dedicated key for this.*

CALENDAR.BS *Print any calendar from the year 1582 on.*

$15 each. Includes mailing.

1983 INDEX

The following index is for the 1983 issue of PL. Back-issues may be purchased for $3.50 each. All six 1983 issues may be purchased for $16.00, a savings of $5. (Back issue prices include shipping and handling.)

1983 SUBJECT INDEX

Abstract Systems, Jan/Feb 4; May/Jun 5
BDMI, May/Jun 3
BUGS
  BASIC C03, Jan/Feb 3; Mar/Apr 7; May/Jun 7
  Edit 3.3, May/Jun 7
  Exec 95 Jan/Feb 3; May/Jun 7
  FTP, May/Jun 7
  ROMs 81 Jan/Feb 3
  Sio.PS, May/Jun 7
  Syquest, Nov/Dec 2
CP/M Commands, Mar/Apr 3;
  May/Jun 4; Sep/Oct 4
CP/M, compatibility Jul/Aug 7
CP/M, patches Jul/Aug 3
Disk failure modes, Nov/Dec 9
80 x 24 Displays, Sep/Oct 5
88/MS - New ROMs, Nov/Dec 3
8810, Jan/Feb 1
Exec 96, Jan/Feb 7; Mar/Apr 4; May/Jun 1
FTP as a terminal, Sep/Oct 2
HD/18 Mar/Apr 5; Jul/Aug 1; Sep/Oct 3
Header connector, serial card, Nov/Dec 5
Hex I/O in BASIC, Jan/Feb 7
Hidden Format.GO commands, Jul/Aug 9
Load Address, Start Address, Sep/Oct 3
Macros, Use of, Nov/Dec 4
Modem software Jul/Aug 5
Modems and Communications,
  Sep/Oct 7; Nov/Dec 5
New System Jan/Feb 1; Mar/Apr 1;
  May/Jun 1; Sep/Oct 1
Pascal Mar/Apr 1
Perfect Calc, Jul/Aug 5
PolyGlot, Jul/Aug 2
PolyServe, May/Jun 6
PolyShop, Nov/Dec 3
Printer Hookups, see
  Modems and Communications
SA-400 Disk drive, Jan/Feb 5; Mar/Apr 7
Serial card, header connector, Nov/Dec 5
Servicing (GSI), May/Jun 3
Setup.GO, new feature, May/Jun 2
Source, Poly users on, Mar/Apr 2
Subscriber List Jan/Feb 8-9
Upgrading the 8813, Jan/Feb 1
Using two printers from BASIC, Jul/Aug 9
Visicalc, Sep/Oct 5

1983 AUTHOR INDEX

Kenyon, Ralph
  How many Polys are There? May/Jun 8
  How Format.GO Works, Jul/Aug 6

Stearns, Frank
  Cook's Tour of the Shugart SA-400
  (Part 1), Jan/Feb 5
  (Part 2), Mar/Apr 7

Thompson, Charles
  Helpful Hints in Laymen's
  Language, Jul/Aug 8

Warkentin, John
  Volume Manager, Jan/Feb 4
  CP/M Commands, Sep/Oct 4

POLY ADS

Don Moe, where are you? Last we heard, you're in Europe somewhere. Contact Al Levy, P.O. Box 71, Hicksville, NY 11802. 516/293-8368

FOR SALE

PRINTER-Olivetti 221 Full featured electronic typewriter & versatile computer printer in a single unit. 16K buffer, selection of print wheels, serial and parallel, stated compatibles include Apple, TRS-80, Commodore, IBM PC+. Being used with Poly. 1 yr+ old, $1650 214/349-2367

VERBATIM DISKS (101). Some new, others limited use. Hard sectored (Poly compatible). $2.00 each. Will sell all or part. 214/349-2367.

Poly 8813, 56K, 2 double-sided drives, $1200.00
Poly MS DSDD, $1295
Poly Video Board, $70.00
Poly SSSD Disk Controller, 5 1/4 inch, $95.00
Poly 10 MB Priam Hard Disk, $2195

Call Charles V. Trayser: 415/651-0100 (days) or 415/651-5931 (evenings)

THE SMARTEST MODEM PROGRAM

SM.GO is the smartest terminal-emulator program ever written for the PolyMorphic. Features include:

- File transmit with variable "throttling"
- Data capture, with numerical display of remaining buffer space
- Any baud rate, any serial port (0 or 1)
- Half or full duplex
- SM 2.0 contains the popular XMODEM file transfer protocol used by many CP/M systems

SM will work with any modem, but is especially suited for use with the D. C. Hayes Smartmodem 300. When used with a Smartmodem, SM provides auto-dialing, using either tone or pulse dialing. SM uses a "macro command library" which can include all your favorite phone numbers. A single macro-command to SM can execute all functions needed to dial another computer.

SM is available on 5 inch SSSD diskette, for $50. Order from:

Poly Peripherals
1437 Sugarwood Lane
Norcross GA 30093
404/925-2480 (SOURCE: TCD125)

Poly 8813, 48K, keyboard and monitor, 2 single-sided single density drives, $1095.00. 800/231-4141 any time. Ask for Buck Ballas.

Poly 8810, 32K, perfect condition. Keyboard and monitor $795.00. 412/462-6421 9:30 am to 3:30 pm (EST); 412/683-0509 evenings.

WANTED: Poly CPU card without ROMs. Need not be equipped to run CP/M.

G.R. Gamble
5615 NW 63rd Place
Des Moines, Iowa 50323
515/278-5332

MORE HARD (DISK) FACTS

Thanks for your response to **Poly Peripherals** hard disk subsystem! At **your** request, our Hard Disk III is now available with up to 15 megabytes of storage.

5 megabytes	$1990
10 megabytes	$2290
15 megabytes	$2390

Prices include cables, instructions, device driver software, and shipping to the continental United States. Our hard disk subsystems will plug in and run on your single-user 8813 or 8810, running Exec/96 with Volume Manager. No need to modify your programs... just watch 'em run faster! For more information, call or write:

Poly Peripherals
1437 Sugarwood Lane
Norcross GA 30093
404/925-2480 (SOURCE: TCD125)

If you own more than one Poly, ask us about a quantity discount!

It's tax time again! Chuck Thompson, Poly owner, Dallas lawyer and tax accountant, will soon be releasing the 1983 version of The 1040 Tax Preparation System. This program package calculates and prints the 1040, Schedules A, B, C, G, and W, and allows storage and retrieval of your tax data. During the preparation phase, you can change any figure on any form and get instant recalculations! (Try doing that with your pencil and pocket calculator.)

$150.00 for first-purchase by tax preparers
$75.00 for first-purchase by individuals

For more information contact:

Chuck Thompson
PolyCom Associates
2909 Rosedale Ave
Dallas, TX 75205
214/691-1114, days; 214/368-8223 evenings (SOURCE: STJ970)

Command Files continued

The command file will be effectively shortened or turned off by a "flush" or "killi" call within the applications program or Exec command. (Placing the system command INIT in a command file is an example of this.) "Flush" or "killi" are system routines that effectively clear the keyboard buffer of its present load of characters. "killi" completely terminates the command file, where "flush" may continue if there is more than a keyboard buffer's worth (64 characters) of data in the command file. This can be both a precautionary feature and a hindrance.

Poly's command files are non-interactive. That is, once started, you can't get into the act midstream with "live" keyboard commands. You must use a control-Y to interrupt a running command file. This isn't necessarily a limitation. It is, in fact, a property of "batch processing".

The next time it seems as though you're typing a lot of the same commands or key sequences over and over again, consider placing them in a command file. Make Poly your slave—not the other way around. PL

The Last Minute

Events always move faster than typesetters and printers. This just in from the East coast:

Soon we'll have to start referring to the owner of Abstract Systems, Ralph Kenyon, as Dr. Kenyon. Ralph is pursuing an ambitious Ph.D. program, and hopes to be done within four years. As Ralph put it, "the clock is running!" Ralph is still doing Poly things, but has set up a dealership arrangement with Al Levy. To obtain Abstract Systems products (and for information on other software libraries), contact Al Levy/P.O. Box 71/Hicksville, NY11802/(516) 293-8368.

Ralph is still available for special projects. Contact him at Abstract Systems, etc./RFD Lower Prospect Hill/Chester, MA 01011/(413) 354-7875.

Next issue, PL has some interesting features and begins two new series.

- We hope to be able to tell you in detail about the new system. There may even be a photograph or two.
- "Pascal Pastures" will be launched. In the first two articles of the series, structured programming in general will be overviewed. The first will be Bob Bybee's article on how to structure software design in notoriously unstructured languages, such as BASIC and Assembler. The next article will overview Pascal and how it differs from BASIC. Later articles in the series will deal exclusively with Pascal and Pascal's implementation on Poly.
- "Building a Spares Kit" will examine how you can minimize your downtime risk. As many 8813s and 8810s approach their fourth and fifth years of operation, this becomes a very timely topic.
- Continuing on the theme in this issue's editorial, next issue will compare the number of keystrokes it takes to do certain routine tasks in a Poly, and the equivalent tasks in one of the popular micros.
- As space allows, a "How it Works" series will begin. Each article in this series will be short and as nontechnical as possible. Both hardware and software will be explored.

PolyLetter

PolyLetter
14307 NE 16th St
Vancouver, WA 98664
206/892-3970

FIRST CLASS MAIL

PolyLetter

The Newsletter for PolyMorphic Systems Owners and Users

March/April #8402

News

EPSON AMERICA OFFERS 1/3 HEIGHT FLOPPIES (8810 owners, an 8813 is within your grasp—assuming you're willing to pay for it): PolyLetter was recently alerted that 1/3 height floppies do exist. However, they are expensive ($320+ each), hard to get (90 to 120 day delivery), and they are only half as fast as the 1/2 height Teac drives costing $100 less. One optimistic engineering friend of *PL's* suggested that perhaps the little drives would cost less than a 1/2 height simply because fewer and smaller parts were involved. Not so, it seems. Why the high cost and slow delivery? The usual: no competition, it's a new product, and the marketing people are probably being cautious, thus production is not as high as it could be. Industry wide, many people view *three* drives as a little strange, even though the convenience of an extra data drive is well known to Poly owners.

The outlook? Maybe a drop in price, maybe not. *PL* sees the drive's purpose in life as not that of putting three drives in the place of one, but rather putting two drives in a third less space than two 1/2 heights.

The New System is "here", finally, and PolyLetter is awaiting official specifications from PolyMorphic. There are two versions. The "new system" is a two-board, 8186-based system living in an 8810 chassis. It's structured so that three more "users" can be added simply by adding three additional boards. While the price seems a little high ($4495) it really isn't when the amount of software and up-front expandability of the system is considered. Features include full-color graphics and local networking. The price also includes the latest versions of CP/M (3.0) and concurrent CP/M, a "Windows" package, and networking software. Considering how expensive local networking can be, getting it thrown in free is indeed a good deal. For Poly owners not exposed to this concept, local networking is nothing more than a way to set up communications links between multiple users in an office environment thus allowing the sharing of expensive peripherals such as high-quality printers, massive amounts of hard disk storage, and so on. Local networking also provides some form of electronic mail, which can be very useful and time-saving.

The "upgrade system" consists of one board that drops into your existing 8813. This is the one we've heard the most about in the past year. Pricing still seems a little hazy, but it's in the $1500 range. Definitely worth looking into.

PolyMorphic tells us that they have had a "generous" allocation of parts from Intel, and that AMD is coming on line as a second source of supply. Production is scheduled to begin in earnest in April.

EDITOR'S CHIPS

There is a lot of interesting material in this issue. While not specifically about Pascal, Bob Bybee's article about structured programming is fascinating (page 2). The uninitiated occasionally feel overwhelmed by the mere concept of programming. Bob shows us that "divide and conquer" is a valid concept, and that programming is not as hard as it looks (and sometimes is). *Pascal Pastures* may be combined with *C Ways*, since the language *C* is as powerful as Pascal but has few of the eccentricities. Watch future issues of *PL*. *PL* has taken a crack at the spare parts issue. The text grew so fast that the article has to run in two (or more) parts: this month talks about tricks to avoid going down in the first place; the continuation next issue delves into simple troubleshooting techniques, and spares to have on hand.

A very special *thank you* to the many *PL* readers who wrote. The January/February issue was produced hastily under many converging vortices of Murphy's Law; your comments were sincerely appreciated. Time did not allow individual replies, but hopefully this will not be the case in the future.

The famous PolyMorphic Systems Editor on CP/M: During the course of preparing CP/M-based software for the new system (and obviously being disgusted with all available CP/M editors) Lenny Araki of PolyMorphic Systems has rewritten the Exec Editor to run under CP/M. PolyMorphic plans to market the editor as a stand-alone piece of CP/M software. Now, finally, the "rest" of the computing world will have a chance to see one of the reasons why we Poly owners so adamantly refuse to give up our systems.

"C" ON POLY: What is "C"? C is one of the hotter languages these days. Invented by BELL LABS for serious programming, "C" has many of the structured niceties of Pascal (without Pascal's eccentricities), and much of the power of writing in machine language. Art Norton is putting together a mini-*C* compiler for Poly, and in the future we hope to offer it on a Disk of the Month.

THE PARROT IS DOING NICELY: Extensive light-table and razor blade surgery has been performed on the *PL* parrot, and the bird is recuperating in El Segundo. While this was to be the issue of the bird's return, things didn't quite work out that way. Next issue, definitely.

Pascal Pastures

Before getting into the specifics of Pascal, an overview and review of structured programming is in order. Try digesting this article on two planes: first, as an interesting discusion of how good programming is put together, regardless of the language; and second, as an introduction to structured programming.

TABLE-DRIVEN PROGRAMS

Bob Bybee, Poly Peripherals

Whether you realize it or not, you have used many table-driven programs on any computer you have operated. A table-driven program is simply a program whose actions are controlled, in some fashion, by data stored in a table. Most large programs lend themselves to this structure, since they usually have a list of functions to perform, based on some input from their user.

The assembler, editor, formatter, and BASIC interpreter, are all examples of table-driven programs. In the case of BASIC, the tables consist of keywords, such as *REM* and *IF*. The assembler's tables contain assembly language mnemonics (instructions), such as *LXI* and *STA*. The editor has a table of control-characters including Ctrl-B and ESC/Ctrl-E.

Even the operating system is a table-driven program. Like many programs, it has more than one table. It has a table of built-in commands which includes DISABLE and LIST, and then it uses the disk directories as secondary tables of commands and filenames.

Most large programs can, and should, be written this way. A table-driven structure organizes your thinking. Usually your program will have a list of "things coming in," and "things to do with what just came in." Without using any particular programming language, we can describe the general structure of a table as follows:

```
Input #1
Function #1
Input #2
Function #2
.
.
END-MARKER
```

Notice that the inputs and functions are paired. They could both be stored in the same table, or they could each be in a separate table. If you separate the "inputs" and the "functions" into two tables, it becomes easier to get them out of sync with each other... a nasty bug! Input #31 might perform function #32, and so on.

Also notice that we use some sort of marker at the end of the table. Some programmers prefer to count the number of items in a table, rather than using an END marker. The disadvantage of a count is that it has to change whenever the table is modified. Also, why should we bother to count things? The computer is much better at counting than we are.

Let's write an example program. Suppose we have a fancy printer, one of those new dot-matrix jobs, with functions like italics and underlining. WordMaster probably can't make use of those functions, so we need to write a program that will switch them on and off for us. Most printers use some very cryptic code sequences for their control functions. Let's make up a fictional set for our example: Pretend that "italics on" is ESC I, "underline on" is ESC U, and "back to normal" is ESC N. We would like to write a BASIC program that lets us enter these commands without having to remember what the magic code sequences are.

As usual, having a good data structure will help us design the program. Let's set up two tables: the first is a list of strings, which will be the commands we enter.

```
300 DATA "ITALIC","UNDERLINE","NORMAL","+"
```

(Notice we used a "+" sign by itself as an end-of-table marker.) The second table is a list of the magic codes for the printer's commands. To simplify the program, we will assume that all the printer commands begin with ESC, and we won't bother to enter that in the table. Some other piece of the program will have to add the ESC for us.

```
301 DATA "I","U","N","+"
```

Now that the tables are defined, it's usually a simple matter to write the program that handles the data. Here is one way it could be done.

```
10 DIM A$(20:64),B$(20:1),C$(1:64)
20 REM  read the command table
30 I=1
40 READ A$(I)
50 I=I+1\IF A$(I-1)<>"+" THEN 40
60 I=I-2\REM  I is # of commands
70 REM  read the output table
80 FOR J=1 TO I  \READ B$(J)  \NEXT
90 REM show the command list
100 PRINT "Your choices are:"
110 FOR J=1 TO I  \PRINT A$(J)  \NEXT
120 REM  get his choice
130 INPUT "Which one? ",C$
140 REM  see if it's valid
150 FOR J=1 TO I
160 IF C$=A$(J) THEN 230  \REM  got it
170 NEXT
180 PRINT "Invalid choice."  \PRINT
190 GOTO 100
200 REM  his choice matches A$(J)
210 REM  send corresponding B$(J) to the
220 REM  printer, with ESC prefix
230 FILE:2,LIST
240 PRINT:2,CHR$(27),B$(J),
250 STOP
300 DATA "ITALIC","UNDERLINE","NORMAL","+"
301 DATA "I","U","N","+"
```

One great advantage to a table-driven structure is the ease with which it can be modified. Suppose you want to add a new function to your program. No problem—just add some entries to the tables! In our example program, we would need to add a command to the first data statement, and a "thing-to-send-out" to the second one. Note that we dimensioned A$ and B$ to have 20 entries each, even though our table only has three items in it now. We have allowed the tables to grow to 20 items before the DIM statement would need changing.

