

The Z-Letter

Newsletter of the CP/M and Z-System community

Number 12

May 1991

56	47	230	217	99	255	41	237	199	21	256	235	15	151	170	202
107	7	200	128	172	195	147	146	119	91	141	97	127	38	17	198
203	94	214	250	106	110	30	145	100	8	105	134	124	177	165	222
60	68	162	24	136	173	183	156	190	197	210	238	184	61	132	118
140	116	57	133	26	44	229	121	201	178	79	36	73	55	174	231
76	236	4	245	176	28	103	46	54	92	29	23	138	152	1	93
209	2	111	78	6	137	39	108	58	157	142	143	77	216	130	233
65	129	254	63	126	191	135	242	52	102	49	112	115	37	248	13
179	104	98	5	243	123	101	34	48	35	71	208	159	3	212	51
114	40	168	90	192	246	164	87	185	85	252	89	226	80	20	213
158	196	11	153	182	122	180	240	69	19	27	84	224	188	189	194
228	43	144	150	62	227	59	239	22	53	9	219	187	95	75	72
247	207	109	253	82	186	131	113	16	66	204	96	88	148	206	81
83	117	70	18	14	249	31	139	64	244	232	50	149	205	221	218
169	181	193	234	163	171	225	241	33	125	86	167	120	42	32	220
251	10	215	154	223	45	211	166	160	25	155	175	67	74	12	161

Func = S, N = 256: Comps = 0, Swaps = 0, Total = 0
Score (Log N=8 N*Log N=2048): C = 0, S = 0, T = 0

D=Disorder, O=In order, R=Reverse, 1=Ones, L=Last, T=Disp OFF
B=Bubble, I=Insert, S=Shell, W=Wright, Q=Quick, N=New, X=Exit x

Sort algorithms in C, by Joe Wright

A cautionary tale by Bob Vinisky

Saving new directory names, by Jay Sage

Choosing the right language for the job

TABLE OF CONTENTS

ABOUT THIS NEWSLETTER

Submitting material for publication	3
Letter policy	3
Subscriptions	3
How to read your mailing label	3
Advertisements	3
Trademarks	3
Index available	3
Advertisers this issue	34

THE STATE OF THE ART *News of our community*

S. F. Computer Society will continue	4
CP/M users sought for SVCS SIG	4

Sort Algorithms in C

<i>by Joseph W. Wright</i>	7
----------------------------------	---

How to Feel Like a Fool (with almost no trouble at all)

<i>by Robert Vinisky</i>	17
--------------------------------	----

SCRIPT OF THE MONTH CLUB

Saving new directory names <i>by Jay Sage</i>	20
---	----

A PROGRAMMER CORNERED

Choosing the right language for the job	21
---	----

LETTERS	27
---------------	----

PERSONAL ADS	31
--------------------	----

MAGAZINE ARTICLES	32
-------------------------	----

EAGLE COMPUTER USERS GROUP

Meeting place	36
May 11 meeting	36
June 8 meeting	36
ECUG library	36
ECUG software libraries	36
Eagle Hardware Bulletin: Eagle vs IBM keyboards	37

ABOUT THIS NEWSLETTER

Welcome to *The Z-Letter*, a newsletter for the community of CP/M and Z-system users. Everything in this issue is copyright © 1991 by the editor: David A.J. McGlone, Lambda Software Publishing, 720 S. Second Street, San Jose, California 95112-5820, phone (408) 293-5176.

The purpose of this magazine is to spread the news about new developments in the community, and to help newcomers get the most out of their machines. So send us the news about your new software or hardware, your opinion of someone else's product, that article you've been meaning to write, your praise, gripes, or just plain questions! This is the place.

Submitting material for publication

Material may be submitted on 5¼" diskette in almost any format, on 8" diskette, or printed or typewritten on clean white unlined paper. The deadline for submission of material is the end of the month. We cannot pay for articles, but for every article we publish, the author will receive that issue of *The Z-Letter* free. If the author has a subscription, the subscription will be extended for one issue.

Letter policy

The Z-Letter reserves the right to edit letters received to conform to standards of taste, decency, and language. We will NOT distort the meaning of any letter; we'll simply not print it first. If you are not willing to have any letter you send printed, or edited before printing, please say so in the letter. All other letters will be assumed to be for publication and become the property of Lambda Software Publishing upon receipt.

Subscriptions

The Z-Letter is a monthly publication, and subscriptions will be accepted for 1 or 2 years. A subscription starts with the first issue after the subscription payment is received. The cost is \$15 per year for subscriptions mailed to U.S., FPO, or APO addresses. Canadian and Mexican

subscriptions cost \$18 per year. Other foreign subscriptions cost \$45 per year. Subscriptions should be paid by check or international money order in U.S. dollars, mailed to Lambda Software Publishing. Back issues cost \$2 apiece; every back issue is kept in print.

How to read your mailing label

If you are a subscriber, your address label lists when your subscription expires, for example, *Your last issue: 12*. If we have sent you a single issue in hopes that you will subscribe, it will be marked *Sample copy*. *Complimentary* copies go to people we expect to spread the word of the newsletter's existence, and perhaps contribute information or articles.

Advertisements

There is no charge for either business or personal ads. Business ads are carried as a public service to the community. If you sell a product or provide a service to the community, please send us ad copy, either camera-ready or on disk. If you stop doing business in our community, please let us know so that we can drop your ad. Readers who find a product or service unsatisfactory, or discover that a vendor has gone out of business, are requested to inform us.

Subscribers may place personal ads at any time. Each ad will be run three times automatically. If the ad doesn't achieve its end in that time, it can be reinstated after a lapse of one issue. If the ad succeeds before it has run three times, please inform us so we can drop it at once.

Trademarks

All trademarks used herein are acknowledged to be the property of their owners.

Index available

The Z-Letter is indexed annually. The index for issues 1-5 appeared in issue 5. The index for issues 6-8 appears soon! Honest!

THE STATE OF THE ART

S.F. Computer Society will continue

The San Francisco Computer Society met May 6, 1991 at the Laguna Honda School building at 1350 Seventh Avenue, San Francisco. The purpose of the meeting was the annual election of Trustees for the coming year. If there had not been enough candidates for election, the current Trustees had decided to disband the Society, due to the apparent lack of interest in continuing. Ken Thomson, a member of the SFCS, reports that Trustees were elected and the Society will continue.

The speaker for this meeting was John Kerbleski, a lawyer who specializes in Estate Planning, speaking on his specialty.

CP/M users sought for SVCS SIG

I attended my first Silicon Valley Computer Society meeting on April 24. SVCS meets on the fourth Wednesday of each month at Hewlett-Packard Building 48, 19447 Pruneridge Avenue, Cupertino, California, from 7:30 P.M.

Attendance at these meetings is huge. By counting rows of chairs and the number of people per row, I got an estimate of over 200 people present! Business taken care of before the featured presentation included requests for further nominations for the upcoming annual elections, announcements from existing SIGs and ones just forming, and a reminder that SVCS has a resume referral section at (408) 283-1333. Side events included SVCS member registration, a member with a PC-shareware business selling disks, and another member selling proprietary PC software such as GeoWorks and DR DOS.

Resisting the urge to summon up my herald's voice, primarily out of consideration for the ears of the other people standing up front, I took the little clip-on microphone when my turn came and announced that I was looking for people still using CP/M computers to form a SIG for CP/M and the Z-System within SVCS. I invited anyone interested to see me after the

meeting. About six people did so. One doesn't have a CP/M machine, three were only interested in selling me their machines for my collection, and two others wanted information about transferring files from their old machines to PCs. I will keep trying for at least a few more meetings.

The presentation of Borland's Object Vision was a perfect example of the computer adage "Garbage In, Garbage Out". Object Vision is a Windows application which allows one to create screen-oriented interface to a dBase, FoxBase, Excel, etc. data base very quickly. Indeed, the demonstrator created a form to calculate and display the number of vacation days due an employee, depending on his starting date, within about half an hour. However, his haste led him to two mistakes in that time, one minor, and one major.

The minor one was not taking leap years into account when calculating years since date hired. His algorithm was

$$\text{Years_Worked} = \text{ROUND2}((\text{TODAY} - \text{Date_Hired}) / 365)$$

which will often be off by a few days around the anniversary of the date the person started. The correct algorithm is:

```
IF JULIANDATE (Date_Hired) ≤
  JULIANDATE (TODAY)
  THEN Years_Worked =
    YEAR (TODAY) - YEAR (Date_Hired)
  ELSE Years_Worked =
    YEAR (TODAY) - YEAR (Date_Hired) - 1
```

The major mistake had to do with sabbaticals. The decision tree he whipped up, instead of giving an employee a sabbatical every ten years, awards a sabbatical every year from year ten on! I pointed this out at the meeting, but he never corrected it.

The next meeting of the Silicon Valley Computer Society will be June 26. For more information, call Bert Stephens, President, at (408) 286-1271. Membership is \$30 per year.

Lambda Software Publishing

720 South Second Street, San Jose, CA 95112-5820, phone (408) 293-5176

Regular products – quantities not limited

- 1. Spellbinder** Version 5.3H \$60
 The Rolls Royce of word processors. This includes the software, the User's Guide which always came with the software, the Technical Manual and Macro Manual (which always cost extra), and a new Introduction. All four manuals come a 2" binder in 8½ X 11" format. Both generic and Eagle versions are available. (Ltek)
- 2. CP/M** Version 2.2 \$15
 Bought the hardware, and don't have the operating system? I can sell you the license and the manuals, making the copy you get from your user group legal. For many brands of computers, I can even send you the actual operating system with the BIOS for your machine. (DRI)
- 3. MagicIndex** Version 3.00 \$100
 Extends even Spellbinder's control of printers, and what it does to WordStar has to be seen, and then you still won't believe it! Used by us to produce this newsletter. Versions available are: SL (for Spellbinder or other ASCII word processor plus HP LaserJet or laser with HP emulation), SD (for standard word processors, Diablo 630 and similar daisy-wheel printers), WL (WordStar or WordStar clone plus laser), and WD (WordStar and Diablo). Please specify your word processor and printer when ordering. (CES)
- 4. Various computer manuals** Each \$15
 Complete manuals now available for: Eagle CP/M, Eagle 1600, Eagle PC Plus and Spirit, Otrona Attache, and Pied Piper. (Various companies)
- 5. Eagle Computer Users Group newsletter** July 87 to October 90 \$15
 All the issues done by the present editor of the only Eagle user group left. (Lambda)
- 6. The Z-Letter (back issues)** \$3/issue (US, Canada, Mexico), \$5/issue (elsewhere)
 Past issues of our newsletter for the CP/M and Z-System world. (Lambda)
- 7. The Z-Letter (subscription)** \$15/year (US), \$18/year (Canada & Mexico), \$45/year (all other)
 Published monthly. (Lambda)
- 8. Z-Fonts catalog** \$3 (US, Canada, Mexico), \$5 (elsewhere)
 Shows samples of all the fonts available from Lambda for HP LaserJet printers, with instructions on how to choose what size, orientation, etc. you wish. (Lambda)
- 9. Z-Fonts** \$2 per font plus \$2 per disk
 Once you've perused our catalog, you can order the fonts you want very cheaply. (Digi-Fonts)

Opportunity products – limited to quantity in stock

A. CBASIC Reference Manual **\$15**
 CBASIC comes with CP/M as CBAS2.COM, CRUN2.COM, and XREF.COM. However, the manual for the language was rarely included and is hard to find. Get your copy while you can. (DRI)

B. The CP/M Handbook with MP/M **\$15**
 A good introduction to CP/M by Dr. Rodney Zaks. It assumes no prior knowledge of computers. This or another book like it is a must for a novice handed a CP/M computer and told "You're on your own." (SYBEX)

C. Nevada COBOL **\$15**
 This is an excellent COBOL which I've used myself for years. It's the only one I know of that ever sold for less than \$700. Packaged for the Commodore 64, but will run on any CP/M computer. (Ellis)

D. Customizable diskette carriers **\$3**
 These sturdy little carriers hold up to three 5¼" diskettes safe from harm. Ideal to carry diskettes in a briefcase and protect them from your lunch, heavy books, etc. Front and back covers are clear pockets that will hold covers of your design, allowing you to customize these diskettes as you would a binder.

Disk copying – \$10 per disk

I can copy most soft-sector double-density 5¼" CP/M formats, including Northstar and Apple II CP/M. Sorry, no hard-sector formats, no single-density formats, no 3½" or 8" formats, and no Commodore CP/M formats yet; some of this will change in the future. Copies can be CP/M to CP/M, CP/M to MS-DOS, or MS-DOS to CP/M; specify the format of the disks you send, and the format in which you want the copies. Both originals and copies will be returned.

Please note that this is a service provided to the CP/M and Z-System community. I assume that you are honest, and that you are the legal owner of material you ask me to copy. I refuse to accept any responsibility should this turn out not to be the case.

Note also that the price quoted above is for software you purchase elsewhere, or data diskettes you own. **There is no charge for copying software you buy from me in the format of your choice!** Software bought from Lambda will be sent to you in the disk format you request at no cost but the price listed for the software, so don't order Spellbinder and send me extra money for disk copying.

Please circle each item desired, and write the quantity desired in the margin next to the listing for the item. For item 6, list which issues you want. For fonts, use the order form from a copy of the catalogue.

Name: _____ Company: _____

Address:

City, State, Zip:

Home Phone ()

Business Phone ()

Computer format in which software should be supplied:

Total money enclosed:

SORT ALGORITHMS IN C

by Joseph W. Wright

How often have you heard the phrase self-documenting code? Here we have a real-life example. The following program by Joe Wright, which actually compiles in C and runs, is so well described by its comments that I thought no further text was needed! With no further ado, then, here it is.

/*

Program: DEMO.C
Author: Joe Wright
Date: 28 March 1991

I have implemented a number of sort algorithms for comparison, specifically an Insertion sort, the Shellsort, the Wrightsort and the Quicksort.

The sorts themselves operate on integer indices and have no knowledge of the actual data they are sorting. They depend on a global variable N (number of elements) and formal pointers to the appropriate Compare and Swap functions. Assigning N, CMP and SWP correctly will allow them to sort anything. In these examples we are sorting a simple array of integers.

Most texts compare sorts on how much time they take. This can be misleading if our interest is in the algorithm rather than the data structure or CPU. I prefer to compare algorithms based on the number of Compare and Exchange operations they require.

DEMO allows you to create integer arrays of any size (up to 16k) and sort them with the various sorts. It counts Compare and Exchange operations and displays them at the end of the sort. It also rings the console 'bell' at the beginning and end of the sort for those of you with a stopwatch. If the array is 8k or smaller, each Disorder command will save a copy of the disordered array so that it can be recalled with the Last command. In this way you can compare the sorts on identical disordered arrays.

This program is written in straight Kernighan & Ritchie C without regard to any specific compiler or operating system. I suppose it should compile and run with little or no change on your system.

Well, almost. I use the ln() 'Natural Logarithm' and ftoa() 'Float to Ascii' functions from the C/80 MATHPAK. If you don't have these simply comment out (or remove) the '#define FP' statement. You will lose an interesting report line but everything else should work.

I will appreciate any and all comment on this program. If you can improve on any of the algorithms please do so and tell me. None of this is proprietary as far as I know, and not new except for the 'Wright' number.

Joe Wright
Alpha Systems
711 Chatsworth Place
San Jose, CA 95128
(408) 297-5593

*/

```

#include < stdio.h>
#define ver 10                                /* Version 1.0 */

#define FP                                    /* If floating point ln( ) function available */

#define bell 7                                /* Console BELL character */
#define cols 16                               /* Display 16 columns of 5 characters */
#define wrap NO                               /* Send '\n' after column 80? */
#define num 256                               /* Default number of elements */
#define nmax 0x4000                          /* Maximum 16k (32k bytes) */

#ifdef FP
    float ln( );                             /* Natural logarithm function */
    float logn, nlog;                         /* Global variables */
#endif

    long comps, swaps;                       /* Global sort counters */
    int n;                                    /* Number of elements */
    int *a, *b;                               /* Two integer arrays */
    int c;                                    /* Last key pressed */
    int disp = YES;                          /* Display the array? */

    int  comp( ), swap( );                   /* Declare Actual function pointers */
    int  (*cmp)( ), (*swp)( );              /* Formal pointers to functions */

    char str[8];                             /* Space for user input */

/*
    Start of the Program
*/

main(argc,argv) int argc; char *argv[]; {

    printf("SORT DEMO Vers %d.%d", ver / 10, ver % 10);

    cmp = (*comp);                            /* Assign Actual to Formal function */
    swp = (*swp);                             /* Assign Actual to Formal function */

    if (argc > 1) {
        if (!(isdigit(*argv[1])))
            goto help;
        else
            n = atoi(argv[1]);
    }
    if (argc == 1) {
new:    printf("\n\n\tEnter No. of elements: [%d] ", num);
        if ((c = getline(str,8)) > 1) {
            str[c-1] = '\0';
            if (isdigit(*str))
                n = atoi(str);
            else {
help:    printf("\n Syntax: demo [num] < cr > ");