Table-driven programs in BASIC are fairly limited because BASIC can only have tables (arrays) of numbers and strings. In many other languages, tables can hold much more interesting data types, like the addresses of subroutines. In Pascal, you can even define your own "custom" data types!

Here is a portion of the table used in Poly Peripherals'

SM modem program, *SM.GO* written in 8080 assembly:

```
DB 'ECHO',0,2,'OFF',0,'ON',0
DW Echo
DB 'BUFFER',0,3,'OFF',0,'ON',0,'CLEAR',0
DW Buffer
```

In this table, the entire command strategy of *SM.GO* can be seen. There are two complete commands in this portion of the table: ECHO and BUFFER. The ECHO command is in quotes, and has a zero after it. This is analogous to the "command name" in our BASIC program. (The zero at the end of a string is necessary in assembly language.) Following the zero is a 2, which is the number of options this command accepts. The 2 is followed by a list of those two options, each with its own terminating zero. So the ECHO command can be used in two forms: ECHO OFF or ECHO ON. On the next line is the memory address of the subroutine that performs the ECHO command. It is named, logically enough, Echo.

Knowing this much, we can immediately discover how the BUFFER command works. It has three options, OFF, ON, and CLEAR. And it is executed by the subroutine named Buffer.

Adding a command to this kind of program becomes trivially easy. First, add the command name to the table, along with the number of options and a list of them. Next, write a subroutine to perform that command, and add that subroutine's name to the table also.

The idea for writing a "control-your-printer" program is actually a very useful one. Most of the printers being sold today do have special options, like the ones we invented above. The trouble is that everybody's options are different. The second problem is that writing this program in BASIC is cumbersome—to run this little tiny program, BASIC must be loaded, after which you probably want to PRINT a file or use WordMaster, not BASIC.

PolyLetter to the rescue again! In a future PolyLetter Disk-Of-The-Month, we will include a program called SETPR, written in assembly language. It will have a table of printer commands, much like the BASIC program above. Even if you don't know how to program in assembly, you will be able to edit this table and insert the commands that your printer understands. Then you can assemble the program (instructions will be included!) and use it to control your printer's magic functions. And that will demonstrate one more selling point of table-driven programs: without knowing how to program at all, you can modify the *table* (not the program) and customize its function to your needs. PL

SETPR *will be available in the May 1984 Disk of the Month.*

POLYLETTER, March/April, 1984

Editor and Publisher: Frank Stearns, PolyServe

Contributing Editor: Bob Bybee

Subscriptions: (Domestic) $15.00 per year; (Canada) $18.00 per year, payable in US Dollars; (Overseas) $20.00 per year, payable in US Dollars.

Editorial Contributions: Your contributions to this newsletter are always welcome. Articles, suggestions for articles, or questions you'd like answered are readily accepted. This is *your* newsletter, please help to support it.

Advertising (camera-ready or PL typesetting): $2.00/column inch; full column: $18; full page $30 (Personal ads are still free of charge.)

PolyLetter is published bi-monthly by PolyServe. PolyLetter and PolyServe are not affiliated with PolyMorphic Systems.

WHAT DO YOU SAY TO A DEAD POLY?

(or: Building a Spares Kit, Part 1)

What do you do when your Poly won't boot, the keyboard is dead, or your top of RAM is suddenly 3DF8? Depending on your application and technical knowledge, the reaction can range from a sigh of annoyance to utter panic.

Fortunately, we Poly owners are blessed. The hardware is fundamentally sound. A '10 or '13 can go for years and years without a hitch, and when something does finally give, it is usually quite "findable". But, crashes still do happen.

There are several ways to avoid and cope with mishaps. Part One of this article looks at preventive maintenance, Part Two, Troubleshooting and Spares.

ENVIRONMENT

Dust and moisture (or excessive dryness) are long-term killers. The 8813 has two 100 CFM (cubic feet per minute) fans providing life-prolonging cooling. Unfortunately, if you run your system for 10 hours a day, you pull over 43 *million* cubic feet of air through it each year. That's a lot of air, particularly if the air is not as clean as it could be.

Electricity and particulates have an interesting attraction. If you've ever looked in the back of a TV set that's been in service for a while, you can see the dust motes neatly and often horrifically stood up on end in several parts of the chassis. This is due to the very high voltages involved. In computers, a similar problem exists only in this case the dirt attraction is due to the high-frequency activity of the system. While the attractive force is much weaker, smaller particles still make their way into important places such as the contact point between the edge connectors and printed circuit boards, or even where the metal legs of the chips meet the IC socket contacts.

Ever noticed what happens to a dusty surface wiped with a damp cloth? On the first pass, streaks of mud form. In a high-moisture environment, the same can happen to those minute areas of metal-to-metal contact inside your computer. Under the right conditions, these are disastors waiting to occur.

There isn't a great deal to be done about this, other than being aware of your climate, and adjusting your maintenance schedule accordingly.

CHIP SEATING

Many problems stem from chips working out of their sockets. Fan vibration, pin tension, and time work together to eventually force many ICs at least to a point of marginal contact. When this point is reached several things can happen. First, seemingly random intermittent problems may occur. Error messages are seen for no particular reason; maybe the system will jump into front panel (while the printer is NOT in use); there may be excessive noise in the video display; and so on. One is left with a feeling that all is not well. You're right. Circuit impedances actually change, forcing various chips outside of their respective operating tolerances, thus causing unpredictable results.

Fortunately, this is one of the simplest problems to fix. Remove power from the Poly, remove the lid, and one by one remove the boards. Note the position of any plugs or cables running to or from the boards. (If this is the first

time you've been in the system, make a simple diagram of what goes where, especially the cables and their orientation.) Lay the boards component side up on a flat, relatively static-free surface. (Don't use the carpet!) At each socketed IC, press down firmly with your thumb. Did you hear a metallic-sounding crinkle? The contact points just shifted down deeper into the socket where they should be. If you actually feel the chip physically move, it was barely seated and this was probably the cause of any problems you may have had. Continue, chip to chip, until all chips have been checked. Many will make no sound, indicating that they have bottomed out. Don't press too hard, or you may hear a "crack" instead of a crinkle!

As long as your desk is covered with circuit boards and the chassis it stripped, there are several other things that can be done:

CLEANING EDGE CONNECTOR "FINGERS"

The row of contact points at the bottom of each board can become very corroded, again depending on your hours of use and environment. The contacts should be a lustrous to shiny brass or gold color, with very faint or no visible streaks in them. If any streaks are plainly visible and very dark, you have a potential problem. Using *pure* denatured alcohol (**not** rubbing alcohol or any other alcohol containing any amount of water) and hard cotton swabs (not the fluffy kind), gently rub the row of contacts. Dark streaks should disappear almost immediately. If the fingers haven't been cleaned for more than or year repeat the rub five or six times with a new, alcohol-soaked swab. In later cleaning missions (typically two to three times a year), one or two passes should be enough.

If you actually feel the chip physically move, it was barely seated and this was probably the cause of any problems you may have had.

You can't do a great deal with the edge connectors on the backplane, but these seldom cause problems and are theoretically "self-cleaning". Just the action of unseating/reseating the boards will help.

TIGHTENING THE REGULATORS

Each board has at least one power-supply regulator associated with it, often three. These are the devices mounted in (or with) the finned heatsinks. In the case of the larger regulators (+5 volts), one of the contact points is made through the mounting screw. Sometimes these screws can become very, very loose, again raising havoc with the system, especially as the operating temperature changes. Using the appropriate screwdriver, snug these screws. Not too tight! You can damage the board, the insulator, or the device.

CLEANING THE CHASSIS

Take a vacuum with a duster attachment and go over the mother board, power supply, and fan housing. Watch out—if your vacuum is high-powered, there may be fewer parts in the system when you're done than when you started! You may want to begin with a fresh bag so

GOTO 5

A Two-drive 8810

Part One: *Mechanical*

Bob Bybee, Poly Peripherals

Several reagers have asked about upgrading the 8810 into a multi-drive system. PolyMorphic offers a kit, but if you're creative and have access to some special tools, you can not only make the mechanical conversion but can also improve the response of the 5 inch disk system. Bob recounts his experiences here.

Part One covers the mechanical aspects of converting to 1/2 height drives, Part Two explains how to modify the ROMs to get the most out of the drives.

I recently finished upgrading an 8810 to a two-drive system using the Teac model FD55A half-height drives, which are single sided, double density, even though they are in single-density system. These drives cost between $199 and $230 each. (Pick up a recent edition of BYTE and call around to the various mail order houses advertising a toll-free number. PolyLetter has had personal experience with *Priority-1 Electronics* and *Jade Computer.*)

As you can imagine, a two-drive 8810 is a vast improvement over a single-drive unit! The extra drive allows us to IMAGE disks and copy files easily, instead of going to another system to do backups, or going through a tedious disk-swapping procedure.

The upgrade was fairly easy, and if you have a little mechanical ability you should have no problem with it. Here are some of the highlights of the conversion:

Mounting

The Teac drives have a plastic faceplate that's a little more than half the size of the original Shugart SA-400 drive, so it was necessary to file down one edge of one drive's faceplate. Of course, the drive mounting hardware had to change. A couple of pieces of scrap metal were bent into new mounting brackets. Then I discovered the Teac drives use metric hardware! A tapping set was used to convert the mounting holes to a standard 6-32 thread.

To mount the drives, the 8810 card cage has to be taken out of the system. This is accomplished by removing some screws on the bottom of the chassis, and sliding out the cage. Before re-installing the card cage, this is a good time to put in the new disk cable, or modify the existing cable by adding another 34-pin edge connector for the extra drive.

Configuration

The drive-select jumpers must be set on each drive. One drive can be left as drive 1, and the other should be jumpered as drive 2. The usual convention is drive 1 on top, drive 2 is on the bottom.

Remove the terminating resistor package from "first" drive that the controller will see. If you cable from drive 1 to drive 2 to the controller, then the pack should be removed from drive 2. The resistor pack looks like a black or grey IC chip, and is the only IC on the drive board that has a socket. Do not remove the pack from the other drive.

After the modifications above, the two-drive 8810 worked very well. You can stop at this point, and have a very pleasant upgrade to your 8810. But there's still more that can be done.

NEXT ISSUE: Modifying the ROMS Pl

READERS WANT TO KNOW...

Don Barrett, of Micro 100 in Riverside, California, writes: *I recently bought a printer that can do graphics and a lot of other neat things—but I need to be able to pass 8 bit words out the serial port. No matter what I try, I can't seem to get 8 bit words out of Poly's printer driver. And what about BASIC? Even if I figure out a way to get 8 bit characters out, can BASIC pass them?*

There seems to have been one oversight in the design of the Poly printer driver. There are no options in Setup.GO's dialog regarding word length, stop bits, and parity, even though the hardware is quite capable of being programmed for whatever is needed. Lenny Araki of PolyMorphic tells *PL* that the printer driver is fixed to 7 bits, odd parity, two stop bits, regardless of the speed used.

To answer the question about BASIC, no, it makes no difference. No matter where the data to be printed comes from, it must ultimately pass through the USART hardware, which is programmed when the printer driver is loaded or when a new printer is "connected" in software.

There are several ways around the 7 bit problem, let's start hard and work to the more simple solution. If you wanted ultimate control, you would have to write a dialog program that would include those parameters not covered by the Poly dialog; create a new data byte from

No matter what I try, I can't seem to get 8 bit words out of Poly's printer driver.

the results of the dialog; search the Sio.PS file for the call to a ROM subroutine called *Setup* (different versions of Sio.PS may have the call in a different place within Sio.PS); and then count off six more bytes and overwrite the old USART programming data byte with the new byte. When you were through, the printer driver would have to be reloaded to execute the new USART programming. From then on, the modified Sio.PS would be loaded on boot.

Sound hard? Not really, just tedious. *(PL will try to include such a program in the next Disk of the Month.)*

There is a simpler way, at least in the short term, and your printer probably supports this solution. For example, a Gemini 10 printer can be told in software to "define" the state of the eighth bit, even though information is entering in a seven bit format. Sending an "escape" character (27D) followed by a greater-than sign (62D) tells the Gemini printer to force the 8th bit high. Escape/equals-sign forces the bit low. In this way printer characters between 0 and 127 or 128 and 255 can be accessed while still using a seven-bit interface. Thus, in BASIC, a PRINT CHR$(27),CHR$(62) tells a Gemini printer to add 128 to all incoming characters.

WordMaster can be a little more difficult, because the actual characters must be used, not their ASCII values in decimal. WordMaster likes to filter out those characters. Here is a good use for Ralph Kenyon's added formatter command, *chr X*, where X is the decimal value of the wanted character. (Contact Ralph at 413/354-7875.)

So you can see that the problem is not insurmountable, simply annoying. Perhaps we'll see an official revision of Setup.GO, which will take care of this problem. PL

continued from page 4

that any gobbled parts can be recovered and reinstalled. There aren't many of these at risk in the chassis (just the pullup arrays at the rear of the mother board and a socketed chip or two on the power supply boards). Be aware and alert.

REINSTALLING THE BOARDS

Careful! If this is the first time you've pulled the boards, be sure to reseat them correctly. Face them the way you found them (component side toward the front of the system), slide them into position gently, then apply even, constant, pressure. Don't force! (8810 users note: **support the rear of the backplane with one hand.**)

HEAT

Heat is the enemy of all things electronic. Don't obstruct the fans or the airflow in to or out of the system. Make sure the fans are clean and working properly.

RUN THE CONFIDENCE DISKS.

Run the confidence disks. No kidding. Run the confidence disks.

DISK DRIVES

Refer to Frank Stearns' two-part article *A Cook's Tour of the Shugart SA-400* in the January/February and March/April 1983 PolyLetters.

Next issue: Troubleshooting and Spares.

PL

WANTED—LISP for Poly. (LISP is the programming language of artificial intelligence.) Contact Ralph Kenyon; Abstract Systems, etc.; RFD Lower Prospect Hill; Chester, MA 01011

PL'S FIRST CONTEST: That's right, *PL* is going to stoop a bit and hold a contest. Grand prize is a free Disk-of-the-month (your choice) or a year's free subscription. The contest? Name the parrot. (No, it is *not* "Poly".) Your name will also determine the bird's gender, which has been something of a mystery for many years now. Even after surgery this bit of information is still unknown. Entries must be postmarked no later than 5-1-84.

The Last Minute

This "Last Minute" really isn't a last minute. Much to the horror of the production department, the "next issue" copy didn't get to the typesetters! Anyway, next issue PL concludes Bob Bybee's "Two Drive 8810" article. Bob tells how to alter the ROMs to enhance the speed of the Teac drives. A related piece by Bob, "What's in a Rom?", is also featured. Pascal Pastures begins an overview of Pascal and where appropriate, "C". PL's "Building a Spares Kit" series continues. A new Disk of the Month will be announced.

MORE HARD (DISK) FACTS

Thanks for your response to **Poly Peripherals** hard disk subsystem! At **your** request, our Hard Disk III is now available with up to 15 megabytes of storage.

5 megabytes	$1990
10 megabytes	$2290
15 megabytes	$2390

Prices include cables, instructions, device driver software, and shipping to the continental United States. Our hard disk subsystems will plug in and run on your single-user 8813 or 8810, running Exec/96 with Volume Manager. No need to modify your programs... just watch 'em run faster! For more information, call or write:

Poly Peripherals
1437 Sugarwood Lane
Norcross GA 30093
404/925-2480 (SOURCE: TCD125)

If you own more than one Poly, ask us about a quantity discount!

PolyLetter

PolyLetter
14307 NE 16th St
Vancouver, WA 98664
206/892-3970

20

FIRST CLASS MAIL

PolyLetter

The Newsletter for PolyMorphic Systems Owners and Users

May/June #8403

News

SOURCE TELECOMPUTING OFFERS MAJOR DISCOUNTS: Remember the SOURCE? They were one of the first "consumer data bases" around, offering many services to the average home computer owner. Trouble was, the registration fee ($100) was rather steep for many people, and the fact that you had to also buy the manual ($29+) seemed painful. That has changed. The prices have dropped terrifically ($29.95 registration), and the manual is now included free. Contract Source Telecomputing, 1616 Anderson Road, Mc Lean, VA 22101, 800/336-3366. (Note: *PL* has been a SOURCE subscriber for over four years. If you would like additional, "non-marketing" information, contact *PL* at 206/892-3970.)