```



```

                                exit( );
                                }
                                }
                                else
                                n = num;
                                }

if (n > nmax) {
    printf("\n Error! N too large (max %d)", nmax);
    goto new;
}

#ifdef FP
    logn = (ln((float)n)/ln(2.0));
    nlog = n * logn;          /* N*Log2 N */
#endif

a = sbrk(0);                  /* Allocate buffer A */
b = (n < nmax/2) ? a + n : 0; /* Allocate buffer B */

fill( );                      /* Fill buffer A */
shuffle( );                   /* Mix it up a little */
first( );                     /* Copy buffer A to buffer B */
c = 'S';                       /* Show it Shuffled in report() */

while (c != 'X') {
    putchar(bell);
    report( );
    comps = swaps = 0;
    c = toupper(getchar( ));
    putchar(bell);
    switch (c) {
        case 'O' : fill( ); break;
        case 'R' : rev( ); break;
        case 'I' : ones( ); break;
        case 'D' : shuffle( ); first( ); break;
        case 'L' : last( ); break;
        case 'T' : disp = (disp) ? NO : YES; break;
        case 'N' : goto new;

        case 'Q' : qsort( ); break;
        case 'W' : wright( ); break;
        case 'S' : shell( ); break;
        case 'I' : isort( ); break;
        case 'B' : bsort( ); break;
        case 'K' : ssort( ); break;
    }
}
}

```

```

/*
    The Selection sort, simply to try it out . . .
*/

```

```

ssort() {
    static int i,j,x;
    for (i = 0; i < n - 1; ++i) {
        for (x = i, j = i + 1; j < n; ++j)
            if ((*cmp)(x,j))
                x = j;
        if (x != i) (*swp)(x,i);
    }
}

```

```

/*
    The ubiquitous Bubblesort. It is presented here only as an example of how not to sort
    anything. The inner for loop scans the entire array looking for adjacent items which are out
    of order. If it finds one (or more) it exchanges the two items and sets the 'dirty' flag to
    indicate at least one more pass is needed. When it gets through the entire array without
    setting the 'dirty' flag, the sort is complete. Worst case is about N**2 operations. Avoid it like
    plague.
*/

```

```

bsort() {
    static int i,d;
    d = 1;
    while (d) {
        for (d = 0, i = 1; i < n; ++i)
            if ((*cmp)(i - 1,i)) {
                (*swp)(i - 1,i);
                d = 1;
            }
    }
}

```

```

/*
    A simple Insertion sort for illustration of method. It is about twice as 'good' as the bubblesort
    because, having found something to swap, it backs up (goes left) looking for the next place to
    put the swapped item. Worst case is about N**2/2 operations.
*/

```

```

isort() {
    static int i,j;
    for (i = j = 1; i < n; j = ++i)
        while (((j - =1) >= 0) && ((*cmp)(j,j + 1)))
            (*swp)(j,j + 1);
}

```

```

/*
    The Shellsort is the grandfather of logarithmic sorts. Invented in 1959 by D. L. Shell, it has
    gained wide popularity as a good general purpose sort. It works by establishing a Gap (N/2)
    and comparing (and swapping) elements which are relatively far apart. It divides the gap by

```

The Computer Journal

Applications — Programming — User Support

Sh... Quiet!

...They Don't Know We're Here!

They search for ever more RAM, we build custom interfaces. They add \$300 coprocessors to compensate for bad programming, we automate our homes with \$50 controllers. They write macros to add a column of numbers, we write operating systems. Their magazines carry endless reviews of computers only a corporation can afford. Our journal publishes schematics and source code.

There are whole other worlds of computing beyond Windows 3 and DOS, but they don't know about it. Maybe you do. If S100, CP/M, Forth, embedded controllers or robotics mean anything to you, then you need to know about *The Computer Journal*.

What You Read —

TCJ is written and read by people who remember where all this started. Our articles teach the principles behind digital control. You will see real-life applications and be given the tools to do it yourself. Topics include Assembly Programming for the High Level Language Programmer, writing and using IOPs, and more. We discuss computer languages: Modula-2, C, Forth, Pascal. You will read award winning articles, such as the first place winner of the Harris RTX Design Contest.

- Embedded controller concepts, applications
- Instrumentation and control with D/A, A/D
- Motion control with DC, servos, steppers
- Use of logarithms in controllers
- Lazy evaluation
- Operating system design, modification
- Encryption techniques
- System design, interfacing
- Plus monthly columns: Jay Sage, author of ZCPR 3.4, telling you how to get the best from Z-System; Richard Rodman on Minix and National Semiconductor cpu's; Matt Mercaldo with the F68HC11; Wayne Sung on LANs. Bill Kibler keeps an eye on the future of the industry.
- Programming the 8051, F68FC11, RTX and other specialized CPUs
- Programming in Forth, Modula-2, C, Assembler
- Hardware projects ranging from interfacing a Bernoulli removable hard disk to a CP/M computer to dedicated embedded controllers
- Modifying and repairing printed circuits
- T1, X.25, related communications topics

What You Write —

The Computer Journal is just that—a journal. Our readers provide many of the articles. If you have a paper on a significant aspect of micro-computers or embedded controllers, algorithms or programming, submit it for consideration. The spirit of the individual made the computer industry. At *TCJ*, we have never forgotten that.

Where You Go From Here —

If all this quickens your pulse, then you are a *TCJ* person! Find out for yourself. Mail the coupon below for a free copy today! If you like what you see, then pay just \$18 (US rates) for another five issues (six in all). If you would rather go back to loading more TSR's, that's okay, too. Just cancel the invoice and keep the sample copy.

Mail this coupon for your first issue! Free!

- Published six times a year
- Subscriptions (check one):
 - US: ___\$18/1 year, ___\$32/2 years
 - Foreign, Surface: ___\$24/1 year, ___\$44/2 years
 - Foreign, Air Mail: ___\$38/1 year, ___\$72/2 years
- Send no money! If you don't agree that *TCJ* is for you, just cancel the invoice!

Name: _____

Address: _____

My interests: _____

TCJ The Computer Journal

Socrates Press

P.O. Box 12

South Plainfield, NJ 07080-0012 USA

(908) 755-6186

two and does it again and again until the gap becomes zero. Like the Insertion sort, when it swaps two elements it goes left looking for the next (lower) place for the new element.

```

*/
shell() {
    static int g,i,j;
    for (g = n/2; g > 0; g /= 2)
        for (i = g; i < n; ++i)
            for (j = i - g; j >= 0 && (*cmp)(j,j + g); j -= g)
                (*swp)(j,j + g);
}
/*

```

The Shellsort Gap value has been the subject of much study. The idea of starting with a relatively large one is clearly good so that we exchange items over larger distances. The Last Gap value must be unity (1) to put the final touches on the sort. The real question is how to get from N to 1. The simplest way is to divide N by two successively until 0. Although sublimely binary and beautiful, this does not yield the best result. When the current gap is exactly half of the previous one, some unnecessary compares are made. Apparently a 4-sort on an 8-sorted array will find items which have already been sorted.

There is no known way to predict these occurrences precisely because the initial order of the array is not known (otherwise, why sort?) but we can avoid the one known deficiency of the algorithm, that a gap should not be a binary factor of the previous one(s).

D. Knuth has suggested gaps . . . , 121, 40, 13, 4, 1. In fact, any choice of gaps will suffice as long as the last one is unity (1) but the nature of the algorithm wants gaps closer to $G/2$ than $G/3$.

The Wrightsort does a rigorous job of choosing the Gap by dividing by 2 and then ensuring that Gap is always Odd. It is therefore never a precise power of 2 and never an even factor of the previous Gaps. If we start with (or fall onto) a power of 2, the series looks like . . . , 63, 31, 15, 7, 3, 1. If we miss the power of 2 it looks like . . . , 125, 61, 29, 13, 5, 1.

You are invited to experiment further with the Gap but I think this will prove best in most cases. Tell me.

```

*/
wright() {
    static int g,i,j;
    g = n;
    while (g > 1) {
        if (((g /= 2) & 1) == 0) g -= 1; /* The Wright number */
        for (i = j = g; i < n; j = ++i)
            while ((j -= g) >= 0 && (*cmp)(j,j + g))
                (*swp)(j,j + g);
    }
}
/*

```

C.A.R. Hoare's Quicksort adapted from N. Wirth's Pascal example in "Algorithms + Data Structures=Programs". It is improved by maintaining the 'sentinel' within the array (no extra storage) and by avoiding the Swap when indices $i = j$.

This Quicksort is the fastest and most efficient sort we know of. Its drawbacks are that it is recursive and pattern sensitive so that we cannot predict how much stack space it will use. The Quicksort works almost equally well on in-ordered arrays and on reverse ordered arrays as well as randomly ordered arrays.