ABSTRACT SYSTEMS introduces "Super Disk"—1600 sectors on a 5 inch disk. Ralph Kenyon, busy with a Doctoral program, still cannot stay away from his Poly. Ralph has developed the ROM modifications necessary to use the full 96 tracks per inch capability of the Shugart SA-465. What's neat about Ralph's modification (besides the obvious greater than four-fold increase in online storage over standard SSSD storage) is that you can use your *existing* Poly SSSD controller board (revision "E" or later). One major expense of converting to higher-density disks is eliminated. Ralph says the drives are coming down in price. The SA-465 is in the $150-$175 range. Contact Ralph Kenyon, Abstract Systems etc., RFD Lower Prospect Hill, Chester, MA 01011; 413/354-7875.

8810 Owners—the options just don't stop. *PL* has talked about 1/2 height drives and 1/3 height drives, all in the name of making your little system a big system (witness Bob Bybee's concluding article on 1/2 height conversion in this issue). But for those of us unwilling at this time to buy new drives and "hammer and saw", **Bob Bybee, Poly Peripherals,** has a new, elegant solution. Bob has developed a tiny piece of software that resides in high memory and intercepts calls to Dio. The program is smart enough to turn one drive into three—you can specify drives two or three, and this program will prompt you for the disk required. Versions 81 of the ROMs or later are required. Contract Bob Bybee, Poly Peripherals, for more information: 1437 Sugarwood Lane, Norcross, GA 30093, 404/925-2480.

ESC-Control-K: Just when you thought you knew everything there was to know about the editor, Russ Nobbs, switching from his Oz running *Wordstar* to his Poly, accidently hit an ESC-Control-K command. According to Poly (second hand), this undocumented sequence is just like the familiar ESC-Control-E, except that the editor itself is *not* exited. Once ESC-Control-K is executed, the entire contents of memory plus any part of the input file still on disk is dumped to the output file. You can at this point open a new output file (ESC-Control-O) and go on about your business, or open another input file. You can get a similar effect using markers and ESC-Control-D, but ESC-Control-K is simpler, cleaner, and doesn't forget the balance of the input file that may still be on disk. (ESC-control-K is available only in Editor versions 4.0 and later.)

WHAT DO YOU SAY TO A DEAD POLY?

Building a Spares Kit, Part II

The Philosophy of Spare Parts—from a Standby System to box full of components.

The following sections cover the "levels" of spares. It is assumed that to reach this point you have been convinced that "the problem" is due to a system failure in hardware, and that the problem is *not* due to faulty media, buggy software, dirty contacts, or other possibilities discussed in part 1 of this article. It is also assumed that, if possible, you have used the confidence disks to help isolate the problem(s).

SYSTEM LEVEL

Of course, the best way of handling an emergency is to have the ability to move your disks to a second system. Ideally, this means that system #1 and system #2 are clones of one another in all respects. At second glance this may not seem so impractical. Some people are selling their Polys, and for an investment of $500 to $1500, you can buy the ultimate peace of mind. Trouble is, if your main application is business you may find that the "spare" system quickly reaches the same "critical path" status as the system being backed up. If one of the systems goes down, you may not be in quite as much trouble, but almost! On a positive note, it takes little technical background to transport the disks of a dead system to another system.

SUBASSEMBLY or "BOARD" LEVEL

By this we mean disembodied boards and disk drives, waiting to be dropped into place should something fail. This is not a bad way to go if you can spot a "deal" here and there, and buy back-up boards for what you think may cause you grief. Ultimately, however, Murphy says that the part for which there is no spare will be the one to fail at the most critical moment. By the time you've covered all the bases (don't forget the keyboard, monitor, and power supplies), you have a second system, sans box. It seems foolish to let those parts stand idle so, as others have done, you buy an S-100 chassis to house your "working" spares. In this action you have created a second, fully operational system. GOTO .3

A Two-drive 8810

Part Two: *Modifying the ROMS*

Bob Bybee, Poly Peripherals

(Last issue, Bob told us about the mechanical conversion of one-drive 8810 to a two-drive system. This month Bob tells how to get the most out of the new drive technology.)

The new Teac drives are much faster than the Shugart SA-400 drives originally shipped with the 8813s and 8810s, so it's possible to modify the ROMs to take advantage of this extra speed. Here are the speed differences:

	Shugart	Teac
Seek time:	40 ms	6 ms
Settle time:	10 ms	15 ms

Notice that the Teac drive steps from track to track much faster than the Teac ("Seek time"), but requires a longer settling time. Since most "seeks" are more than one track, the Teac is still faster on the average.

In version 81 ROMs, the routine that does the required time delays is at address 09D0. It is entered with a value in the A-register, and it will delay for *A* milliseconds (plus a little). Here is the coding for that delay routine:

```
Delay   EI
        PUSH    B
Delay1  MVI     B,8FH
Delay2  DCR     B
        JNZ     Delay2
        DCR     A
        JNZ     Delay1
        POP     B
        DI
        RET
```

This routine is called twice: once for the head step time, with A = 28H (40 decimal), and once for the settle time with A = 0AH (10 decimal). The values of 28H and 0AH are what we wanted to change, to optimize the ROMs for the new Teac drives. These values get loaded by the following instructions:

```
At 09FA:
        MVI     A,28H   (This is 3E 28)

At 0A03:
        MVI     A,0AH   (This is 3E 0A)
```

Now, to change the values in ROM, we obviously need to know where these bytes exist in ROM. We have located the addresses of the proper bytes, but there are three ROMs on the CPU card—in which would we find these bytes?

The ROMs, or PROMs, or EPROMs, are an old style chip that is equivalent to the 2708 EPROM. This part holds 1024 bytes of data. So one ROM must hold all bytes from 0000 to 03FF, the next from 0400 to 07FF, and the last from 0800 to 0BFF. If you hold the CPU card upright, with the gold edge connector toward the bottom, the ROM on the right is 0000-03FF. The one on the left is 0800-0BFF, and this is the one we must change.

We don't actually want to modify the ROMs that Poly provided us. We would prefer to modify them the same way we modify Poly's system disk: make a copy, then modify the copy. In any case, we will need access to an EPROM programmer, in order to "blow" our own PROMs. Removing the left-hand ROM from the CPU card, we insert it into the EPROM programmer and read the ROM into the programmer. Then, using the keyboard on the programmer, we modify the desired locations. Notice that since we took the 0800-0BFF ROM and loaded it into the EPROM programmer, the programmer sees that data as coming from addresses 0000-03FF. (It doesn't know that two more ROMs precede this one on the CPU card.) So, we have to subtract 0800 from the addresses that we want to modify. As far as the EPROM programmer is concerned, then, the data is at these addresses:

```
At 01FA:
        MVI     A,28H   (This is 3E 28)

At 0203:
        MVI     A,0AH   (This is 3E 0A)
```

So, using the EPROM programmer, you should find a 3E at 01FA, and a 28 at 01FB. Change the 28 to an 06. Similarly, find a 0A at 0204, and change it to a 0F. Are we done yet? Not quite. While we're here, let's see what else can be done to improve the system's speed.

Have you ever noticed that when the system accesses any drive for the first time, it takes a lot longer than it does on subsequent accesses? This is because the system steps the drive all the way in to track 35, then all the way back out again, on the first access to any drive. Stepping all the way out to track zero is necessary, and is done according to the manufacturer's recommendations... but stepping in to track 35 is not necessary, and wastes a great deal of time, especially when booting the system. In fact, if the drive happens to be at some intermediate track (say 20) when the LOAD button is pressed, the system will step it 35 times, attempting to step to track 20+35 = 55! The Teac drives make a nasty "clunk" noise when you try to step in this far. So, while we're making changes to the ROMs, let's fix this annoying problem too.

In the same ROM we're modifying, the third CPU ROM, are the two instructions starting at address 0A24:

```
MVI     A,35
CALL    Step
```

Changing the 35 (23H) to some nice low value, like 03, will eliminate the problem of too many steps. This will save an incredible amount of time when booting the system! This 23H is at location 0225 in the third ROM.

We're almost done now. Just one more change to go.

Each of the three ROMs, in version 81, have a checksum byte at the end. The checksum is arranged so that the sum of all the bytes in that ROM will end with FF. (It might be 01FF, or B9FF, or anything ending with FF.) Since we modified the contents of this ROM, we also need to modify the checksum. Certain versions of the Confidence disk will complain if the checksum is wrong.

Here's how we calculate the new checksum byte: The old values in the three locations we modified were 28H, 0AH, and 23H. These add up to 55 hex. The three new values are 06H, 0FH, and 03H, which add to 18 hex. We have lost 3D hex in our total, and this 3D must be added to the old checksum to correct the total sum of the ROM. the old sum, found in location 03FF, was 72H. Adding 3D to this gives AF, which is our new checksum.

Now the copy of our ROM is completely modified. Remove the Poly ROM from the EPROM programmer, and insert a blank 2708 EPROM (available through many ads in Byte and similar magazines). Using the EPROM burner, generate a new EPROM that contains all our desired changes. If the EPROM burner displays the

GOTO 6

Spares *continued*

COMPONENT LEVEL

While component level is the most demanding method of backing up your Poly, it may be the most practical, assuming you (a), have some technical knowledge; (b), own or have access to the most basic kinds of test equipment, such as a volt-meter, logic probe, and an oscilloscope (a $10,000 logic analyzer is nice, but not always necessary); and (c), have the time to do some component swapping.

Most of the spares noted below are more for confidence than anticipation of actual component failure. Short of heavy mechanical or electrical abuse, almost all non-mechanical parts in the 8810 and 8813 should last literally forever. "Confidence" in this case means that you can swap the new part for the suspected old part and verify that the problem *did not* go away, and that you should look elsewhere. (Asterisked part numbers are those which *could* be expected to fail over a 10 year period. Numbers not asterisked are those confidence parts which could but mostly likely will *not* fail over a 10 year period. Parts not listed here which you see on your schematics are considered too low-risk or non-critical. Feel free to add to the "confidence" list.)

NOTE: before thinking about replacing any component, you should verify that the power supply voltages are present and at the correct level. Power supply problems, particularly subtle ones like low voltage or bad regulation, can cause many misleading problems. This is due to the fact that some parts manage to live with bad juice while others do not.

GENERAL PARTS TO HAVE ON HAND: Fuses, +5 volt regulators (assorted sizes), +12 and -12 volt regulators (plus insulating and mounting hardware), IC sockets, 14, 16, and 24 pin; a "junk box" selection of 7400 series TTL chips, signal diodes, tantalum* and ceramic decoupling capacitors, resistor packs, extra interconnect cables (power, video, serial signal); and a clean set of schematics and assembly (or parts locator) drawings. Make sure the documentation matches the revision designation of the hardware *you* own, or that notations have been made on the documentation as to the nature of the revision.

CPU: 8080A, 8251, 9111A, set of ROMs.

Comments : Since the CPU is the focus of the system, it can be nearly impossible to troubleshoot it in a stand-alone mode. A complete, functional back-up CPU board may well be worth the expense. (A human body can do wonderful things missing a lung or kidney, but it is useless without a brain.)

MEMORY: Requirements here depend on which memory board(s) your system is using. Therefore, it is difficult to suggest any particular part numbers. Poly manufactured different memory boards, and virtually any S-100 memory board can be used in a Poly system. But there are some general guidelines that may be of help.

You will probably want quantity two of the memory chip that is being used in the memory array. For memory support chips, check your junk box for the 7400 series chips—you'll probably have what you need. Be aware, however, that dynamic memories typically employ other support chips that may not be as common.

VTI (video terminal interface): 74123, 2A5449 (video out transistor), 74S412 (or 8212), 2111AL-2 (video RAM), and 7400 series chips from the "junk box".

Comments: If the CPU is the brain, the VTI is the mouth. The brain can't tell you what's wrong with the body without being able to communicate with you. The VTI, therefore, is probably the second most important board in the system. Again, consider a complete back-up board.

SERIAL I/O CARD: 1489*, 1488*.

Comments : The two level shifters, 1488 and 1489, have to face the outside world. You may want two of each, particularly if you're doing a lot of printer plugging and unplugging, or if you seem to get a lot of shocks when touching the printer. (If you are using the current loop circuitry, keep a couple of TIL116s on hand instead of the 1488s and 89s.)

DISK CONTROLLER (SSSD): 6852, 8255, 8T97

Comments: This board is very reliable. So reliable, in fact, that almost all of the chips in *PL's* controller are soldered in place. When you have disk problems, first suspect the media, then the alignment, then the mechanical parts of the drive. (Refer to the *Cook's Tour of the SA-400* article in the May/June and July/August 1983 issues of *PL.*)

POWER SUPPLY (8813): high-current diode bridge, 2200 uf*, 83,000 uf*, and 11,000 uf* capacitors; LM323K and TIP640 regulators.

Comments: The capacitors are expensive items and probably don't need to be on hand. (Smaller values can be used temporarily for testing purposes.) However, you should think about replacing them as a matter of course in the sixth or seventh year of system operation. Replace them immediately if there appears to have been any seepage or oozing of liquid from the capacitors. (Liquid or goo indicates an "internal hemorrhage", and a possible threat to the electrical integrity of the system. Do not confuse this phenomenon with the term "leaky capacitor". "Leaky" refers to the electrical problem of the cap being unable to hold a charge.)

For the rectifiers, any general replacement high-current bridge should be satisfactory. Use one rated for at least 10 amps. You can also wire up a bridge of high-current diodes (follow the power supply schematic). Even though the original part numbers on the schematic are different for the two bridges, your replacement bridge should work in either location. Some replacement bridges cannot be soldered to; rather, terminal spade lugs are used to slip over the contacts.

DIAGNOSTICS

Even if you're all thumbs when it comes to troubleshooting, the game is not lost. Poly is one of the few systems (even today) that comes with a useful test package: Our old friends, the Confidence disks. Other systems do a "memory check" on power up, but sometimes the reporting is less than helpful. (*PL* had a memory chip fail recently—the first ever in over five years of system service. The simplicity of isolating the bad chip using Poly's Extensive Memory test and the accompanying chip isolation procedure was remarkable. It took less than four minutes.)

Diagnosing more complex computer problems becomes an arcane art involving science, logic, and occasionally bright intuition or devine guidance. But most pros admit that computer troubleshooting is not that difficult (unlike some problems in television or stereo repair). You can become as clever as the pro by using two methods: selectively removing parts from a faulty system; and moving parts around within the system. For example, if things go awry when the printer is on, disconnect the printer. If the problem persists, disconnect the

serial board from the CPU board. (You can also set the printer driver to "Null".) If the problem goes away but comes back when things are reconnected, assume that the serial card is faulty, or possibly the USART or the -12V regulator. Here's one instance where your voltmeter or 'scope and "confidence" parts come into play. (In this example, it is assumed that the printer cable is ok, and that the printer passes its self-test. Don't forget to check obvious items such as these!)

If the whole system appears dead, a given board may be hanging up the bus. Remove power, pull all the boards except the VTI. Apply power; garbage should fill the screen. Power down again and add the CPU card. Apply power, the front panel display should appear. Power down again, then you can add the disk controller and try to boot. Alternately, add a memory board. Move the front panel window to addresses within the range of the memory board and attempt to write new bytes.

Individual parts can be removed to see if a predicted failure takes place. (WARNING: to avoid possible damage to components, power supply integrity should always be maintained. That is, don't purposefully remove one of the power supply voltages and leave the others applied.)

Swapping is most useful when chasing memory problems. You can swap chips around to verify that the memory problem has moved to the predicted area. If the memory problems are more gross (such as one board pulling the whole system down or creating a block of dead memory), you can exchange the address settings of the boards, assuming you are using more than one memory board, and see where the problem goes. (CAUTION: do not place two memory boards in the system that have the same addresses set.)

And so it goes—there are endless combinations of simple "see what it does now" techniques.

OTHER HARDWARE NOTES

Dennis Cherry, of Dallas, Texas, mentioned two interesting items, one of which you may have heard before but it is worth repeating. The second item was a surprise to *PL,* and is something that few people outside computer design/manufacturing and service circles knew about.

1. +5 volt regulator mask problems: On some Poly boards, the green solder mask extended partly or completely over the of conductive material that is to make contact with the case or mounting bolt of the +5 volt regulator (the +5 volt regulator is the big silver item located either on the top left or top right edge of each board). Under the right conditions, such as a normal temperature change due to operating conditions, the mask will completely insulate one connection of the regulator, thus switching it "off". The most common result is that the system will suddenly go deaf and blind. You may want to unbolt and then unsolder your regulators and check the condition of the mask.