The worst case for the Quicksort is when each choice of the sentinel is the highest (or lowest) item in all partitions. We can force the case by selecting l or r as x and sorting an already ordered array. This results in about $N^2/2$ compares. Selecting x as $(l + r)/2$ avoids this problem in that the ordered array becomes the best, not the worst case. Jon Bentley says he doesn't like Hoare's Quicksort because he has never seen it implemented correctly. Have a look at this one, Jon.

```

*/
qsort( ) {
    sort(0,n - 1);
}
sort(l,r) int l,r; {
    int i,j,x;
    if (l < r) {
        i = l; j = r; x = (l + r)/2;
        do {
            while ((*cmp)(x,i) == 0) ++i;
            while ((*cmp)(j,x) == 0) --j;
            if (i <= j) {
                if (i != j) {
                    if (i == x) x = j; else
                    if (j == x) x = i;
                    (*swp)(i,j);
                }
                ++i; --j;
            }
        } while (i <= j);
        sort(l,j);
        sort(i,r);
    }
}
/*

```

I have written a non-recursive (reiterative) version of the Quicksort which does not have the stack runaway problem. It will be the subject of another article on the implementation and use of sorts.

```

*/
/*
The sorts use the same Compare and Swap routines to treat an array of N integers. They have been
assigned to the formal pointers (*cmp)( ) and (*swp)( ) above.
*/

```

```

/*
Exchange two integers in a[] and increment swaps counter.
*/

```

```

swap(i,j) int i,j; {

```

```
        static int x;
        ++swaps;
        x = a[i]; a[i] = a[j]; a[j] = x;
    }

    /*
    Compare two integers in a[] and increment the comps counter. Return True if a[i] is greater
    than a[j].
    */

    comp(i,j) int i,j; {
        ++comps;
        return (a[i] > a[j]);
    }

    /*
    The Shuffle provides a pseudo random ordering of a[] .
    */

    shuffle() {
        static int i;
        for (i = 0; i < n - 2; ++i)
            (*swp)(i,rand(n - i - 1) + i + 1);
        swaps = 0;
    }

    /*
    Initialize a[] in reverse or descending order.
    */

    rev() {
        static int i;
        for (i = 0; i < n; ++i)
            a[i] = n - i;
    }

    /*
    Initialize a[] in ascending order.
    */

    fill() {
        static int i;
        for (i = 0; i < n; ++i)
            a[i] = i + 1;
    }

    /*
    Initialize a[] to all ones.
    */
```

```

ones() {
    static int i;
    for (i = 0; i < n; ++i)
        a[i] = 1;
}

/*
    This 'random' number generator was thrown together rather quickly and is not meant to be
    anybody's model. Its results are random only to the extent that I can't predict them.
*/

rand(i) int i; {
    /* Random integer (0 .. i - 1) */
    static int rnum = 2417;
    rnum = abs((rnum/2 + 2417)*2417);
    return (rnum % i);
}

first() {
    copy(b,a);
}

last() {
    copy(a,b);
}

copy(x,y) int x[],y[]; {
    static int i;
    if (x && y)
        for (i = 0; i < n; ++i)
            x[i] = y[i];
}

/* Print a long (positive) integer, 0 .. 2 billion. */

prntl(l) long l; {
    static long i;
    if ((i = l / 10) != 0)
        prntl(i);
    putchar(l % 10 + '0');
    /* Recurse until 0 */
    /* Unwind the stack, printing remainders */
}

#ifdef FP
prntf(fl) float fl; {
    static char buf[10];
    ftoa('F',3,fl,buf);
    printf(buf);
    /* Float to Ascii, not a standard function */
}
#endif

display() {
    static int i;
    for (i = 0; i < n; ++i) {

```

SAGE MICROSYSTEMS EAST

Selling & Supporting the Best in 8-Bit Software

- Automatic, Dynamic, Universal Z-Systems: Z3PLUS for CP/M-Plus computers, NZCOM for CP/M-2.2 computers (\$70 each)
- XBIOS: the banked-BIOS Z-System for SB180 computers at a new, lower price (\$50)
- PCED — the closest thing to Z-System ARUNZ, and LSH under MS-DOS (\$50)
- DSD: Dynamic Screen Debugger, the fabulous full-screen debugger and simulator, at an incredible new price, down from \$130 (\$50)
- ZSUS: Z-System Software Update Service, public-domain software distribution service (write for a flyer with full information)
- Plu*Perfect Systems
 - Backgrounder ii: CP/M-2.2 multitasker (\$75)
 - ZSDOS/ZDDOS: date-stamping DOS (\$75, \$60 for ZRDOS owners, \$10 for Programmer's Manual)
 - DosDisk: MS-DOS disk-format emulator, supports subdirectories and date stamps (\$30 standard, \$35 XBIOS BSX, \$45 kit)
 - JetFind: super fast, extremely flexible regular-expression text file scanner (\$50)
- ZMATE: macro text editor and customizable wordprocessor (\$50)
- BDS C — including special Z-System version (\$90)
- Turbo Pascal — with new loose-leaf manual (\$60)
- ZMAC — Al Hawley's Z-System macro assembler with linker and librarian (\$50 with documentation on disk, \$70 with printed manual)
- SLR Systems (The Ultimate Assembly Language Tools)
 - Z80 assemblers using Zilog (Z80ASM), Hitachi (SLR180), or Intel (SLRMAC) mnemonics, and general-purpose linker SLRNK
 - TPA-based (\$50 *each* tool) or virtual-memory (\$160 *each* tool)
- NightOwl (advanced telecommunications, CP/M and MS-DOS versions)
 - MEX-Plus: automated modem operation with scripts (\$60)
 - MEX-Pack: remote operation, terminal emulation (\$100)

Next-day shipping of most products with modem download and support available. Order by phone, mail, or modem. Shipping and handling \$3 per order (USA). Check, VISA, or MasterCard. Specify exact disk format.

Sage Microsystems East

1435 Centre St., Newton Centre, MA 02159-2469

Voice: 617-965-3552 (9:00am - 11:30pm)

Modem: 617-965-7259 (pw=DDT) (MABOS on PC-Pursuit)


```

        if (wrap && (i % cols) == 0)
            putchar('\n');
        printf("%5d",a[ i]);
    }
    if (wrap || (n % cols) != 0)
        putchar('\n');
}

report() {
    printf("\n\n");
    if (disp) display( );
    printf("\n\tFunc = %c, N = %d: Comps = ", c, n);
    prntl(comps);
    printf(", Swaps = ");
    prntl(swaps);
    printf(", Total = ");
    prntl(comps + swaps);

#ifdef FP
    printf("\n\tScore (Log N=);
    prntl(logn);
    printf(" N*Log N=);
    prntl(nlog);
    printf("): C = ");
    prntl((float)comps/nlog);
    printf(", S = ");
    prntl((float)swaps/nlog);
    printf(", T = ");
    prntl((float)(comps + swaps)/nlog);
#endif

    printf("\n\n\tD=Disorder, O=In order, R=Reverse, 1=Ones,");
    printf(" L=Last, T=Disp O%s\n", (disp) ? "FF" : "N");
    printf("\tB=Bubble, I=Insert, S=Shell, W=Wright, Q=Quick,");
    printf(" N=New, X=Exit ");
}

/* End of DEMO.C */

```

HOW TO FEEL LIKE A FOOL (with almost no trouble at all)

by Robert Vinisky

It's the little things. The picky details. The footnotes. There I was, just minding my own business (actually I was checking out a program patch), when my hard disk went to garbageville. All but four of my A: files were now in user areas greater than 32 (where they couldn't be used), and the four were useful ones

like SB.OVL! Well, I had just gotten this Eagle IV and was thinking of reformatting the hard disk anyway, so ...

I reformatted it, tested it, and was ready to re-install NZCOM. I copied NZCOM.COM, NZCOM.LBR and MKZCM.COM onto A: and

ran MKZCM. After saving a NZCOM.ZCM (and NZCOMENV) file I was ready to boot. I ran NZCOM and soon there was the familiar A0> prompt (but without a named directory). Now I was ready to build the disk. First went CMDRUN.COM and ALIAS.CMD, then ZF.COM and ZFILER.CMD (because I use ZFILER a lot). Just for the heck of it, I then decided to go back to CP/M and reload NZCOM just to see it work. Imagine my surprise when the system went nuts!

The old familiar sequence began OK. NZCOM.ZCM was loaded, NZCOM.LBR was opened, all of the stuff went where they were supposed to go and all was well. Then, after NZCOM.CCP and NZCPM.COM were made, NZCOM came up . . .

And stayed up. And up. Forever. Until it got too much for poor old CP/M (or something) and the screen started filling up with Ovfl messages! I tried it again with the same result. I think I cried a lot then. What the heck was going on? I tried to load a NZBLITZ-created file and there was the Z-System. Hmmm . . .

I emptied the disk and ran SAPZ to clean it up (while still in the Z-System, of course) and reloaded the NZCOM and MKZCM files. Create the .ZCM file and boot it. It booted! Hmmm . . .

Maybe it was just one of those things that seems to happen just when you're thinking you *really* have this stuff figured. It could happen, you know. So I started to add files and then decided to try CP/M again. Not a good idea!

Run NZCPM, run NZCOM (doesn't that sound like a rap group?) [M.C. BIOS? D.J. BDOS? Naaaah . . . -DAJM] and then the system, along with Peter Pan and a bunch of kids, went to never-never land. [Expletive deleted]!!!

So, here's the problem. From a cold boot the system was like Elmer Fudd running around and around the tree while Bugs sat on top laughing. If the Z-System was running, however, everything was fine. I could even use NZCOM to load new systems, change modules or whatever. Just don't try to start from scratch.

Why? . . . Why? . . . Why me?

To make a long story long, here is what happened. As per custom, I had made ARUNZ my extended command processor by renaming it to CMDRUN. No problem here. Then, following the (good) advice of Jay Sage in a TCJ article, I made the last alias in my ALIAS.CMD script chain to LX.COM. Now, when a command can't be found on the path, the extended command processor (ARUNZ) gets it. If there is no alias definition that matches the command, LX gets to look in COMMANDLBR for one last try before the error handler takes over. Just as it should be, and a mighty handy thing to have around, especially with a floppy-based system. But who bothers to look over the ALIAS.CMD file before building up a new system? I know my only thoughts regarding ALIAS.CMD were how much faster the aliases would run.

Well, when NZCOM gets through building up the Z-System and is ready to boot it, the last thing it does (from a cold boot only) is look around for a file named STARTZCM.COM. In fact, in the NZCOM manual it says ignore the STARTZCM? message when you first bring up the Z-System (because you haven't created it yet). In my case, NZCOM loaded everything properly and started looking for STARTZCM. And kept looking. Remember the sequence of events; first the path, then the ECP (extended command processor), then the error handler.

Now, on startup, the internal NZCOM path is A0: \$\$: A0:. When STARTZCM.COM could not be found the ECP searched for an alias called that in ALIAS.CMD. When it was not found there, we chained to LX. All very proper, except for one little detail. I had not yet copied anything to A0: but the NZCOM files, so the command processor spent the summer looking in A0: for a non-existent LX.COM!

Of course I could have loaded the error handler (or LX.COM) before trying anything silly like returning to CP/M, but you know how it is, you get used to all of the Z-features and forget the little things. To make matters worse, I had called Joe Wright about this and could visualize him rolling around on the floor laughing like a

hyena.

The moral? Well, for the most part we have

good computers running a superb operating system. Now if we can just get the operators up to speed, we will really have something!

SOUND POTENTIALS CP/M PUBLIC-DOMAIN SOFTWARE CATALOG

We publish a catalog that lists the titles of 720 CP/M library files from which you can pick and choose. The titles are current up to 1990. The catalog lists each title, author, date, size of library, and a description. We charge you a copying fee of only five cents per 1K of library file copied. We can format for over 180 5¼" disk formats, both 48- and 96-tpi, and we charge you no extra for your format. We offer discounts for large orders if your disk format holds over 300K. You receive your selections plus our catalog on disk and a library utility to remove the files.

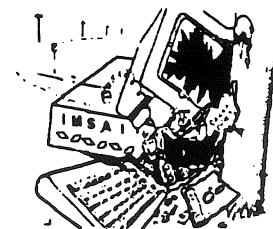
We also offer a data-copying service to copy data between any of the formats we support. Full details included with the catalog.

As a special introduction to the public domain of CP/M, we have put together a Sampler of some of the best utilities and word processing programs. This large collection is available on your disk format for \$10.00 plus \$4.00 shipping and handling. We throw in the printed catalog for \$1.00 more (\$15.00 total). If you just want the printed catalog and other info, send \$2.00.

SOUND POTENTIALS, BOX 46, BRACKNEY PA 18812

S100/IEEE-696 System Crashed? or ...

- need to upgrade your present system?
- need to buy spare boards for easy maintenance and repair?
- need to restore your system with original manufacturer's boards?
- need documents for repairs, modifications and programming?



While we are specialists in IMSAI, Compupro, Cromemco, and the Heath/Zenith Z100; we also have boards or complete systems for many other manufacturers! We also know others who have boards to sell...like maybe YOU.

"Hi, Mr. Bitz here! Look at what we can do for YOU! We can test what we sell! We want to talk to YOU about YOUR systems and needs! We have knowledge and experience in engineering and programming! AND...our prices are reasonable!"

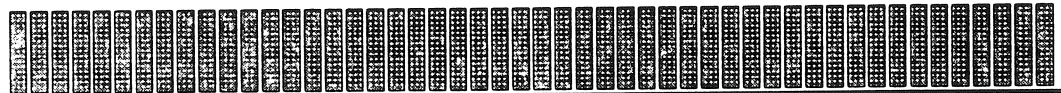
To keep prices down, we also sell non-working or untested boards. Our loss is your opportunity to experiment, modify, and tinker! Another cost saving is our lists of 3rd-party S100 equipment! If we don't have to stock it we can save you money!

"This is Mr. Bitz again! Our warehouse shelves are BURSTING WITH BARGAINS!.. Compupro STATIC RAM cards starting at \$50! 4-port serial cards at \$40! and Cromemco boards OF ALL KINDS as low as \$20! Some boards are untested or incomplete, so we've PRICED THEM TO MOVE for as low as \$10! And, I love to bargain so these prices are NOT CAST IN CONCRETE!! Remember, if I can't sell it, YOU CAN'T USE IT!"

Send for our FREE catalog today! Or call and ask for "Mr. Bitz"!

Herbert R. Johnson, 1519 Mt Everett St,
Colorado Springs CO 80909 (719) 578-0997

Layout by Night Owl Computing (719) 578-0997



SCRIPT OF THE MONTH CLUB

Saving new directory names by Jay Sage

Last time I described my NAME alias for changing the assignment of directory names to drive/user areas. This month I will describe a second alias that is used along with NAME.

Sometimes I am changing directory names just to experiment, but most often I am starting on a new activity and want to establish the new name permanently (so that it is still there after the next cold boot). Therefore, I am not satisfied with just changing the name in the in-memory named directory register (NDR). I want to update the NDR disk file as well.

Al Hawley, the author of EDITNDR, on which last month's script was based, also wrote a nice utility called SAVENDR. It takes the NDR in memory and writes it out to a file of type NDR for loading using one of the Z-System package loader utilities, such as JETLDR, LDR, NZCOM, or Z3PLUS.

On my system, to save disk space and (mainly) to reduce clutter, I keep all the system packages in a single LBR file called SYS.LBR. This contains my flow control package (FCB.ZRL), resident command package (RCP.ZRL), terminal capabilities descriptor (WYSE50.Z3T), and the named-directory register (HD.NDR). My startup alias invokes JETLDR to load the whole lot in one fell swoop.

In order to update my directory names permanently, I have to create a new HD.NDR file and then replace the old one in SYS.LBR with the new one. Here is my script, called PUTNDR, which does this.

```
PUTNDR b0: ;
      era hd.ndr;
      : savendr hd.ndr;
      lput sys hd.ndr;
      $hb:
```

The first command logs me into the directory where I keep my system files. It normally has the name SYS:, but just in case I should ever

experiment with changing it, I use the DU form of reference in this script. Also, I could have stayed in whatever directory I was in, but then I would have had to include explicit references to directory B0: in all the other commands. Since I have several things to do there, it seemed best to log into the directory.

As I recall, SAVENDR has a problem when there is already a file with the specified name, so I begin by deleting the old HD.NDR. Since it normally exists, I issue the ERA command without checking for the existence of the file. If, for some reason, it does not exist, I will just get a warning message. If I wanted to be fastidious, I could have used the following more complex sequence of commands:

```
if ex hd.ndr; era hd.ndr; fi
```

The third line runs SAVENDR. Note the colon in front of the command verb. This makes the program run from the current directory, even though neither the current directory nor B0: is on my command search path.

The fourth line uses Bruce Morgen's excellent LBR-building utility, called LPUT. There can be problems with storing the same file over and over into a library. If the new file is bigger than the old one, the space used by the old one is not recovered. The old file is marked as deleted, but the new file (with the same name) has to be stored at the end of the library. This is not very efficient. Fortunately, if the new file is the same size or smaller, LPUT simply replaces the old file in the library. The NDR file is always the same size unless a new Z-System configuration is installed. Thus, PUTNDR can be run any number of times.

The final line takes me back to the directory I was in when all this started. It was convenient to do the real work of the script in directory B0:, but I find it very annoying when operations leave me in an unintended directory. The internals of an operation like PUTNDR should

be transparent to me. The ARUNZ parameter \$H returns information about the home

directory; the B option indicates that Both the drive and user number are to be returned.