2. Corrosive chip storage foam: when conductive chip carrier foam was first invented, it was a major relief to the chip manufacturing industry, not to mention distributors and system manufacturers. Sensitive chips could be shipped and handled without worrying about blasts of damaging static. Unfortunately, that early foam had an unintended chemical side-effect: it severely corroded IC leads! You can see evidence of this if any of your chip leads have what appears to be black goo or blackish mottling on them. If the chip is soldered in place, there is probably nothing to worry about. If the chip is in a socket, definitely clean the leads using denatured alcohol or a contact burnishing tool. It may be wise to replace the socket.

Of course, the time to build your spares kit is now, *before* something happens. Unfortunately, many of us (your editor included) are like the grasshopper character in the fable about the grasshopper and the ants. At the very least, you may want to spend a few hours one Saturday checking the stock of the local Radio Shack or Jimpak dealer. They will probably have 95% of the parts noted above. The other 5% (especially old memory chips) may be very hard to find. Look now while you have time.PL

POLYLETTER, May/June, 1984
Editor and Publisher: Frank Stearns, PolyServe
Contributing Editors: Bob Bybee, Don Barrett
Photos: Bob Bybee
Subscriptions: (Domestic) $15.00 per year; (Canada) $18.00 per year, payable in US Dollars; (Overseas) $20.00 per year, payable in US Dollars.
Editorial Contributions: Your contributions to this newsletter are always welcome. Articles, suggestions for articles, or questions you'd like answered are readily accepted. This is *your* newsletter, please help to support it.
Advertising (camera-ready or PL typesetting): $2.00/column inch; full column: $18; full page $30 (Personal ads are still free of charge.)
PolyLetter is published bi-monthly by PolyServe. PolyLetter and PolyServe are not affiliated with PolyMorphic Systems.

For Sale: 8813, 48 KB, keyboard and monitor, 88MS SSSD 8 inch drives. Completely refurbished by PolyMorphic Systems. $1500 plus shipping. Call Steve Rooney 412/486-8166 M-F 9am-5pm EST; or 412/486-7679 evenings.

WANTED: Poly's combination monitor/keyboard in one chassis. Please contact Bob Kelso, 11616 Industriplex, Baton Rouge, LA, 70809. 504/924-4608

WANTED: 2108 RAM chips for the "EXTENSYS" series RAM cards. Need the "uppers", S1573 or S1626. Contact PolyLetter, 14307 NE 16th St, Vancouver, WA 98664, 206/892-3970.

And the Bit Goes On...

Don Barrett, in pursuit of that elusive 8th bit through Poly's printer driver, finally made it. He encountered more twists and turns than *PL* had anticipated, but Don was persistent, to say the least. Don made the following modifications to enable passage of the 8th bit:

Sio.PS (Version 42): Once the printer driver is loaded, change the contents of the following memory locations: 302E from DA to 5E; 30DB from 7F to FF; 2F17 from 7F to FF; and 2F54 from F2 to C3. One additional location should be changed but only from BASIC using a POKE statement: change location 2FE8 from C2 to C3 (12264 from 194 to 195 in decimal: POKE 12264,195). Once you're through (we're assuming that this entire exercise has been done to support the graphics mode of your printer while in BASIC), return the value to C2 (194 decimal: POKE 12264,194).

Don said he had to define a new printer using Setup.GO; the lines-per-page specification was set to 255. **NOTE:** Invoke this printer driver *before* implementing the above changes. To cancel all the changes, simply reload the default printer driver or your standard printer. These changes should be cancelled when the FORMAT section of WordMaster is used.

PL has a suggestion on how you can implement these changes: write a command file to load the desired printer driver (\$Pr NEW-NAME), then call BASIC and perform the pokes (don't forget to convert the hex values to decimal).PL

WHAT'S IN A ROM?

Bob Bybee, Poly Peripherals

In past issues of PolyLetter we've discussed the ROMs on the CPU card, and how to find the version number of your ROMs. Most of your systems contain version 75 of the ROMs. That was the version which supported the 88/MS, and the MS was Poly's biggest disk storage device until the hard disks became available.

You've heard us mention that ROMs version 81 are required in order to use the hard disk or the TwinSystem. What exactly does that mean, and what's new about version 81 of the ROMs?

First let's understand exactly what a ROM is. It stands for "read-only memory," which means it can't be modified. It also doesn't lose its contents when the power is turned off, unlike the RAM in your Poly. When the system is powered-up, the only "intelligence" in the machine is contained in a program in the ROMs. The program is about 3000 bytes long. This program must be smart enough to get the system started, run the disk drives, read the disk directory, and load the operating system from a disk. No small task!

When the storage device in the Poly system consisted only of 5 inch drives, the ROMs had only one type of drive to work with. When the 8 inch drives were added, PolyMorphic's designers realized that some systems would have only 8 inch drives, some would have only 5 inch drives, and some would have both. The system needed to be able to "boot" from either type of drive. This meant the ROMs had to know how to access both drives.

So the program in the ROMs would first try to boot from drive 4 (an MS), and if it couldn't, it would then try to boot from drive 1. If you have only 5 inch drives on your system, you might not have realized that every time you press LOAD, the system tries to boot from a non-existent MS!

As with all computing resources, we eventually run out of ROM space. If you've looked at your ROMs with SuperZap (memory addresses 0000 to 0BFF), you've seen that there isn't much empty space in them. But like other computer manufacturers, PolyMorphic realized the need for hard disk storage. Could the ROMs be modified to run a hard disk in addition to all the other types of disk drives?

The answer is no. There was simply not enough space left in the ROMs to add anything as complex as a "device driver" for a new disk drive. Poly's designers had to get clever to work their way around this problem! They decided that every system would need a floppy disk anyway, even if it had a hard disk as the main storage device. So it would be logical to "boot" from a floppy, load the necessary hard disk software from that floppy, and then use it to access the hard disk. Thus was born *Volume Manager.*

But if the hard disk software is not in ROM, then what *is* different about version 81 of the ROMs?

All programs that read or write the disk will eventually make use of a routine called *Dio.* Dio is located at address 0406 in the ROMs. In older versions of the ROMs, location 0406 simply jumped to the actual disk input/output routines, which were also stored nearby in ROM.

0406 — ROM-based disk I/O

In version 81, an important change was made. Any time a program went to Dio at location 0406, it was "vectored" (jumped) to a location in RAM instead of going directly to the disk I/O routines. That location in RAM would contain another jump instruction which would then lead to the disk I/O program.

0406 — 0C44

0C44 — ROM-based disk I/O

Is this inefficient? Not really. One extra JMP (jump) instruction is all it takes, and that's a very small percentage of the number of instructions required to actually do the disk I/O.

So what did we gain? Since location 0C44 is in RAM, we can modify it. It starts off pointing to the ROM-based disk routines, as shown above. But once Vmgr.OV is loaded, location 0C44 jumps into Volume Manager instead.

0406 — 0C44
0C44 — Volume Manager
Vmgr — ROM-based disk I/O or hard disk I/O

Now, any call for disk I/O goes to 0406, then to 0C44, then to Volume Manager. Volume Manager can then decide whether that particular disk I/O request is for the hard disk or for floppies. If it's for floppies, Volume Manager simply jumps back into the ROMs and lets them do what they've been doing for years. But if it's a call for the hard disk, Volume Manager decides which "volume," or portion, of the hard disk is to be accessed. Then it actually performs the disk read or write, using the device driver that was loaded into memory from the Driver.DD file.

It is obvious that Poly's designers intended Volume Manager to be an "open" system, one that other programmers could use and enhance.

This mechanism adds great flexibility to the operating system. You may recall that Poly's initial hard disk system was an 8 inch Priam disk, but now they are offering a 5 inch winchester drive and a removable cartridge drive as well. The ROMs weren't affected by these latest two drives; only the Driver.DD file had to be changed. Volume Manager has stood the test of time.

Volume Manager was designed to make such alterations simple. Section 10 of the System Programmer's Guide contains a complete description of Volume Manager, and even includes a discussion of how to add new devices (disks) to the system. It is obvious that Poly's designers intended Volume Manager to be an "open" system, one that other programmers could use and enhance. This "open" approach also characterizes most of the systems on the market today, including the IBM-PC. (The manual for the PC even contains a listing of its ROMs. But then, so did the manuals that came with my first Poly-88.)

PL

continued from page 2

checksum of the completed ROM, you should see that it does end in FF, just as it should.

This article has given the details of upgrading an 8810 to a two-drive system. The mechanical changes were necessary, but the firmware changes are what really make the modification shine. In the process, we've shown some of the things to be considered when you make changes in the operating system ROMs. (These particular changes are for ROMs version 81, for 5" SSSD disks only, and wouldn't be valid for any other version of ROMs.)

Don't forget that PolyMorphic is now offering an upgrade kit for 8810 owners, which include the drives and all mechanical parts. If you're not sure of your mechanical abilities, you might want to purchase the necessary parts from Poly.

The completed two-drive 8810.

PL

Harrumph. Nobody took the challenge to name the *PL* bird—it is assumed the meager prize had something to do with it. Sorry, folks, cars and trips to Hawaii are out. A free Disk of the Month or subscription is the best we can do. (Just think, if *you* had entered, you would have been the winner by default.) The contest is extended until the closing date of the next issue, 7-5-84.

PolyLetter

14307 NE 16th St
Vancouver, WA 98664
206/892-3970

FIRST CLASS MAIL

PolyLetter

The Newsletter for PolyMorphic Systems Owners and Users

July/August #8404

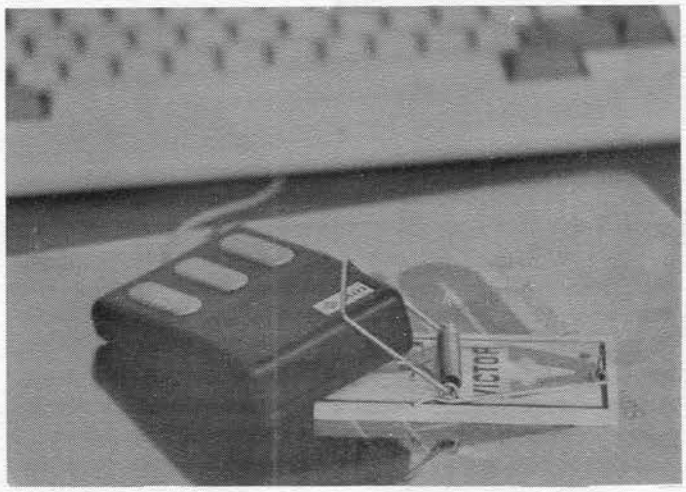
NEWS

CHEAP (BUT GOOD) DISKS: *PL* has always been outraged at the prices charged by most places for quality diskettes. (Strike "quality" and insert "name brand"—the two have little to do with one another as *PL* has discovered over the years.) But wait! There is hope. *Transaction Storage Systems* (TSS) has been quietly selling Control Data disks mail order for the past few years. The CD disks include hub rings and 40 track SSSD certification. *PL* has bought over 50 of these disks, and found them to be reliable and quiet. (We always get nervous when a we insert a brand new, top grade, "major brand" disk, access it, and it sounds like we're in a furniture refinishing shop. CD disks are mechanically very quiet.)

The best is yet to come: from TSS these disks are about $2.15 each (5 inch hard sector) in lots of 10. But there's more—for a limited time TSS is tossing in two free disks if you order 10, and 10 free disks if you order 50. They're also giving away "diskbanks" with every box ("...a $7.00 value..."). Give them a try: 1-800-FLOPPYS or 1-800-521-5700 or 313-557-3036.

NEW SYSTEM: PolyMorphic Systems is no longer giving anticipated shipping dates for the new system. It seems that the industry-wide parts scarcity problem has just now finally eased. This problem was not only in processors, but TTL chips as well. (Our local Jimpak dealer was even yelling about the shortage and cost of TTL.) Small companies such as PolyMorphic Systems were low on the list to receive parts. Hardware hasn't been the end of it either. Even before release, the system has undergone software upgrades, based on Digital Research's sudden changes in the new Concurrent CP/M.

PL has grown numb to the whole subject, and no longer has any recommendations. All we can say, however, is that from our observations here in the "Silicon Forest" this kind of product development cycle is not unusual. Whatever the reasons, everybody involved—manufacturer, dealer, potential buyer—all grow aggravated and frustrated, but it takes time to move a mountain when all you can get your hands on is a spoon. We can hope at this point that Poly will have already gone through revisions "1.0 through 2.9" *internally*, and not force such on the end-user. Will it be worth the wait? We can answer that with another question: What about those who rushed to the 8088-based machines (IBM, Compaq, etc...)? Once the new generation systems are out (such as Poly's) you will hear some very loud screams from those with machines based on what some analysts are calling "interim" or more unkindly, processors "hurriedly designed around kneejerk responses to



Snap!

THE BIRD HAS A NAME.

PL was about to give up hope that the old gander would ever get an official name. Our thanks to all the Poly owners who got into the spirit and sent some wonderful names and great stories (some of which cannot be reprinted).

First, a sampling of the entries, and again our thanks to:

Ron Moffatt, Rochester, NY, and his apt name if the bird *never did* get named and was left to obscurity: "Polygon".

Chuck Thompson, Dallas, TX, and his entry of "Splinter", a name which probably reflects the group of people still willing to own and use such an old system;

Charles Mach, Irving, TX, and his story about Poly's history (side bar). Mr. Mach opened his letter: "Dear PolyLetter, since in the last issue of PL you gave us readers the bird for ignoring the bird, I decided to write back..."

Bill Davis, Portland, OR, and his tongue-in-beak entry "Poly *wannacracker* ";

Michael Aquino, San Francisco, CA, and his "unofficial" entry of "Mr. Squock";

Ralph Kenyon, Chester, MA, for his dozen plus entries (plus explanations), not the least of which included "Exec, *in honor of Poly's operating system* ", "PolyAnna, *bright, optimistic, hopeful one* ", and "FlockType, *an encoding of* many forms *symbolic of PolyMorphic* ";

and Al Levy, Hicksville, NY, and his rather droll entry "Logo".

So finally, the winner:

WordMaster Tricks

Word-processing on Poly is probably the weakest area of the 8813 software—it always has been and probably will continue to be. While the editor is simply wonderful, there are one or two features that would make it downright spectacular: autowrap for one thing, seeing files formatted on screen immediately would be another. However, considering the "general" nature of the editor (and formatter)—and system architecture on the whole—implementing features such as these would be somewhat difficult. So, we have to live with them. But there are ways of "seeing things" on screen more or less as they will appear on paper.

There are two levels at which you can attack this. First, we know that the formatter is designed to read immediate format commands serially. That is, if the system sees a "paragraph" command, regardless of how it is placed in the text, the line will be broken, justified left, and the text following the command started two lines down, and indented five spaces. Could we not do the same thing in the file as we edit it, thus giving us some idea of how the paragraph break will look?

```
allows upward expansion of the units
within the system.{tind}
____Applications for the drive
include word-processing and text editing
```

Note that the paragraph command is the last item on the line before the break, and that a few leading spaces have been inserted before the next line starts. (The leading spaces are represented here by underscores. In an actual file, you would use spaces, not underscores.) None of this really bothers the formatter; it skips the spaces and sets the correct indent. More often than not, though, two carriage returns in a row are passed literally. (In fact, this is a feature of WordMaster II. It saves using a great many skip commands.) For this reason, a second return plus the paragraph indent produces two blank lines instead of one. So it would be best *not* to add the second blank line. But just by placing a command visually "out of the way" and adding a few leading spaces, a great deal has been done to make a text file appear on screen, in the editor, more as it may on the final hardcopy. But who wants to spend a great deal of time carefully placing and aligning format commands? Key definitions, with all the graphic cosmetics built in, can be used.

We'll return to formatting on screen in a moment.

Key Definition Tricks

We all know how useful the editor's key definition utility can be. We could, for example, define the "p" key so that when called (ESC-p), the paragraph command would appear, followed by a return, followed by four leading spaces. Even though five leading spaces is the default amount used by the formatter, it seems out of proportion with the 64 character-wide screen to use more than four leading spaces.

Here's what that definition looks like:

```
^[=p{par}^m____^[
```

A press of the control key can be represented in the editor's key definition string by using the carat symbol. (A carriage return is really a control-M.)

With this definition trick in mind, we can do some other interesting things. Do you write a great deal of dialog? You've probably found that entering the necessary WordMaster commands almost more trouble than writing the quote itself.

Here's a very handy key definition string set to the "q" key. Use it for entering a temporary indent, a return, and the quotes. Four spaces lead the text so it looks nice while on screen (and this doesn't bother the formatter); the opening and closing quotes are printed; and the cursor is backed up one space so that it's *inside* the quotes, ready for you to write the dialog. No shift key and no furious cursor-key pounding is required:

```
^[=q{tind}^m____""^m^t^t^[
```

If you're writing conversation, this is invaluable!

Entering the Key Definitions

Remember that any file name in the form "EXtensiondef" with the extension of .ED, such as *TXdef.ED* or *BSdef.ED*, becomes the keyboard definition file that is loaded automatically by the editor. Which key definition file the editor loads depends on the extension of the input file that is being edited. Thus, editing or creating text files (.TX extension) loads TXdef.ED, if present. (If no extension is assigned when creating a new file, the editor assigns the default extension of ".TX".) If you're editing BASIC program files (.BS), a key definition library file with the name BSdef.ED loads instead (if present).

The point to remember is that you can create and

GOTO 5

continued from page 1

MR. SQUOCK

PL is sure Dr. Aquino is not expecting this win; in his letter he told PL that he'd "given up" trying to think of a suitable name. He had apparently not cared for *Mr. Squock* or the others he'd thought of. But congratulations, anyway, Mike! May Mr. Squock Live Long and Prosper. PL

In reflecting back on the Poly bird's probable heritage, my imagination discovers the following story: The parrot's full name is *Polygrammer*, but he also goes by his Americanized name of Poly-G. P-G was born some years ago in South America. As he grew, he discovered his natural ability to pick up multi-languages. He become *poly-lingual*. P-G soon heard of the business called data processing, where wonderful machines would accept willingly all the input Poly-G could output. Poly-G quickly immigrated to that hot-bed of D. P. activity—the United States. He settled for a time in Atlanta, GA, because the abundant vegetation reminded him of home.

Poly-G was somewhat small of being, so he shopped around for a computer of his scale—micro computers, of course. Poly-G soon noticed an outstanding poly-talented computer with a fetching name. Poly-G had discovered PolyMorphic. It was love at first IPL. Poly-G went on to become proficient in a number of programming languages and thus settled on the professional name of Polygrammer. Friends shortened his name to Poly-G in the American inclination for nicknames. This is the story as best as my imagination can remember it.

-- Charles Mach, Irving, TX

Truth? Legend? Sounds good to us until someone else comes up with an alternate explanation of the origin of the bird. PL

John Warkentin provided the following review of his new Toshiba P1351 printer. PL has taken John's review, which was printed on the P1351 printer, and reproduced it directly here. Sometimes these reproductions do not work well. If it doesn't, PL can testify that the original looked good.

A REVIEW OF THE TOSHIBA P1351 PRINTER

by

John J. Warkentin

For a few years I was in the somewhat enviable position of having a Qume Sprint 5 printer on semi-permanent loan. When it returned whence it came recently, I was left with a need for a letter-quality printer. My research led me to the realization that printer technology has made great advancements in recent years, and that there are now dot-matrix printers on the market with quite good character quality.

After seeing samples of output from various printers, I determined to purchase the Toshiba P1351. The manufacturer claims compatibility with the Qume Sprint series, plus the capability of down-loading fonts. Examination of the manual revealed that the command set does indeed emulate that of the Qume Sprint, with 1 exception-- print-hammer intensity cannot be controlled. This is not too important with a dot matrix printer, as it is a feature that was originally provided to allow the user of the Qume to adjust for the variations in the character footprints (compare that of the 'M' to that of a '.').

General printer specifications are: weight: 45 pounds, measurements: 24"w x 15"d x 6"h. Print speeds vary according to the font in use and character density. Some representative values are: Draft font at 10 characters per inch-- 160 characters per second; Elite font at 12 characters per inch-- 93 characters per second.

The printer I purchased came with a serial interface to match my Poly system 8813, but the P1351 is also available with a Centronics-type parallel interface. Building a cable (none is provided) was not difficult, as the manual provides plenty of information. I chose to implement the hardware handshake protocol, but X-on/X-off and ETX/ACK protocols are also available.

A nice thing about the manual is a section called "Quick results from your P1351 printer". It gives step-by-step instructions for initial testing. When I followed the proceedure, nothing happened! My wife said, "pack it up and ship it back." Well, that seemed a bit extreme, so I went looking for interlock switches, and found one for the cover. Unfortunately, it was wired upside down so that closing the cover disabled the printer. After rectifing that, I got "quick results".

I then sat down with the manual and went through the internal switches, checking and changing as needed. The switch settings in the manual are marked to indicate the factory default settings, but I found a couple that were set otherwise. This activity also uncovered 2 defiencies: the manual does not reveal the location of the switches or how to gain access to them, and the location of the switches is such that you don't really want to do this very often.

GOTO 4

PASCAL PASTURES

When undertaking any new language, it is helpful to look at the same program done in two languages: the new language, and an old, familiar language. (Hopefully BASIC is old and familiar.) While most of the rules and subtleties are lost in this first presentation, you may begin to see some forms and the gross differences. We'll call out the major points in a moment. First the programs:

BASIC

```
100 INPUT N
110 S=0
120 FOR K=1 TO N
130 INPUT X
140 S=S+K
150 NEXT
160 PRINT "AVERAGE =",S/N
```

PASCAL

```
PROGRAM average(input,output)
VAR
  n:integer;
  k:integer;
  s:real;
  x:real;
BEGIN
  real(n)
  s:=0;
  for k:=1 to n do
  begin
   read(x);
   s:=s+x;
  end;
  writeln('Average = ',s/n);
END.
```

The obvious difference is that not a single line number is used in the Pascal program. While this example does not show it, there are some major advantages in *not* using line numbers. You can name something—a subroutine or a vector point—with any mnemonic that fits the task at hand. It is not necessary to recall an arbitrary line number.

Pascal begins with a series of *declarations* under the heading *VAR* (variables). At first one wonders, "doesn't the Pascal program know what's going on?" No, not unless you tell it. This may seem like a pain at first, but later we'll discover that you can create your own custom variables, designed to handle whatever complex chores you can think of. In a round about way, this leads to the important point that Pascal is able to manipulate *sets*, something that BASIC can't really do. (Yes, one can do a limited number of arithmetic operations with arrays, but the applications are very narrow and the rules rigid. More about this as *Pascal Pastures* progresses.)

We see that Pascal is not case-sensitive—upper and lower case commands can be mixed—though this capability ultimately depends on the compiler. There are other minor differences between the two example programs, and the spotting of these can be left as an exercise for you.

But let's place the above programs out of the way for a moment, and look at Pascal from a larger view. There are several important things to keep in mind. Pascal is well suited for the larger, more complex programs—but if you need only a dozen or so lines of code for a given item, stay with BASIC. Pascal is wonderful in that you can start with the simplest blocks ("procedures") and string them together. The ultimate goal of any Pascal programmer, regardless of the magnitude or scope of the program, is that the finished *program* should not be

GOTO 6

continued from page 3

The complaints fade into significance, though, once the printer starts doing its job. It is FAST. The high quality fonts look good, although the discerning eye might be able to tell that it is not daisy-wheel output. Here are some samples of the print:

Draft font at 10 characters per inch.
Draft font at 12 characters per inch.
Courier font at 10 characters per inch.
Courier font at 12 characters per inch.
Elite font at 10 characters per inch.
Elite font at 12 characters per inch.

Some of the features of the printer that the ads don't talk about are: a four thousand character buffer, a paper-out switch, or that there are 3 fonts permanently part of the printer, and space for 2 down-loadable fonts. You can issue commands to switch between fonts while printing.

There are several vendors of diskettes containing several fonts, although none of the diskettes are available in the PolyMorphics format.

In closing, the Toshiba P1351 is a fast, versatile printer, well worth the price.

continued from page 2

modify the keyboard definition files from the editor just like any other text files. You need not create and maintain definitions indirectly through the editor while editing some other file. Just be sure to follow the examples of the definition strings carefully. You may want to create some definitions while in the editor (ESC=key-to-define), save them to a definition file (ESC-control-W, filename), then *exit* the editor and edit the newly-created key definition file, looking carefully at the control character sequence. (If you want to load a different definition file from within the editor, enter the editor and type ESC-control-L, filename.)

Below is one of PL's most frequently-used key definition files. See an interesting definition? Key in it and try it. (Spaces are represented by underscores).

```
^[=B{br}^M^[
^[=b^m{bpg}^m^m^[
^[=d{dind}^M^M{nind}^M^Q^Q^T^T^[
^[=e{nfil}^M^M^[
^[=f{ne 4,fjl}^[
^[=F{ne 4,fill}^[
^[=h{he}{jl} {pno}^M{skp 2}{end}^M^Q^Q^T^[
^[=i{ind}^M^M{nind}^M^Q^Q^T^T^[
^[=L{ne 4,bold}^R{nbold}^M^Q^T^[
^[=u{ul}^M{nul}^m^q^t^[
^[=v{ne 5,ce}^M^M{fill}^M^Q^Q^T^[
^[=w{wund}^T ^[
^[=.{point} ^[
```

Printing to the Screen

Let's return to synthesizing on screen what the document may look like on paper. With format commands massaged by carriage returns and spaces, you're part of the way there to simulating on screen what the document will look like before going to hardcopy. But that's just while you're in the editor. Better still is to route the text through the formatter and send it to the screen, though we're going to be hurt by the 64 character screen width limit and the 16 lines per "page" vertical limit. Still, a dump to screen can be very useful for catching formatting errors, and for getting a rough idea of how your document will "block".

Place the following instructions into a file named PSCR (or whatever); it will become a command file:

```
Pr Screen
FORMAT ENVSCR (?)    (don't enter a return here)
```

When this command file is called, it will connect the screen to the printer driver and tell the formatter to use an environment file called ENVSCR (we'll get to that in just a moment). As a convenience, the file search facility (bracketed question mark) is also included. All you do is type the name of the file you wish to format and press return.

You'll see your document on screen, lines zipping by, stopping only at the end of each page. What good is this, you ask. You can't actually "read" anything but the last few lines of each screen. True, but save your reading for the editor. The purpose of this, as you'll see after a few times tries, is to give you a *sense* of what the document will look like on paper.

You don't have to override the printer driver's screen parameters as outlined in ENVSCR below, it's just easier this way if your document has headers and footers, which can burn up six lines of the sixteen per screen "page".

If you want to know *exactly* how the final document will come out (say you want a page count), you can alter the "wid" specification in ENVSCR to reflect the printer's line length, less any left offset. In this case, you will have a line wrap on screen which will make things even harder to read, but the technique is still useful. (The need for changing your "wid" value assumes your printer's page width specification is greater than 63.)

ENVSCR

ENVSCR (environment screen) does two simple things: it temporarily changes the printer driver to whatever line length we need for screen use (63; don't use 64 or you'll have what appears to be double spacing), and it forces, at your option, whatever lines per page specification you'd like—you aren't stuck with 16-line pages, which may provide pages so small you can't tell much about the page-level formatting of your document. Here's ENVSCR for a screen-width dump:

```
wid 63 (you can change this to match your printer)
lpp 66
rm 0
lm 0
```

Don't forget: when you're ready to dump your files to the printer, reconnect the printer driver to whatever printer you normally use (type "$Pr NORMAL"). Then type FORMAT (FILENAME). Everything will be back to normal.

PL has found the above techniques useful in the process of creating documents, especially large documents. We hope you will find them of value too.

PL *readers are invited to share their WordMaster tricks with their fellow readers. Drop us a line, or call.*

PL

NEWS continued from page 1

volatile market forces".

PL's closing editorial comment? All chit-chat aside, the new system had durn well **better** be worth the wait.

CORRECTIONS: Ralph Kenyon spotted the following flubs in the last issue of PL: ESC-CONTROL-K, reported in the May/June 84 PL as having appeared only in editor versions 4.0 and later, in fact appears as early as version 2.0. In the *NEWS* column of that same issue, regarding Ralph's "Superdisk" product, the drives Ralph interfaced were 460s (full height) not 465s (half-height) as reported, and that the *reconditioned* price of the 460s were $159 each. Ralph also wanted to make it clear that he has the software to write either double-plus or the standard Poly single density format (version 81 PROMs are required).

NEW ADDRESS FOR ABSTRACT SYSTEMS: Ralph Kenyon has moved. His new address and phone number: Ralph Kenyon, Abstract Systems etc., 191 White Oaks Road, Williamstown, MA, 01267; 413/458-8421.

Sidney F. Mullen, of Pittsburgh, PA, sold his Poly and has about 23 "top quality" used 5 inch SSSD disks for sale (Dysan and Verbatim Datalife). $25 takes all. Business phone: 412/462-6421 (Monday-Wednesday, Friday).

continued from page 4

longer than one page, and it should contain no GOTOs of any kind! On the face of it, particularly if you have written one and two thousand line (and up) programs in BASIC, that goal seems ludicrous. And yet, it isn't. Stop for a moment and think about the initial design phases of a major BASIC program. At that starting point you are probably thinking very abstractly, saying to yourself we'll start here, do this, go here, come back, finish this, update this, and be done. You realize that you'll have to write many, many lines of code to fill in all those generalized statements of activity. Using the words and phrases that come to mind, wouldn't it be nicer to simply write your abstractions and their natural flow as a computer program? This is exactly what Pascal allows you to do, though there are some rather tight rules of punctuation and grammar.

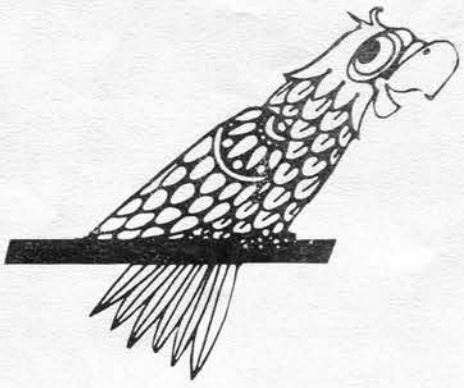
But now, once the main thrust is done, you're still faced with the awful task of creating the guts to make the whole thing fly, right? Not exactly. First, you won't lose that wonderful verbal flow that is the program. (Yes, in BASIC you can use GOSUBs, but you still can't refer to the GOSUBs with mnemonics, and the GOSUBs—packed and stacked row after row—tend to look alike when you're a little bleary-eyed.) Secondly, you can stay in an abstract frame of mind for a while yet. All you need to do is concentrate on one of the large areas dictated by the main program, and let it flow through your mind. In turn, you can break down its elements, verbalize them, and so on. For each part created, you can use whatever convenient (and mnemonic) English language construct *you* wish.

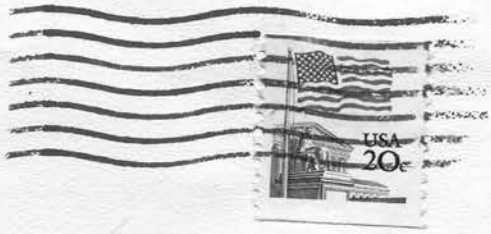
Hopefully, you begin to get an inkling of the power of a truly structured language. However, Pascal is not without its problems. These will become evident as this series progresses. PL

Dr. Michael Aquino, San Francisco, suggests the following source for "weird parts", especially old memory chips: Component Sales, Inc., 778-A Brannan Street, San Francisco, CA, 94103, 415/861-1345. Mike says an old RAM chip of his died recently, and it seems that Component Sales "has every chip ever made."

POLYLETTER, July/August, 1984
Editor and Publisher: Frank Stearns, PolyServe
Contributing Editors: Bob Bybee, John Warkentin
Photos: Bob Bybee
Subscriptions: (Domestic) $15.00 per year; (Canada) $18.00 per year, payable in US Dollars; (Overseas) $20.00 per year, payable in US Dollars.
Editorial Contributions: Your contributions to this newsletter are always welcome. Articles, suggestions for articles, or questions you'd like answered are readily accepted. This is *your* newsletter, please help to support it.
Advertising (camera-ready or PL typesetting): $2.00/column inch; full column: $18; full page $30 (Personal ads are still free of charge.)
PolyLetter is published bi-monthly by PolyServe. PolyLetter and PolyServe are not affiliated with PolyMorphic Systems.

PolyLetter
14307 NE 16th St
Vancouver, WA 98684
206/892-3970





20

PolyLetter

The Newsletter for PolyMorphic Systems Owners and Users

September/October #8405

NEWS

Mid-November: by now the late autumn air is very crisp, the last of the bright but fading colors have drifted earthward, huge amber moons will later appear three or four nights in a row, and maybe, just maybe, the new system from Poly will be released.

In a recent interview with Ken Gudis, "middle November" seems to be the magic date, with the following disclaimer: "provided nothing goes wrong." The new system is out being beta-tested now, but by whom and for how long was not specified. Still more software is being ported, and yet another DRI release of the operating software (3.2, now called "PC-Dos") will be the cornerstone. This is reportedly the same Digital Research operating software AT&T is using on their system. IBM, rumor has it, may also use this software.

The additional delay, designed in part to accommodate the latest software, will give the new Poly a still broader base of software. Some Poly observers feel PolyMorphic has thus changed an earlier stated market strategy, that of defining and going after "narrow vertical markets".