A PROGRAMMER CORNERED

Choosing the right language for the job

When you write a program, one of the most important decisions you make occurs before you touch the keyboard, or set pen to paper. The choice of which language you use can make the difference between a job being easy or difficult, or sometimes between possible and impossible.

Human languages are not the same. French always sounds like you're making love, even when you're swearing. German always sounds like you're swearing, even when you're making love.

Languages not only sound different, they have different areas in which their picture of the world is particularly detailed, making them particularly useful in those areas. English, for example, has borrowed so many words from so many other languages that it has the largest vocabulary of any language in the world.

The artificial languages we use to program computers are very limited compared with natural languages, but like natural languages, they have their areas of expertise. Where the specialties of natural languages are determined by environment (it's no accident that Finnish and Eskimo have so many words for snow and winter weather conditions), the specialties of computer languages are the purposes for which they were designed.

When writing operating-system software, speed and small size are the most important considerations. Thus operating systems are written in machine language, assembly language, or thinly-disguised assembly languages like C and Forth. Teaching programming can be done with any language, but BASIC, Pascal, and Modula-2 were designed for that purpose.

Suppose we wanted to write a program that would take a heraldic blazon such as *Ermine, a lion passant guardant between three battle axes Sable* and print this on a sheet of paper underneath a color picture of the arms described by the blazon. This could be attempted in any language, but it would be extremely difficult in any language but PostScript. More precisely, a parser program running on the computer would translate the blazon into a PostScript program, which the PostScript interpreter in the color printer would execute to create the picture. ALGOL and LISP are two good candidates for the language in which to write the parser program, because they are designed for writing parsers.

My personal favorite programming language is COBOL, and I will write a program in COBOL if it is at all suitable for the job. COBOL has an undeserved bad rap among lazy typists as a "verbose" language. What this means is that it encourages, by its structure, the kind of programs which are easy to read and understand. To add an amount received to the total amount received in BASIC, for instance, could be coded like this:

```
TOTAL = TOTAL + AMOUNT
```

In COBOL a typical version would be

```
ADD AMOUNT-RECEIVED TO TOTAL-AMOUNT-RECEIVED.
```

Gosh, that's hard to figure out, isn't it? Well worth the extra keystrokes, when you look at the program months later and have to figure out how it works.

COBOL is also the computer language that comes closest to being plain English. It is the only language I know that has sentences

ending in periods, containing clauses ending in commas, as well as verbs like ADD, MOVE, SUBTRACT, etc.

The following Nevada COBOL program is one I wrote as part of the conversion of *The Z-Letter's* subscriber data base when I got it from Joe Wright. I had already converted Joe's dBase II files to SELECTOR V files. One of the problems with Joe's files was that his customer key was a number. Sorting the records and printing the mailing labels by this key meant that they came out in order by when they were entered into the data base, which meant the labels had to be sorted by hand if, for example, I wanted to insert a colored sheet in a particular envelope with a reminder that the subscription was expiring. Why not sort by the last-name and first-name fields? Fine in theory, but in practice these fields contain titles like Dr., Pastor, and SSG, or things like Jr., III, Esq.

What I really needed was a key based on last name and first name, but containing no extraneous honorifics, initials, or other confusion. This COBOL program produces just such a key. The key I chose was the first 7 letters of the last name, the first 5 letters of the first name, and a two-digit number for distinguishing those cases where the 12 characters came out the same. Sorting by this key would be the same as sorting by last name and first name, without the extra baggage found in those fields.

This is a very typical COBOL program, in that it goes through a file from beginning to end, producing a second file based on the first. COBOL was designed for processing records batched into files, which Pascal courses regard as an advanced topic to be taught late in the semester.

There are two other reasons to use COBOL here, besides the fact that COBOL is designed to go through a file until it reaches the end.

One reason is that COBOL is designed to work with fields. Look at our work record (lines 1180-1340 of our program). Most of the fields of each record are the same as those of the input record (lines 690-800). In most languages we would

have to copy each field one at a time. In COBOL we define the parts of the work record that are the same as the input record as a single entity, called WR-COPY, which is further broken down into the individual fields for clarity (and in case we want to change the program later on). To copy the input record to this part of the work record, we use a single statement, MOVE INPUT-RECORD TO WR-COPY.

The other level-5 item that makes up WORK-RECORD is WR-CUSTID. This is our new key. It is further broken down into WR-NAMEID and WR-NUMERID. The latter we simply define as a constant 50; conflicts are resolved after the program finishes by looking through the new records. This is acceptable for a one-time program to convert a small data base; if the program were run often, or the data base were very large, we'd have to put a paragraph in the Procedure Division to check whether each finished key were identical to any already generated, and assign a different number to the newer key, or ask the user for numbers for both conflicting keys.

WR-NAMEID is further broken down into WR-LASTID, which is for the first 7 characters of the last name, and WR-FIRSTID, which is for the first 5 characters of the first name. When we copy the last name of each input record into WR-LASTID, COBOL *automatically* chops off any extra characters. Other languages would require that we specify that only 7 characters be copied. Similarly copying the first name of each input record into WR-FIRSTID automatically truncates the first name.

The second reason to use COBOL is its INSPECT verb. While Pascal could shift the new key to capital letters with much less typing, we are doing more than that with INSPECT:

1. First we go through the whole set of 12 characters, changing lower-case letters to capitals (so that Joe and joe don't alphabetize differently). We also change all periods and commas to spaces. See how doing this operation on WR-NAMEID avoids having to do it twice,

THE STAUNCH 8/89'er

--> Resources for Heath H-8 and Heath/Zenith '89 Users <--

Bimonthly Newsletter

This quality, 12-page newsletter is prepared on a daisywheel printer in two-column format and is commercially printed. Content includes letters, reviews, hardware and software articles, programming information on a number of languages (assembler, BASIC, C, and Pascal), and vendor reports. Recent issues feature the first two parts of a series on professional software testing (using Pascal in its examples). Another series on home repair of the '89 begins later in the year. **Staunch** supports CP/M 2.2, HDOS 2.0, HDOS 3.0, and the just-released HDOS 3.02 (listed below). Most issues feature a list of newly-available, public domain software, much of it written by **Staunch** subscribers or released by vendors. Most back issues are still available. Subscriptions are \$12/calendar year.

HDOS Systems--Require Heath Hardware

HDOS SYSTEM (By Heath Co.)	Version 2.0	\$15
HDOS MANUAL (By Heath Co.)	Version 2.0 (Prepared by Daniel N. Jerome)	\$72

The last version produced by Heath Co. and placed in the **public domain** (source and object code plus documentation) in the summer of '88 at **Staunch's** request. The distribution system is a 4-disk set. The manual for ver. 2.0 may be had in either machine-readable or hardcopy form. Individual chapters (1 thru 7) are available separately. Chapters 1, 2, and 7 have been extensively updated from Heath's original (1979/80) edition to cover soft-sector media (introduced in '81) and undocumented features.

HDOS (By W.G. Parrot and R.L. Musgrave)	Version 3.0	\$25
HDOS (By Richard L. Musgrave)	Version 3.02	\$75

These enhanced versions of HDOS add more commands to the command processor (SYSCMD.SYS) and system utility (PIP.ABS), introduce or improve batch processing, and introduce user areas. Unlike ver. 2.0, they **remap** memory, so **require** an H-8 capable of booting CP/M. Virtually all '89's but the very earliest will boot these systems. They are compatible with most software written for ver. 2.0. Ver. 3.0 **presumes** knowledge of 2.0 and the on-disk documentation is minimal. **Staunch** includes a 38-page booklet with the package as a more detailed reference guide. Ver. 3.02 includes an 1,100-page manual (written by Daniel N. Jerome) with 3-ring binder and is shipped UPS in the contiguous 48 states.

Software for CP/M and HDOS

Lucidata Pascal (By D.Gibby and L.Reeve)	Version 3.8	\$25
--	-------------	------

A substantial subset of the language developed by Niklaus Wirth. The compiler translates your ASCII source code file into a p-code file. The latter is then interpreted by a run-time system. If execution speed or program independence from the run-time package is critical, a command-line pragmat may combine the run-time system with the p-code file or an included p-code program can translate p-code files to source for Microsoft's M80 assembler. Not as extended as Borland's **Turbo**, nor as fast during compile. A very reasonable alternative to **Turbo** for CP/M; the fullest Pascal implementation presently available for HDOS. Particularly nice for porting software between the two systems. No special terminal is required. Includes a 100-page manual.

MONEY\$WORTH (By David Powers)		\$6
--------------------------------	--	-----

A financial package for setting up categories, recording, and summarizing income and expense transactions. The categories are user-defined. A budget forecasting feature uses a spreadsheet-like display. Data files are ASCII, so easily edited. Requires MBASIC and the H-19/89 terminal.