Several major design changes have been made to the new system, the most important of which is the abandonment of the S-100 bus. This means that the new system will *not* plug directly into your 8813/10 chassis. However, Ken assures us that the appropriate network interface software is being written to make the older systems able to communicate with the new system. The word "workstation" or "smart terminal" comes to mind, but it remains to be seen how 80x24 display information will be sent to an old Poly.

Apparently, there were not enough circuit lines in the S-100 standard to do all the things Poly wanted to do. However, we feel uneasy that Poly will be using a "proprietary bus" in place of S-100 or some other standard. But Ken says that there will be different backplane options. You could, for example, specify how an extra slot is to be configured. You could ask for an IBM-PC type of slot. In addition, Poly will be using Intel's ISBX add-on bussing system. But the strongest feature to counter potential problems here may be the parallel port and the four serial ports, configurable for RS-232 or 422. (232 has a typical speed limit of 19.2 Kilobaud; 422 can go as high as 2 Megabaud.)

But whatever—all one has to do is leaf back through a few *years* of PolyLetter and look at earlier new system references -- many dates have come and gone. We all, it's safe to say, are smiling lopsidedly, and will believe it when one is in our hands.

8813 Pascal has been shelved for the time being, and with it, PolyLetter's *Pascal Pastures* column. Understan-

GOTO 5

As mentioned in the January/February issue, PL *had planned a "How it Works" series, which, in laymen's terms, gives you a better idea of how your system actually operates.* PL *feels strongly that we should never take our systems for granted, especially as they get older. Here then is the first in the series, presented by Bob Bybee.*

HOW IT WORKS—THE VIDEO TERMINAL INTERFACE

by Bob Bybee, Poly Peripherals
1437 Sugarwood Lane
Norcross, GA 30093

The Poly Video Terminal Interface, or VTI, is a card you can find lurking in the back of your chassis. The VTI connects to your keyboard for input, and drives a video monitor for output display. This card was designed as part of the original Poly-88 system, and has changed little since those early days.

Most early microcomputers (circa 1976) required a separate terminal to be connected to the system, using a serial port. The user of such a machine had his choice: a slow, noisy, expensive, printing terminal like a Teletype; or a faster, but even more expensive, video terminal. Poly's designers came up with a third alternative: a single S-100 card to provide all input and output functions, just like a TTY (Teletype), but much faster and cheaper.

The VTI actually contains two separate sections: the keyboard interface and the video display logic. We'll begin with the easier of the two, the keyboard.

Your Poly keyboard has a parallel interface which connects to the VTI with a ribbon cable. Over this cable, the keyboard transmits eight bits of data and a "strobe" signal. This strobe is a pulse, automatically generated whenever a key is pressed. If a key is held down, the strobe continues, causing the key to auto-repeat.

When the VTI sees this strobe, it captures the eight bits of data from the keyboard, and interrupts the CPU card using a pin on the S-100 bus. Then, when the CPU is ready (usually after the next instruction), it notices the interrupt signal and services that interrupt. The CPU reads the keyboard interface to gather up those eight bits the keyboard generated. This data is then stored away in RAM until some program asks for keyboard input. This is why you can "type-ahead," even when the Poly is not yet asking for a new input line.

The other part of the VTI, which actually occupies most of the card, is the video display interface. To the CPU, it looks like a 1024-byte chunk of memory. To the video monitor, it provides a (more or less) standard

GOTO 3

Ralph Kenyon tells us something about installing and running 1600 sector 5 inch disks in his 8813.

96 tpi PROMS (96ROMS)

Ralph E. Kenyon Jr., Abstract Systems, etc.
191 White Oaks Road
Williamstown, MA 01267

By upgrading my system to double sided, 80 track (96 tracks per inch) drives, and by using all 80 tracks, I have converted my 8813 system from 350 to 1600 sectors per drive, or 1.2 megabytes of online storage. I have also gained a tremendous increase in speed as a result of the faster motor start and step times.

The drives I am using are Shugart SA-460s which are 5-1/4 inch, double-sided, 96 tpi (tracks per inch). I have strapped them with MS (motor on from drive select) and SD (side select using direction select). I rewrote the ROMs to use the fact that I don't have to tell the drives to turn the motor on, as well as to increase the track number to 80 and to change the step time to six milliseconds and settle time to 15 milliseconds. It also takes advantage of the 200 millisecond motor start time. I also improved the sector access algorithm as well as sector error checking.

INITing 1600 sectors takes two minutes, 12 seconds, IMAGing 1600 sectors takes three minutes, 22 seconds. I recall that the 88-MS took 10 to 30 minutes!

In Poly's SD ROMs, an allowance of one third of a second is made to change drives. One full second is allowed to cold start a drive. The SA-460 comes up to speed five times faster. While starting up, its average speed will be at least 150 rpm (half final speed of 300 rpm). Since the curve rises rapidly and levels off, 200 rpm is a better average. During this time, the disk will have rotated through between six and eight sector interrupt pulses, so the presence of a disk in the drive can be checked while waiting for it to start up. Poly's old ROMS required at least five sector interrupt pulses to check for the presence of media. Because both cold startup and drive switching require checking for the presence of a disk by testing for some sector interrupts, one fifth of a second represents a reasonably fast time for doing both. I elected to turn the drive motors on only when the drive was selected. The time to check for the disk in place corresponded to five sector interrupts, and that is only 30% less than the time it takes to start the media rotating. Even in the worst-case motor startup of 53% of a disk revolution, there will be five sector interrupt pulses. By having the disk drive stopped when not accessed, head and media wear is reduced. Also, the SA-460 drive heads load when the media is loaded, so there is no head load delay, and no head "clicking" noise. In addition, ROM code is saved by not having to keep track of the motors.

While implementing this change I discovered a rather insidious bug in the existing ROM code. With higher track densities on the SA-460, there may be more noise for the disk controller to deal with.

The problem was insidious, in that the incorrect sector read was not reported as an error at all! My formatted document had a sector of misplaced text. After much searching, I finally identified the incorrect sector as coming from sector 0 on a particular track, while the desired sector had been sector nine (which immediately precedes sector zero on the same track).

It seems that the Poly ROM code obtains the Preamble sector data directly from the hardware latch for both writing and reading. For some unknown reason, the software was missing a sector pulse, and skipped to the next sector. Since the sector data was read from the hardware, it matched the data written on the disk, but that was a different sector from the data wanted. To correct this, I modified both the read and write code sequences to make sure that the hardware sector matches the software sector selected for write and read. An error on this condition is now reported as a Preamble error (0102). Such an error may occur on Write if the hardware sector does not match the selected software sector when it gets down to the actual sector read or write code. By making this change, I eliminated a possible source of error in processing disk sector interrupts. I have subsequently decided that this skipping of the sector interrupt may have been caused by a some other hardware problem on the CPU card, but I kept the code changes anyway.

I had two SD disk controllers; one was revision C and one was revision E. My revision C version did NOT work, but the revision E version did work.

In my original testing, I was able to write to a disk in a 460 drive, but was unable to read it. Checking that same

The SA-460 comes up to speed five times faster.

disk with a SA-400 drive, I was able to determine that the SA-460 drive had written perfectly. The problem was in reading the drive.

Len Araki at Poly, tells me the reason my revision C controller does not work has to do with the data separator on the two boards and the higher track density. He says the higher track density is an electrically noisier environment. The reason the revision E controller works while the revision C does not, is that there was a design change which gave the revision E card a better data separator circuit. It works in the noiser environment. Subsequently, I swapped my revision C controller for another revision E controller; it also worked.

Of course, in writing the 96ROMs, I kept all the enhancements and corrections I had made and incorporated into the ASROM PROMs.

To accomplish the change, I first wrote a program (Dio96.GO) to hook into Dio and run a SA-460 on drive 3. This program requires Poly PROMS version 81, or ASROM PROMS (which are compatible with Poly PROMS version 81) because it uses the WH9 Dio connection. I installed one SA-460 in drive 3 and then, with standard system software, initialized a diskette at 96 tpi and copied the operating system onto it. Next I replaced the ROMs with my new 96ROMs, and put in the other two SA-460s. Then I booted on the 96 tpi system diskette, and was off and running. Finally, I wrote a program to hook into Dio (Dio35.GO) and run the SA-460 in drive 3 in 35 track, 48 tpi mode. With this software installed, I can read (and write) the old standard format disks on drive 3.

I have been using standard (cheap) SSSD disketts, and have had only 5 out of 36 fail to initialize, or show up with a media error. In those cases, I was able to initialize them to standard 35 track 48 tpi format. Of those five, three were *very* used, ancient diskettes. I have been able

to read and write to those disks using the Dio35.GO program.

The double sided feature of 96ROMs matches Poly's double sided feature for SA-450s, and requires a modification to the disk controller card. 96ROMs would support SA-410s (SS 96tpi—800 sectors), SA-465s (Half height DS 96tpi—1600 sectors), and SA-415 alf height SS 96tpi—800 sectors) If other drives have the Motor On from Drive Select and Side Select from Direction Select features and have the same or smaller step, settle, and motor start times, then they should work without modification to 96roms.

The SA-460s are longer than the SA-400, and I had to file a small amount of the plastic (1/8 inch) from the power plug to make the drives fit flush. I also had to run the power cables through one of the upper air flow holes because the drive 1 power plug blocks the bottom hole through which the cables normally run.

I have been using the system for four months now, and everything seems to be running fine.

For more information contact Ralph at the address listed at the beginning of this article. PL

LETTERS

PL often gets very thoughtful letters from you, the readers. *PL* will periodically devote space to these letters, sharing with other Poly readers various items of interest.

Ron Moffatt, of Rochester, New York, tells of his recent memory upgrade from four older memory boards to one 64KB board. Ron writes: "My older boards were Vector Graphic 8K static. The board I installed was a Digital Research Computers (of Texas), 64K static, which I modified to work with Exec.

"Before I removed the old memory boards, I connected Poly to a watt meter and measured a 190 watts of power consumption (disks off). I then repeated the test with the new memory board installed and measured only 130 watts. As you can see, this represents a sizeable reduction in power consumption. More importantly, with less power being consumed and dissipated as heat, the internal temperatures will be lower. Also, because the overall board count is reduced by three, there is better air circulation around the cards. This should improve the overall reliability of the system."

PL did some calculations based on that 60 watt gain. If you run your system 10 hours per day, five days a week, that's three kilowatt hours a week, or 156 kilowatt hours per year. Assuming a national average electric rate of eight cents a kilowatt hour, that's almost $13 year, nearly enough to cover your subscription to PolyLetter. In Ron's area the kilowatt hour rate may be double, and the savings would of course match.

$13 dollars (or even $26) probably isn't that much to get excited about, but as Ron mentioned, you should consider the reduced heat dissipation (remember, that extra 60 watts is shed largely in *heat*). Less heat means longer component life throughout the system. Result? Better reliability.

Should you replace multiple memory boards with a single board immediately? Probably not, but if you have, say, a 32 KB machine and want to expand it to a full 56 KB, get the single board memory and save the old ones for spares. PL

continued from page 1

black-and-white video signal. The VTI bridges the gap between these two points of view.

To understand how the VTI, or any other character display system works, first look at how the display is organized. The VTI puts out 16 rows, each containing 64 characters. Furthermore, each row is divided up into 15 raster lines. (If you look closely at your video monitor, you should see horizontal stripes in every character. Most monitors blur these together somewhat, but they get clearer if you turn down the brightness and contrast.) So the total number of raster lines in the picture is 15 times 16, or 240.

Now consider how the VTI displays the top line of characters on your screen. These characters are stored in memory locations 0 to 63, counting 0 as the first RAM location of the VTI board. The VTI must scan the top row of dots from characters 0 to 63, to produce the top raster line of the screen. Then, it must go back and scan the same 64 characters again, only this time displaying the second row of dots from each character. This process continues until the 15th row of dots has been displayed from each character. Only then can we move on to the second row of characters, stored in locations 64 to 127.

If we were to write a "program" to simulate how the VTI generates a display, it would look like this:

```
10 FOR ROW = 0 TO 15
20   FOR CHAR = 0 TO 63
30     FOR RLINE = 0 TO 14
40       C = PEEK(ROW*64 + CHAR)
50       D = DOTS(C, RLINE)
60       DISPLAY D
70     NEXT RLINE
80   NEXT CHAR
90 NEXT ROW
```

ROW is the row number, 0 to 15, of the display. CHAR is the character number on this row, 0 to 63. RLINE is the raster line number, 0 to 14. In line 40, C is loaded with the memory data from address ROW*64+CHAR. This is the character we want to display next. In line 50, D is loaded with a pattern of dots corresponding to the character, C, and the raster line, RLINE. For example, a capital "A" might look like

```
..XX.....
.X..X....
X....X...
XXXXXX...
X....X...
X....X...
X....X...
```

If so, the first raster line should be displayed as two "off" dots, two "on" dots, and five more "off" dots. This data is loaded into D and displayed. These patterns are actually stored in a ROM on the VTI card.

Well, it should be obvious that we can't really write a program to perform a video display. But there is logic on the VTI which does each of these steps, and there are even storage locations called "latches" which correspond to the variables we've defined above. A program could never do this in real life, because it would be much too slow. The time allowed to display one raster line is about 52 microseconds, less than one microsecond per character. The Poly's 8080 CPU can't execute a single in-

struction in less than 2 microseconds!

In addition to the dots which make up the character display, the VTI has to create sync signals which are mixed with the video. These are horizontal and vertical sync pulses, required by all TV sets and video monitors. A horizontal sync pulse is required at the end of each raster line, and a vertical sync pulse occurs at the end of each "frame" (one complete sweep of the screen). These tell the video monitor where the lines and frames begin.

There are two controls on the top of the VTI card. One adjusts the width of the display, and the other alters its position with respect to the left edge of the screen. They do this by altering the speed of certain functions on the card, like the time between dots, and the width of the sync pulses. By adjusting these controls with a small screwdriver, you may be able to center the display on your monitor, or make it fill out to the edge of the screen.

Why 16 lines of 64 characters? True, that's a design decision we Poly owners often regret now, as our CP/M friends enjoy their 24x80 displays. There are some solid reasons for 16x64, though. For one thing, 16 times 64 is 1024, exactly one 'K' of memory. And since 16 and 64 are both powers of two, the VTI hardware design became much simpler. And finally, Poly wasn't the only one to use this format. Processor Technology's SOL computer was available at about the same time as the Poly-88, and also used a memory-mapped, 16x64 display.

The VTI card in the Poly-88 was originally addressed at F800 hex. As the system grew in complexity, the designers decided to move it down to address 1800, leaving everything from 2000 to FFFF for program memory. Switches on the VTI card select its address, and it can actually be located in any 1K block of memory. But, moving it somewhere other than 1800 in the current Poly system would be a bad idea.

But regardless of its address, the VTI is a "memory-mapped" display. A program can read and write the screen memory just like any other RAM. This makes the VTI extremely fast. Compare the time it takes to write a character to memory (a few microseconds) with the time it takes to send that same character over a serial port (1 millisecond, or 1000 microseconds). For screen-intensive programs like editors, you just can't beat the VTI.

Troubleshooting

The VTI is one of those essential cards, like the CPU. Without a working video display, it's difficult to get any information at all about what's wrong with the system. For this reason, a well-equipped computer room should keep a spare VTI on hand, or be able to borrow one from another nearby Poly system.

Before chasing VTI troubles, though, check the other possibilities. Is your video monitor working? Plug your videocassette recorder's "video out" into your video monitor and see if you get a picture. Is the CPU working? Watch the disk drives and see if they respond normally when you type a command, even if you can't see anything on the screen while you type.

If you have a voltmeter, and you feel comfortable using it, check the voltage regulator. It's the big round fellow in the corner of the VTI card. You should measure about eight volts from one of its center pins to ground, and almost exactly 5.0 volts from the other center pin to ground. Be careful not to short out the regulator while checking it! You could destroy the VTI card and the Poly's power supply too.

If a single character on the screen is defective, the problem has to be in one of the eight RAM chips. These are marked 9111 or 91L11, 2111, or something similar. If you have a spare chip, a defective one can be located by the standard "swap and check" procedure: put your spare in place of one of the chips, and try the card again. If the same problem exists, the chip you removed is probably OK, so use that one to replace the next RAM, and so on.

If the entire screen is affected, the problem still could be a RAM chip, but it's likely to be elsewhere. Without a set of schematics and a good knowledge of electronics, your best bet is to swap chips, one at a time, from a working VTI card. Since the VTI was designed almost 10 years ago, many of its components are becoming harder to find. A spare VTI card is an excellent investment. PL

Software Boots

No, we're not talking about the latest chic footwear from Dallas or Paris. We're talking about the method of booting your Poly *without* pressing the load button. Many of you are familiar with the sequence:

```
$ENABLE
$$b x
```

where "x" is the drive from which to boot. (If no drive is specified, the current system drive is rebooted.) This is called a "software boot". (A "hardware boot" entails the usual sequence of either powering up or pressing the load button).