Staunch General Software Catalog		\$6
---	--	-----

Staunch's holdings are too extensive to list here. An on-disk catalog of software for CP/M and HDOS is available. Much of the software has been released to **Staunch** by various vendors as they concentrate on PC's or close down. It also includes material contributed by **Staunch** subscribers. The catalog **continues** to grow!

Note: Prices include shipping in the continental U.S., first-class for most items, UPS for the HDOS 3.02 package. Supported disk formats are Heath/Zenith soft-sector (H-37) and 10-hard-sector (H-17); most 40-track, single- or double-sided, soft-sector CP/M (such as AMPRO, Cromemco, Kaypro, Osborne, Televideo, or Xerox); and PC-XT.

Kirk L. Thompson

Editor, **The Staunch 8/89'er**

P.O. Box 548, West Branch, IA 52358

Voice: 319-643-7136 (eves and weekends)

once on WR-LASTID and once on WR-FIRSTID. Also, since the last two characters of the key are numeric, and not included in WR-NAMEID, we don't waste any time on it.

2. Next we find the first space in WR-LASTID, and change everything after it to spaces. Say the last-name field of the original record had been "Cox, Esq.". After line 3400 of our program had been executed for this record, WR-LASTID would have contained "Cox, Es" (the first seven letters of the original field). After the first INSPECT statement, WR-LASTID would be transformed to "COX ES". The second INSPECT statement would transform WR-LASTID (and only WR-LASTID) to "COX ". Mission accomplished!
3. A third INSPECT statement does the same thing for WR-FIRSTID. In this way a first-name field of "J. Edmund" would become a key of "J ", just as it should.

Preceding titles have not been eliminated by this processing. "Reverend John Doe" now has a key of "REVER", not "JOHN ". There are too many possibilities and special cases to put in a one-time program, but they could be added easily enough.

The structure of a simple COBOL program like this one is very straightforward. It begins with

```
0010 IDENTIFICATION DIVISION.
0020*
0030 PROGRAM-ID.
0040   CUSTCNVT.
0050*
0051 AUTHOR.
0052   DAVID ANTHONY JOSEPH MCGLONE.
0053*
0054 DATE-WRITTEN.
0055   15 DECEMBER 1990.
0056*
0057 DATE-COMPILED.
0058   15 DECEMBER 1990.
0059*
0060* This program converts the SELECTOR data file
```

certain required comments, such as PROGRAM-ID and AUTHOR, and the FILE-CONTROL which defines both streams of data as sequential disk files. Next comes the Data Division, also required. In the File Section the structure of the records of the input and output files are defined. Our work record, where we copy most of each input record and create our new key, is defined in the Working-Storage Section.

Finally, in the Procedure Division are the statements that read the input file and create the output file. The main paragraph, 0100-MAINLINE, opens the input file read-only and the output file write-only, and reads the first record of the input file. It performs the statements in 0200-CONVERT-INPUT-FILE on each record of the input file until it reaches the end of that file. Then it closes both files and stops. Pretty simple, right?

The paragraph 0200-CONVERT-INPUT-FILE copies the input record to WR-COPY of the work record, then copies the last name and first name of the input record to their respective sections of the new key and transforms them as I already described. Then it copies the completed work record to the output record, writes the output record to the output file, and gets the next record of the input file. Again, pretty simple. It should not surprise you to learn that this program compiled without errors the first time.

0070* ZLTRCUST.DAT to a new file NEWCUST.DAT. The
 0080* new data file has the same structure as the
 0090* old, except that a new key CUSTID has been
 0100* added by this program.

0310*

0320*

0330 ENVIRONMENT DIVISION.

0340*

0350 CONFIGURATION SECTION.

0360*

0370 SOURCE-COMPUTER.

0380 8080-CPU.

0390 OBJECT-COMPUTER.

0400 8080-CPU.

0410*

0420 INPUT-OUTPUT SECTION.

0430*

0440 FILE-CONTROL.

0450*

0460 SELECT INPUT-FILE ASSIGN TO DISK

0470 ORGANIZATION IS SEQUENTIAL

0480 ACCESS MODE IS SEQUENTIAL

0490 RECORD DELIMITER IS STANDARD.

0500*

0560 SELECT OUTPUT-FILE ASSIGN TO DISK

0570 ORGANIZATION IS SEQUENTIAL

0580 ACCESS MODE IS SEQUENTIAL

0590 RECORD DELIMITER IS STANDARD.

0600*

0610*

0620 DATA DIVISION.

0630*

0640 FILE SECTION.

0650*

0660 FD INPUT-FILE

0670 LABEL RECORDS ARE STANDARD

0680 VALUE OF FILE-ID IS "A:ZLTRCUST.DAT".

0690 01 INPUT-RECORD.

0700 05 IR-CUSTNO PIC X(06).

0710 05 IR-FIRSTNAME PIC X(20).

0720 05 IR-LASTNAME PIC X(20).

0730 05 IR-COMPANY PIC X(40).

0740 05 IR-STREET PIC X(40).

0750 05 IR-CITY PIC X(30).

0760 05 IR-STATE PIC X(20).

0770 05 IR-ZIP PIC X(10).

0780 05 IR-COUNTRY PIC X(20).

0790 05 IR-PHONE PIC X(20).

0800 05 IR-DATE PIC X(08).

0900*

0910 FD OUTPUT-FILE

0920 LABEL RECORDS ARE STANDARD

```

0930  VALUE OF FILE-ID IS "A:NEWCUST.DAT".
0940 01 OUTPUT-RECORD          PIC X(248).
1130*
1140 WORKING-STORAGE SECTION.
1150*
1160 01 WS-MORE-INPUT          PIC X(03)          VALUE SPACES.
1170*
1180 01 WORK-RECORD.
1190 05 WR-CUSTID.
1195 10 WR-NAMEID.
1200 15 WR-LASTID             PIC X(07).
1210 15 WR-FIRSTID           PIC X(05).
1220 10 WR-NUMERID           PIC 9(02)          VALUE 50.
1230 05 WR-COPY.
1240 10 WR-CUSTNO             PIC X(06).
1250 10 WR-FIRSTNAME         PIC X(20).
1260 10 WR-LASTNAME          PIC X(20).
1270 10 WR-COMPANY           PIC X(40).
1280 10 WR-STREET            PIC X(40).
1290 10 WR-CITY              PIC X(30).
1300 10 WR-STATE             PIC X(20).
1310 10 WR-ZIP               PIC X(10).
1320 10 WR-COUNTRY           PIC X(20).
1330 10 WR-PHONE             PIC X(20).
1340 10 WR-DATE              PIC X(08).
3210*
3220*
3230 PROCEDURE DIVISION.
3240*
3250 0100-MAINLINE.
3255  OPEN INPUT INPUT-FILE.
3265  OPEN OUTPUT OUTPUT-FILE.
3270  READ INPUT-FILE
3271  AT END MOVE HIGH-VALUES TO WS-MORE-INPUT.
3280  PERFORM 0200-CONVERT-INPUT-FILE
3281  UNTIL WS-MORE-INPUT IS EQUAL TO HIGH-VALUES.
3340  CLOSE INPUT-FILE.
3350  CLOSE OUTPUT-FILE.
3360  STOP RUN.
3370*
3380 0200-CONVERT-INPUT-FILE.
3390  MOVE INPUT-RECORD TO WR-COPY.
3400  MOVE WR-LASTNAME TO WR-LASTID.
3410  MOVE WR-FIRSTNAME TO WR-FIRSTID.
3420  INSPECT WR-NAMEID REPLACING ALL "a" BY "A"
3430  "b" BY "B"
3440  "c" BY "C"
3450  "d" BY "D"
3460  "e" BY "E"
3470  "f" BY "F"
3480  "g" BY "G"
3490  "h" BY "H"

```

```

3500      "i" BY "I"
3510      "j" BY "J"
3520      "k" BY "K"
3530      "l" BY "L"
3540      "m" BY "M"
3550      "n" BY "N"
3560      "o" BY "O"
3570      "p" BY "P"
3580      "q" BY "Q"
3590      "r" BY "R"
3600      "s" BY "S"
3610      "t" BY "T"
3620      "u" BY "U"
3630      "v" BY "V"
3640      "w" BY "W"
3650      "x" BY "X"
3660      "y" BY "Y"
3670      "z" BY "Z"
3680      " " BY " "
3690      ". " BY " ."
3700      INSPECT WR-LASTID REPLACING CHARACTERS BY " "
3710          AFTER INITIAL " ".
3720      INSPECT WR-FIRSTID REPLACING CHARACTERS BY " "
3730          AFTER INITIAL " ".
3740      MOVE WORK-RECORD TO OUTPUT-RECORD.
3750      WRITE OUTPUT-RECORD.
3760      READ INPUT-FILE
3770          AT END MOVE HIGH-VALUES TO WS-MORE-INPUT.
9997*
9998 END PROGRAM CUSTCNVT.