Software boots are useful if you wish to boot a system disk that is in a drive other than drive 1. For those with hard disks, you can boot from drive 1 and load all the necessary HD driver software, then boot to drive 4, or however you have configured your system. You've probably placed system files in each volume and can boot any volume.

There are, however, some cautions. First, the software boot doesn't "do" everything. Hardware is not reset, several system flags are left uncleared, and other parts of memory are also left uncleared. If you're rebooting to clear an anomaly in the system, the software boot is *not* the way to do it. Press the load button instead. (User memory from 3200 on out to memtop is left intact.)

You must be extremely careful about booting from one version of Exec to another. For example, you can boot from Exec 83 to 96, but going back, from 96 to 83, can be disastrous. The system can crash, requiring a hardware boot. Disk directories can be destroyed quite handily. The older Execs do not know about some of the flags in the newer Execs, thus causing problems. To be on the safe side, especially MS or HD owners, you should standardize your system disks to one Exec, preferably Exec 96.

Emergency System Booting

The one drawback with the software booting procedure outlined above is that the system must have booted "naturally" (load button or power on) at least once. But assume for the moment your default drive (drive 1) is faulty and the system will not boot. Things need not come to a screeching halt. Assuming the problem is in the drive and not in the controller or somewhere else in the system, it is possible to boot initially from another drive. Remove the disk from drive 1 and leave the drive open.

Press the Load button to make sure all hardware is reset. You should see "ERROR 306" on screen ("I can't read the directory—No disk or door open"). If the drive appears to be causing erratic behavior with the system, shut off the system and disconnect drive 1's power supply and data cable. Turn the system back on. You should then get "ERROR 306".

Place a system disk in drive 2. Press Control-Z; you'll enter the familiar front-panel mode. Type the following:

L2D92 (carriage return)
2 (carriage return or space.)
SPJ403 (carriage return)
G (no return required)

At this point, the system boots from drive 2. If you wish to boot from drive 3, enter a "3" in place of the "2" in the second step noted above.

Essentially, this is the entire boot process with a pause to change the default drive (stored at location 2D92H). The "pause" is instigated by the error condition of "no disk in drive, door open." Up to this point, the hardware portion of the boot, performed by the power-up cycle or pressing the load button, has been done. But Exec still needs to come up. Which disk is to be used is what we have changed here. Once that's done, we continue the boot from the warmstart point, 403H.

NOTE: Just to be on the safe side, you should *always* press the load button or power up the system before attempting this kind of boot. If you make a mistake typing and the boot process doesn't start, press the load button before going through the front panel sequence again.

Twin-System and CP/M *users should contact the factory for instructions when booting from front panel.*

NEWS continued
dably, the abandonment of Pascal is due to Poly manpower being directed toward the new system.

Pascal Pastures will return when Poly Pascal is once again alive. Alternately, *C* on Poly is still being investigated. More on that as things progress.

PolyLetter subscribers are disappearing at an alarming rate. A few have sold their systems, many more simply aren't using them anymore. If you're thinking about moving on, please drop your friends at *PL* a line. And double-please, if you sell the system, let the buyer know about *PL.*

POLYLETTER, September/October, 1984
Editor and Publisher: Frank Stearns, PolyServe
Contributing Editors: Bob Bybee, Ralph Kenyon
Subscriptions: (Domestic) $15.00 per year; (Canada) $18.00 per year, payable in US Dollars; (Overseas) $20.00 per year, payable in US Dollars.
Editorial Contributions: Your contributions to this newsletter are always welcome. Articles, suggestions for articles, or questions you'd like answered are readily accepted. This is *your* newsletter, please help to support it.
Advertising (camera-ready or PL typesetting): $2.00/column inch; full column: $18; full page $30 (Personal ads are still free of charge.)
PolyLetter is published bi-monthly by PolyServe. PolyLetter and PolyServe are not affiliated with PolyMorphic Systems.

POLY ADS

FOR SALE: Here's that second system you've been waiting for: 8810, 48K, SSSD, monitor and keyboard (with keypad).

Wayne Norris
215 Palisades Drive
Santa Barbara, CA 93109-1943
805/962-7703

FOR SALE:

Two 8813 twin systems, with keyboards and monitors. One system is equipped with a Diablo IPSI printer. All reasonable offers considered.

Larry Issacson
I and F Ventures
P.O. Box 93887
Atlanta, GA 30377
404/351-3652

PLEASANT MEMORIES

Does your Poly suffer from memory loss? Is your top of RAM less than you'd like it to be? Are you concerned because your chassis is full of old, hot, RAM cards, with hard-to-find chips?

Poly Peripherals introduces a new 56K static RAM card. This low-power card supplies all the memory a Poly needs, putting the top of RAM at FFFF. Increase your Poly's health for more years of reliable service, keep your system running cooler - and at the same time, open up some vacant card slots for future expansion!

The Poly Peripherals RAM card is compatible with any standard 8810 or 8813, except CP/M or TwinSystems. Introductory price is $305; call for current pricing or further information.

Poly Peripherals
1437 Sugarwood Lane,
Norcross GA 30093
404/925-2480
Source Email: TCD125

For sale: 3 drive 8813, Sanyo VM 4509 monitor, printer interface, Serial Epsom MX-80, Vector G 25IO (kit and manual). Full documentation (including Programmer's manual). Software includes PLAN; custom software includes Epsom Driver (BASIC). All fully functional, pretty, and aligned. $2320 complete, $1760 less printer and monitor.

Bob Prater
1200 Green Forest
Austin, TX 78745
512/445-5666

For Sale: 8813, 3 drives, 56KB memory, Amdec Video 100 monitor, standard keyboard. Radio Shack DMT-100 printer, plus manuals and some disks. $1000

Tom Rau
2668 Camille Way
Lawrenceville, GA 30245
404/998-8095 8-5 pm
404/921-1462 evenings

SELECTIVE ARISE

Many of you probably have the ARISE.GO utility program, which allows the "undeleting" of files. Its default invocation is the following:

```
$ARISE <pathname>filename
```

What may have never been documented anywhere is the following invocation,

```
$ARISE <pathname>filename x
```

where "x" is the sequential number of the target filename found in the directory. For example, if on drive 2 you have seven deleted files by the name of "TEXT.TX" and want to revive the fourth one, the following command would be used:

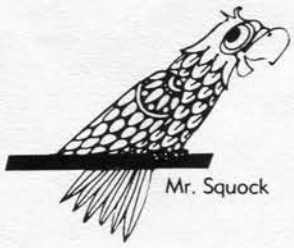
```
$ARISE <2>TEXT 4
```

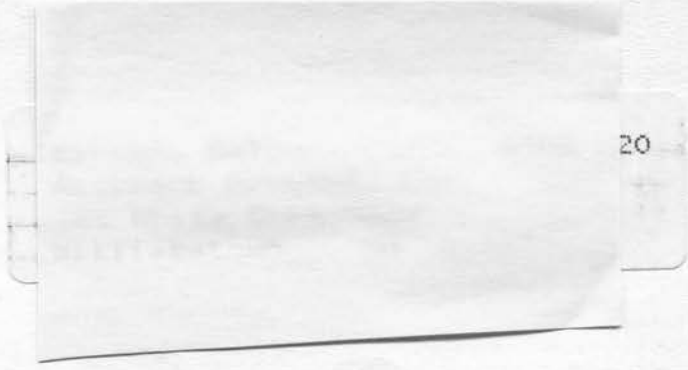
The fourth version of TEXT is undeleted.

NEW POLYLETTER DISK OF THE MONTH: Er-ah, disk of the "*sixth* month". After some delays and other minor difficulties, a new PolyLetter Disk of the Month has just been released. This DOM includes the following: Bob Bybee's SETPR, which allows you to send commands to your printer; Bob's DDBLIST, which allows comprehensive listing of what volumes/devices are connected to your Poly (for hard disk users); a program by PolyLetter similar to Bob's SETPR, but one designed exclusively for the Gemini 10/15 printers; PolyLetter's character and word counter which also checks for parentheses and quotation imbalances; PolyLetter's file merger program (MERGE), which will take smaller text files and place them in one larger file; and PolyLetter's Super Futil, a modified version of Futil (File Utility program). New Futil features include single keystroke action where possible, selective directory reading, and new main menu functions such as SAVE NAMES, and SET NEW (set new bit). Send $15 plus $1 shipping and handling to PolyLetter, 14307 NE 16th St, Vancouver, WA 98684.

PolyLetter

14307 NE 16th St
Vancouver, WA 98684
206/892-3970





FIRST CLASS MAIL

PolyLetter

The Newsletter for PolyMorphic Systems Owners and Users

November/December #8406

NEWS

SPACE SHUTTLE RESCUES TWO SATELLITES. What does this have to do with your Poly? Nothing really, other than old equipment or abused equipment, when well built, can be worth an awful lot of trouble to maintain. Your 8810s and 8813s probably fall into the same category, though we don't know that NASA would send a shuttle to fetch an 8813 out of orbit (perhaps if it had a hard disk attached...).

OLD POLYS WANTED "EN MASSE". We refer you to the ad later in this issue placed by Mark A. Cole. Let PL know if you decide to sell.

NO NEWS FROM POLYMORPHIC: That's right, stony silence. PL contacted PolyMorphic regarding news for this issue and was told there was "nothing new". We didn't even get a status report on the new system. We are hoping to report on the new system by the time the next PL rolls around. Stay tuned.

WE'RE RUNNING LATE. PL is running about 10 days behind its normal press schedule. We won't go into the boring details as to why, just that we are aware of this and hope to have it corrected within the next few issues. Please stay with us.

IDEAS FROM RUSS NOBBS

Russ Nobbs, of Spokane, Washington, long-time friend and booster of PL, dropped us a line with some questions and propositions. PL is interested in your reactions or solutions to the items below. Review this list, then send us your notes. We'll try to publish as much as we can in the next issue. Russ said it best: "Sharing info on what is (still) out there for Poly might help keep PL readers using their Poly..."

1. How about Ron Moffat telling us all how he modified the DR(TX) 64K board to run on Poly?

2. How about PL telling us how they get their Poly to talk with almost any other computer over the RS232 ports at speeds as high as 9600 Baud? *(That's coming in the Typesetting series of articles starting with this issue. But other readers who've successfully gotten their Polys to talk to many other computers are welcome to relate their experiences too.—Ed.)*

3. How about asking active users what commercial software works for them?

4. I've incorporated Kenyon's Hsrt.OV into my BASIC

GOTO 6

TYPESETTING WITH YOUR POLY

Many users have asked PL, "do you do PL's typesetting from your Poly? Really? How did you do it? How much does it cost?"

Because it appears typesetting is growing in popularity for general business applications, and because several new (and old) typesetting houses are specializing in keeping typesetting costs down and turn-around time fast, many small-businesses and consultants are using typesetting for their various projects, but on a "do it yourself" basis.

Three to four years ago, typesetting from your home computer was unheard of, and in fact was actively discouraged by the manufacturers of typesetting equipment. The home computer was seen as a threat to front-end typesetting hardware sales. With the avalanche of the home computer, however, typesetting manufacturers have "seen the light" and now fully support personal computer tie-ins. ("Front-end" generally refers to the hardware and software used to input and process the data entered. The output section is the hardware that actually generates the type.)

In this and future articles, PL will recount some of the highlights of our typesetting "career", and some of the ins and outs of bringing a publication such as PolyLetter to typesetting.

Why Typeset in the First Place?

Typesetting offers a visual impact unmatched by typewriters or most letter-quality printers. It has been proven that readers retain typeset information more than they retain the same information presented in non-typeset forms. From another standpoint, quality typesetting offers such a superb original photographic image that when the inevitable visual losses associated with the printing process occur, the affect of these losses on the final printed sheet is minimal. (Good typesetting is so visually crisp, in fact, that it can be almost painful to read the film originals—the transitions from white to black are bitingly razor sharp.) Typesetting gives a "professional" look to your copy. In addition, you can add subtle shadings to the meaning and "feel" of your copy through the type style chosen.

History

Back in the old, *old* days, type was assembled in a mechanical form *by hand* from existing metal pieces. This assembly was loaded into the press and the paper came in direct contact with the type. Later, the linotype machine simplified the assembly process—the linotype operator keyboarded in the copy. The keyboard was con-

GOTO 3

THE CENTRAL PROCESSING UNIT (CPU)

by Bob Bybee, Poly Peripherals
1437 Sugarwood Lane
Norcross, GA 30093

The Poly Central Processing Unit (CPU) is the "brains" of the entire machine. It directs the operation of all other cards in the system: it tells the disk controller what to read and write, it puts things on the video display, and it even tells you when there's no disk in your drive.

It's easy to become awed by a machine that appears "smart." Just remember that like any machine, the CPU only does what it's told to do. Inside the guts of the machine, the CPU is mindlessly interpreting voltages—little "ones" and "zeros" that float around on the copper traces of the card. Fast, yes, but also quite stupid. My cat has a higher IQ.

The Poly CPU card uses the grand-daddy of all microprocessors, the 8080. This processor chip, designed by Intel as a controller for Datapoint's cash registers, became the first general-purpose computing "thing" that could be used in low-cost systems. There were CPU chips before the 8080—the 8008, 4040, and 4004 are some numbers from the Stone Age—but the 8080 really started us into modern microcomputing. The 8080 chip, by itself, cost about $400 when it was first made available to the hobby market.

The Pieces

In addition to the processor itself, the CPU card contains 3K of ROM (each of the 3 ROM sockets holds a 1 K-byte ROM or PROM chip), 512 bytes of RAM, a serial port (the 8251 IC), a real-time clock, single-step control, and interrupt logic. Plus, any CPU card will have some small "glue" chips, so called because they hold the system together—a CPU does not live by large chips alone! The "glue chips" along the card-edge connector are interface chips which send signals on and off the S-100 bus.

The ROMs

The first ROM is addressed from 0000 to 03FF hex. An 8080 processor always begins executing at address 0000 when it sees a reset signal, caused by turning on the power or pressing the LOAD button. In older Polys, the first ROM was called the "4.0 Monitor." It was a small program that operated the keyboard and video display, could run the Front Panel, and could read and write programs on cassette tape. In the old Poly-88, this was the only ROM.

One feature of the 4.0 Monitor was that it would look at address 0400 (the start of the second ROM). If nothing was installed in that socket, the 4.0 Monitor would expect you to load a cassette program or enter the Front Panel... there wasn't much else it could do. But if it saw a program at 0400, the 4.0 Monitor would jump into that program. This was a very flexible way to expand the system: simply plug in some new ROMs, and you don't even have to change the first ROM in order to do it!

So the early versions of the disk-based Poly simply added two new ROMs to the CPU. These new ROMs contained the disk I/O routines, and the programs which looked up files in disk directories.

Only recently has the first ROM been changed. Enough enhancements were added to the ROMs that Poly's programmers just about ran out of space. Something had to go, and so in ROMs version 81, the cassette tape I/O routines were removed. Now, instead of an independent 4.0 Monitor and two additional ROMs, new CPUs are shipped with a set of three cooperating ROMs. Cassette tape is a thing of the past.

RAM

The bulk of the RAM in a Poly system will be on cards other than the CPU. Usually your system will contain one 48K card, several 16K cards, or some other combination.

In the days of the first Poly-88 systems, RAM was expensive, and one or two 8K RAM cards was all that most people used. So that the CPU could perform at least some functions without other RAM cards in ths system, Poly's designers included 512 bytes of RAM on the CPU card itself. This RAM is addressed from 0C00 to 0DFF, and again from 0E00 to 0FFF. (The contents of address 0C00 are the same as 0E00, and so on.) In current Poly software, this RAM is used for the stack, wormholes, and miscellaneous data areas. There isn't enough RAM here to run any sizeable programs.

Interrupts

The CPU wants to be able to do more than one thing at a time—for example, the system might be printing a file while you type in the next command. The CPU must be able to see that you are typing in a new command, and store it away for future reference. Events such as pressing a key on the keyboard are "unexpected" events, things the processor must deal with at unpredictable times. To handle these events, an interrupt signal is generated. The CPU handles the event by stopping its current program, processing the source of the interrupt (such as a keypress), and returning to its previous task.

The CPU can respond to interrupts from seven different sources. In the Poly, these sources are the disk controller, keyboard, serial port, real-time clock, and the hardware "single step" circuit; some of these are described below. Two interrupt inputs are unused in the Poly. You may have installed some additional cards in your machine which use one, or both, of these interrupts.

Serial Port

The CPU card has two connectors in the upper right corner, one of which is used for the printer interface. The other was originally used as a cassette tape interface, but in many machines it has been modified for use as a second printer port. The serial port uses a USART chip, which converts CPU data (8 bits at a time) into a stream of data which is fed, one bit at a time, to the printer. It also has the ability to receive data in the same fashion. The USART has some control inputs which the printer can use to indicate when it's not ready for new data—these are the "hardware handshaking" lines.

The USART is a large chip, labeled 8251 or something similar. Near it is a smaller chip, the Baud Rate Generator. This chip is labeled MM5307. It creates the timing signals which tell the USART how fast to send data to the printer. The MM5307 is becoming extinct, and it would be a good one to have in your spare parts kit.

Real-Time Clock

The Poly CPU receives interrupts 60 times a second from the 60 cycle power line frequency. This allows it to have a real-time clock function. It's not a perfect clock... when the CPU has interrupts disabled (during some disk I/O, for instance), clock ticks can be missed, causing the clock to lose time. But it's useful for a lot of purposes. For

example, the real-time clock is used by the disk controller to determine when to turn off the disk drives.

Single Step

When debugging a program, it's handy to be able to execute one instruction at a time (such as "WALK" in BASIC). The 8080 CPU doesn't have a way to do this, so some early microcomputers built this function into their front panel. Remember the Altair and IMSAI machines with all the lights and switches? One of the switches would let you trace a machine code program, one step at a time. But then you'd have to interpret the results indicated by the lights, a painful task at best.

Poly eliminated the lights and switches with a "software front panel." You've seen it often, either by typing Ctrl-Z when in Enabled mode, or when your system hiccups. This front panel has many advantages over its hardware ancestor, because it can display more information, and is much easier to interpret. But the single-step feature of a hardware front panel would have been lost if not for Poly's ingenuity. A small circuit on the CPU card lets the Poly execute a single instruction, then generates an interrupt. The interrupt forces Poly to re-enter the front panel, simulating a single step instruction.

Troubleshooting

Most of the chips on the CPU card are still available from electronics distributors. Many machines use the 8080 processor and its family of components, such as the 8251 and 8224. For at least the next few years, you should be able to find these parts easily.

But remember how critical the CPU is to your system. With a faulty CPU, you can't get any output on the screen, so it's often impossible to tell whether the problem is in the CPU or elsewhere. Power supply problems can cause a CPU to seem faulty. In some cases, a bad card elsewhere in the system can prevent the CPU from operating, because the whole system is tied together over the S-100 bus. For this reason, it is a very good idea to have a spare CPU.

If you suspect a CPU problem, try this procedure: Turn off the power and open up the Poly cabinet. Remove all boards except the CPU, the video board, and one memory card (the one addressed at 2000 hex). Turn on the power and you should get "Error 0306" on the display. If you have a flickering Front Panel display, check that you have a memory card installed and set to address 2000. Or, reconfigure another memory card to that address and try with that memory card instead.

If no combination of memory cards will give you the "Error 0306" message, suspect the CPU. If you can get the message with one memory card, the fault is probably not with the CPU. Add cards, one at a time, and re-test the system after each card is added. If adding one card prevents the system from printing "Error 0306", that card is probably at fault.

This is a good time to make an important point: Whenever making changes in your system, *take careful notes* of the switch settings, card locations, and anything else you change. Nothing is worse than not being able to put your machine back together after performing "repairs." Also, remember to turn off the system when removing or insert cards. PL

continued from page 1

nected to an elaborate mechanism that actually formed liquid lead into the characters typed by the operator. Mechanical assembly speed was increased because the copy came out of the machine in the order typed. While an improvement, the process was still unwieldly. Then photographic printing plates came along, giving the printer the option of photographically transferring virtually any image onto a printing plate. Plates, with their indelible image, now made the contact with the paper. This meant that type could also be handled photographically, and also generated photographically. The first generation of contemporary typesetters were born.

Characters are formed by shining a light through a stencil-like disk or card. The resulting image is projected onto special light-sensitive paper or film. The distance from the disk or card to the paper determines the character size; while the choice of type disk or card determines the resulting type style. When developed, the characters appear on photographic paper, not at all unlike the photos you pickup from your favorite camera shop. This paper, when assembled with other copy and art, is then photographed onto the plates that are used in the printing press.

The second generation of typesetters have the various type styles stored in digital form on floppy disk instead of on stencils or cards. The characters are reproduced electromechanically from the digital information directly onto the photosensitive paper. In terms of production flexibility, the digital typesetter is as far ahead of the projection typesetter as was that typesetter from the linotype. However, projection systems are still in common use today. As far as the front-ends go—determining spacing anjustification—not much is different between the two machines.

In this series of articles we're now most concerned with two things: (1) making the bridge from word-processing commands to typesetting commands and the "philosophy" of typesetting commands in general; and (2) getting the text out of your Poly and to the front-end of the typesetting system.

Command Structures—Overview

Typesetting software within the typesetting machine can do a great deal more than can your Poly or most other word-processors. Therefore, the format coding is more intricate and sometimes much more cryptic. There are several areas where there is no way to translate Poly commands into typesetting commands, directly or indirectly. These non-translateable commands include pagination with multiple columns; "merging copy" (picture a multi-line word undent); multiple tab fields with individual justification controls for each field; fully proportional spacing, full control (down to 1/72 of an inch) of vertical spacing; character style and size; and so on.

But Poly owners are lucky in one major respect: the Poly wps (word processing system) uses ASCII alpha characters accessible by the editor to determine formatting events. Many other systems do not. Wordstar, for example, uses the high (eighth) bit to indicate certain things to its formatter, and other "invisible" embedded control characters indicate other items. In the sometimes perhaps misguided design effort to keep the user at arm's length from what is happening inside the system, the Wordstar user and users of like systems have

no direct manipulative control over the format commands. We Poly users do. (There are other pluses and minus to both approaches, which are not relevant here.) This means that a large number of Poly commands are in fact directly convertable into typesetter commands. Moreover, because many typesetting editing programs do in fact use a "visible" embedded command structure rather than an "invisible" one, it is easier for us to deal with the typesetter's embedded command structure.

Code Conversions

There are three ways to do this. (1) you do all the conversions from Poly wps commands to typesetter commands before the file ever leaves your Poly. The advantage is that you have ultimate quality control, and you also save money. The disadvantage is that you have no one to blame but yourself if things don't work right. On the other hand, if the documentation provided by the typesetting house is poor, you may have some recourse, but probably not much. If you take this course, we urge you to do a small, non-killer deadline project first. Save the big, critical projects for later, after you've had more experience. Of course, hang on to your original Poly wps encoded files. Make working copies for typesetting applications.

(2) send the typesetting house straight text files that have no Poly wps commands in them. Tell them they will be responsible for coding the files to get the output you want. If possible, provide a Poly wps hardcopy version of the document so that the typesetter people can get some idea of what you want. Barring that or in addition to that, provide detailed instructions—typesetting people tend to be rather literal. They probably don't envision the final product the way you do, and this is especially true with first-timers!

(3) meet the typesetters halfway. As you might guess, there are several shades of gray here. You can: explain to the typesetter how Poly wps commands work and then let them do the conversions (make sure your explanations are written!); do some code conversion yourself, leaving the balance for the typesetter; or finally, agree on some intermediate command structure. This last one sounds strange, but it did happen to us once. Due to character set and modem problems, we had a severe character set bottleneck, and had to make do with almost no command delimiters such as braces, curly braces, backslashes, and so on. Intermediary character sequences were chosen. Again, carefully tell the typesetters what you want!

Character Sets

The Poly is a neat machine in that all the standard ASCII characters are producible, and the system will handle them without hiccuping. (Earlier Apples are missing quite a few characters; and it is possible to "blow up" a Compaq or IBM-pc by sending certain characters though the system.) The problem with such a full character set can come with "smart" typesetting input software that tries to compensate for the failings of other machines. You may find it necessary to limit the spectrum of Poly output so that it doesn't trip some of the traps set by smart typesetting software. This depends entirely on the typesetting system—you probably won't find out about these until you try a job.

Unfortunately, at this point the average typesetter can't help you—assuming you have problems. But a reputable typesetter will get a hold of the field service representative for his system. If that rep hears the name "PolyMorphic" he will most likely go "huh?"—what you need to do is convince him or her that a "PolyMorphic" is a commercial-grade desk-top microcomputer, far ahead of its time and still ahead of its time in many respects, and that the system has no odd caveats when it comes to the character stream.

File Transfer

Obviously, handing a typesetter a standard Poly format disk is out of the question. After having worked with many typesetters and having called scores more, PL can make the sweeping statement that there probably isn't a typesetter in the whole world that could take your Poly disk directly, unless an 8813/10 was the front end of the typesetting system (incredible if so), or if a Poly was one of the micros parked in the "input room" (unlikely).

This leaves getting your Poly disk transferred to a "standard" disk format, or telecommunicating your files via direct wire link or modem. There really is no preference; there are many considerations. If you're using wire or modem, how good is the modem/wire software on each end? Is it long distance? What kind of error checking is used? How long will the transfer take at 300 or 1200 baud? If you are using a disk format, how will you transfer your Poly files to another disk format? Will the typesetter be able to read that format? Do you have time to mail the disk and wait for the return?

The PL process

PL is composed on a 8813 using standard Poly wps commands. A proof hardcopy is printed using the formatter. When everything looks ok, a command file, designed to automatically edit PL files, converts Poly wps commands to a custom set of Compugraphic typesetting commands. (Compugraphic is a leading manufacturer of typesetting equipment.) The revised files are rewritten back to a Poly disk. Additional manual typesetting coding is also required from time to time. A local Poly owner who also owns a Kaypro allows PL files to be transferred from his Poly to his Kaypro under a CP/M format. The resulting CP/M disk is sent off to the typesetter. In a few days, galleys are returned, and layout and paste-up begin.

While this process may seem a little awkward, it is remarkably fast and very cost-effective.

Comments

There is a whole other dimension to the typesetting process—that of copyfitting. The kind of documents produced by most word-processor users are not concerned with this sometimes hackle-raising part of the project. If there is sufficient interest, PL will dive into this in a future article. While not directly related to the Poly, some custom software written by PL plays a major role in helping during the copy fitting process.

Future articles will also look into the direct link and modem transfer—processes applicable not only to typesetting file transfer but file transfer techniques in general. PL

POLY DISK DRIVE UPGRADE

Charles Steinhauser
903B Allegheny Way
Richardson, TX 75080
214/669-2169

As you read in the last Polyletter Ralph Kenyon upgraded his Poly with 96 tpi drives and now has 1.2 megs of disk storage. I wanted to do the same, but didn't have the "rev E" disk controller card. You will recall that this card was required for the 96 tpi modification. I instead purchased three 40 track drives that read/write 48 tpi. Upon talking with Ralph, he wrote a set ROMs that would tell Dio to use 40 tracks instead of 35 and that we would be using both sides of the disk to read/write a total of 800 sectors per drive.

The drives that were installed were Tandon 100-2. They are DSDD drives, and can be purchased for $160.00 each in either full of half-height versions. The drive is pin-compatible with the SA-400 Shugart drive. They fit in the existing space and the original cables work perfectly.

However, there are two modifications that must be made to the disk controller in addition to the ROM change. These are cutting a trace, and jumping a pin and a jumper made on the top card edge. If you feel comfortable in changing the drives themselves then you will have no problem in modifying the disk controller card.

After all the smoke settled and making sure of the correct changes, the system boots on the original software and runs. All that is left to do is insert a new disk in drive two, initialize it, image the system to it, then use this new disk as the system disk. Now you can write 800 sectors to it. The beauty of this upgrade is that you can read/write all the old disks but you now have 614K of online disk storage. But don't try sending your buddy any software on disks that have been written past 350 sectors—he will be hot if a file continues past 350 sectors and he can't read it!

I have used the 800 sector upgrade now for a month, trying everything I know to make it fail and haven't had any problems. There has only been one disk that failed to INIT, which came from Ralph. I think he used it to level his washing machine!

In my opinion the 800 or 1600 sector upgrade gives Poly yet another lease on life. With Polymorphic dragging its ICs ever so slowly on the new system, one gets very tempted to upgrade to an entirely different system. However, having stuck with the old bird this long, one feels inclined to wait and see what hatches. I hope this Thanksgiving the people at Polymorphic don't have to eat crow.

If I can be of any help, or answer any questions on the 800 sector upgrade, feel free to call.

Advanced User Notes on Changing the Boot Drive

I noticed in PL's article on front panel booting that if you can remove the disk to unplug it (because of a faulty system drive), then all that needs to be done to boot from drive two as if it where drive one is remove the small jumper package from drive one. It is a 16 pin affair and can be located by noting that it is one of two that can be unplugged; all others are soldered. Drive one and two only have one plug-in area used and drive three will have two. The plug-in used in drive three is a termination resistor block. This terminator is always placed in the last drive used.

On the disk's circuit card there is some writing next to the strapping option jumper. If you will notice, the options are something like: HL,DS1,DS2,DS3,MX,MH. All that has to be done is to tell the disk controller which drive will be drive one. By exchanging the jumper from drive one to drive two, the controller selects that drive as system. PL

FOR SALE: 8810 with 64K memory, custom function keys added to keyboard, Amdec Monitor, single and double-side controllers, 1 SSDD and 2 DSDD (external), CP/M, with TEI board for displaying 80x24 under CP/M plus software. Microsoft Basic, Pilot, JRT Pascal, etc...

Dennis Cherry
214/358-3097 after 6:00 pm cst

FOR SALE: 8813 with 64K memory, monitor, keypad added, 2 SSSD drives: $1100. Spare 16K memory board $100.

Don Hollenback
Box 492
Wastucna, WA 99371
509/646-3430

FOR SALE: 8813 with three 5 inch drives plus a single-sided MS with two 8 inch drives; 64K memory; detached keyboard with numeric keypad; Hitachi 9 inch monitor. Make offer.

Brian Biswanger
4235 Dalhart Rd NW
Calgary, Alberta, Canada
T3A 1B6
403/288-7303

WANTED: Information on how Poly MS drives can be connected to the TI Professional. (Contact Brian Biswanger at the address/phone noted above.)

WANTED: Used Poly 8810s and 8813s. Send a description and price to:

Mark A. Cole
125 E. Walnut Street
Richmond, KY 40475

HARD DISK FOR $750

One only—used, 5 megabyte hard disk, with chassis, power supply, controller, host adapter and software, ready to plug into your 8813. This HD has been in use at PolyLetter and Poly Peripherals for over 2 years with no problems. I'm selling this so I can upgrade to a 15 meg disk. Prefer to sell to a "hacker" like myself, since this unit is custom in nature.

Bob Bybee
1437 Sugarwood Lane
Norcross GA 30093
404/925-2480

continued from page 1

programs and use Moe's Form.OV in a few things. Have other people adopted any of the accounting packages that were written on the Poly? Chuck Thompson has heavily modified a GL program so it works for several businesses in his area. I used to use PLAN a lot until I got the OZ (Osborn) with SuperCalc which is much easier and faster. Does Bill Davis use PerfectCalc on his Poly? I still use Keystone Systems' maillist although I have greatly modified it. I use Kenyon's modem operating system for my old DC Hayes S100 modem. Bob's new program runs Hayes smartmodem at 300 or 1200 I understand. *(Bob Bybee's—yes.)*

5. How do you go about getting no-longer-supported but still usable public domain programs? There were a lot of usable sort & merge programs written by people that have undoubtedly gone on to more profitable machines. How can we open that stuff up to wider use? Is it worth trying? *(This is a good one for Mr. Al Levy—Ed.)*

6. How and when does REBUILD.GO rebuild files?

7. How about examples of using RDB for the non assembly language crowd?

8. How about reports from the towns where there are (still) three or four or 10 Poly's running? Are there still any active meetings of local Poly groups? *(Al Levy has a local Poly group through LICA in New York, but are there others?—Ed.)*

POLYLETTER, November/December, 1984

Editor and Publisher: Frank Stearns, PolyServe

Contributing Editors: Bob Bybee, Russ Nobbs, and Charles Steinhauser

Subscriptions: (Domestic) $15.00 per year; (Canada) $18.00 per year, payable in US Dollars; (Overseas) $20.00 per year, payable in US Dollars.

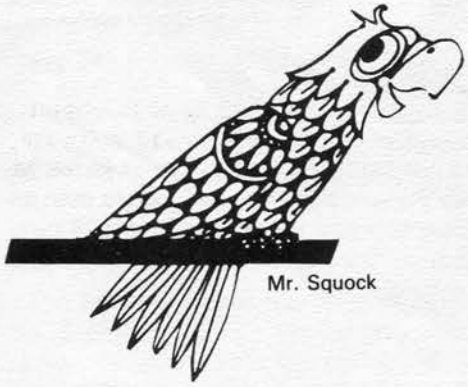
Editorial Contributions: Your contributions to this newsletter are always welcome. Articles, suggestions for articles, or questions you'd like answered are readily accepted. This is *your* newsletter, please help to support it.

Advertising (camera-ready or PL typesetting): $2.00/column inch; full column: $18; full page $30 (Personal ads are still free of charge.)

PolyLetter is published bi-monthly by PolyServe. PolyLetter and PolyServe are not affiliated with PolyMorphic Systems.

PolyLetter

14307 NE 16th St
Vancouver, WA 98684
206/892-3970



Mr. Squock

FIRST CLASS MAIL