```

LETTERS

1 May 1991

Dear David:

I am pleased to report that the Spellbinder that I ordered arrived in good shape. I have since made working copies of the diskettes and my Apple reads them as they should be read. I have worked through the first simple exercise in the well-written User's Manual, and I look forward to seeing what I can do with the macros.

After I have become sufficiently familiar with the system, I propose to take the liberty of sending you my personal appraisal of the relative merits of WordStar and Spellbinder. As an up-to-now addicted user of WordStar, I shall nevertheless do my best to be objective.

Sincerely,
David H. Fax
6558 Beacon Street
Pittsburgh PA 15217-1843

I'm glad to have confirmation that the Matchpoint card I added to my Tandem 6AX can now write Apple CP/M disks correctly. I had no reliable way of testing the resulting diskettes locally. Please do send your writeup of WordStar vs Spellbinder. I'd be very interested in the opinion of a long-time WordStar user.

May 8, 1991

Dear Dave,

Nice job with *The Z-Letter!* I enjoy it very much

and I appreciate the time and effort it takes to deliver such a high-caliber product.

I was wondering whether there were someone out there who could possibly help me with a problem. I have an Ithaca Intersystems S-100 system which found its way to my door. This seems to be an interesting piece of hardware. I have no documentation. It looks like this system is a Z80B with possibly 512K of memory (two 256K boards), a floppy controller, and a SCSI interface, with an ST-506 hard-disk controller (hard disk missing). It came with original MP/M disks. It used to boot MP/M from the floppy, but my expert (?) knowledge of CP/M caused the demise of the floppy's "bootability".

I would like to obtain any information on this machine, and would love to either get CP/M running (BIOS source anyone?) or have someone restore the integrity of my MP/M boot disk. I know nothing of MP/M, so manuals would be helpful here too.

This is not a hot project, nor do I want to invest a lot of time and money. I just hate to have a potentially useful computer sit and collect dust. It looks like it could be fun to get running. Any help would be appreciated. I can be reached by BBS at (203) 673-8752, by voice at (203) 585-9447, or on GENie (UserID RSWENTON), or by mail at the address below.

Kind regards,
Rick Swenton
106 Melinda Lane
Bristol CT 06010-7176

*Thanks for the kind words, Rick, and good luck with your project. Have you phoned Herb Johnson? This is the sort of system he specializes in. As for MP/M, keep an eye out for an Altos system and copy its manuals. MP/M is also covered in Rodnay Zaks' *The CP/M Handbook with MP/M*, of which I still have a few copies.*

Getting an old CP/M system up and running can be quite an adventure. Often the only thing wrong with the system is that the manuals and floppies have been lost. Sometimes the system has a hard disk but won't work because the boot tracks have been wiped by

someone playing with SYSGEN without knowing the drive assignments. One SYSGEN to drive A: from a blank diskette, assuming that A: is the floppy disk drive as on most PCs, and the system tracks on the hard disk are blanked out! I just acquired an Excelan Nutcracker, which is an S-100 system with a hard disk and floppy, on which I suspect this is just what happened.

7 May, 1991

Dear Dave:

Thanks for running my *Staunch* ad in issue #7. I also found your comments on CP/M disk formats interesting and would like to add some remarks to yours.

If a user is running CP/M on a Heath/Zenith '89/90 or H-8 with the soft-sector controller, Anapro Corp. has a pair of packages that permit him to read/write/initialize many CP/M formats, as well as IBM-XT. The name of the former is EMULATE; it supports Heath's (and clones'), Magnolia's, and C.D.R.'s soft-sector controllers and some 60 formats in single or double density, single or double sided, and 48 and 96 tpi. Included are *all* well-known makes and models. The only limitation I've noted is in reading some DS formats, such as Kaypro, produced on the originating machine. Heath's H-37 controller, at least, expects side numbers on track 0; since the Kaypro 4 doesn't write these, EMULATE can't distinguish sides. However, if disks are prepared *with* EMULATE for the Kaypro 4, there's no trouble; Anapro's formatting program writes the side numbers Heath's controller requires and the Kaypro 4 ignores them when reading the disk. EMULATE doesn't support any hard-sector format, nor Apple, Commodore, Northstar, or eight-inch.

As for moving software to CP/M on the Commodore, I have no experience with the 64. But I have provided software to a 128 owner by formatting for the Osborne 1. He apparently had Juggl'r or a similar package. From what I've seen of the 64, although its native SS format has high capacity, the disk drives are cheaply made. One of my brothers has one, and he's *always* having trouble with the drives!

SPECIFICATIONS

Ampro Z80 Little Board/PLUS

CPU:

4 MHz Z80A, 8 bit-microprocessor

MEMORY:

64 kilobytes of dynamic RAM
4-32 kilobytes of EPROM

TIMER:

Z80A CTC (4 channels)
2 channels not used by Ampro software

SERIAL I/O:

Z80A SIO/0
Two RS-232C compatible ports
Software controlled baud rates
Channel A - 75 to 38,400 baud
Channel B - 75 to 9600 baud
Four standard RS-232C signals per port
Data Out
Data In
Handshake Out
Handshake In
Two ground pins

PARALLEL I/O:

Centronics-compatible printer port
Ten signals supported
Data Bits 1-8 - Output
Data Strobe - Output
Printer Busy - Input
12 ground pins

DISK I/O:

Drives supported: 1 - 4
Disk Controller: WD1772
Data Rate: 250k bps (MFM),
125K bps (FM)
Sector Size: 128, 256, 512, or
1024 bytes
Phase locked loop: digital (8 MHz)
Write precompensation: Software enabled
Drive capacity (formatted):
Type 1 (40 track, 1 side) - 200K bytes
Type 2 (40 track, 2 sides) - 400K bytes
Type 3 (80 track, 1 side) - 400K bytes
Type 4 (80 track, 2 sides) - 800K bytes

SCSI/PLUS BUS INTERFACE:

SASI Compatible
ANSI X3T9.2 (SCSI compatible)
SCSI/PLUS Initiator compatible
Uses NCR 5380 SCSI bus controller

POWER:

Same power connector and voltages as 5¼" disk drives.
+5VDC at 0.95A
+9 to +12VDC at 0.05A

ENVIRONMENT:

Temperature: 0 to 32° C, operating
Humidity: 5 to 95%, noncondensing
Altitude: 0 - 10,000 feet

SIZE:

7.75" x 5.75" x 0.75"

SOFTWARE:

Boot program in 2732 EPROM (standard)
Options (see price list for details)
CP/M 2.2 with ZCPR3 enhancements
Little Board/Plus system utilities
BIOS and utilities source code

DOCUMENTATION: (Optional)

Little Board/Plus Technical Manual
Little Board Plus Software Manual

EXPANSION MODULE:

The Ampro Z80 project board is available for special purpose I/O. The board stacks on top of the Little Board and plugs into the CPU socket and provides breadboard space for wire-wrap applications.

Exclusive manufacturing rights for the Ampro Z80 Little Board have been purchased from Ampro by Davidge Corporation. Technical support and repair service is available directly from Davidge. Ampro no longer supports the product.

OEM PRICE LIST

AMPRO Z80 LITTLE BOARD

Manufactured under license by Davidge

HARDWARE

A60060-2	Ampro Series 1B Little Board Plus Computer	250.00
A60060-3	Ampro Little Board without SCSI	240.00
A60156	Project Board/80	75.00

SOFTWARE

A60101-1	CP/M and ZCPR3 (5½", 40 track disks)	65.00
A60101-2	CP/M and ZCPR3 (5½", 80 track disk)	65.00
A60101-3	CP/M and ZCPR3 (3½" disk)	75.00
A60103-1	CP/M, ZCPR3, BIOS Source (40 track disks)	100.00
A60103-2	CP/M, ZCPR3, BIOS Source (80 track disks)	100.00
A60103-3	CP/M, ZCPR3, BIOS Source (3½" disk)	110.00

LITERATURE

A74010	Little Board/Plus Technical Manual	15.00
A74025	Project Board/80 Technical Manual	10.00
A74006	Z80 System Software User's Manual	15.00
A74015	Z80 Hard Disk Software User's Manual	15.00
A74022	Z80 Hard Disk Backup Software Technical Manual	10.00
A74011	CP/M 2.2 Manual	15.00

REPAIR SERVICE

Flat rate repair for any serviceable Little Board	75.00
---	-------

VOLUME DISCOUNTS

10-24 units - 5%; 25-49 - 10%; 50-99 - 15%; 100+ - 20%

Prices are in US dollars. All products shipped FOB Buellton, CA. Prices effective 10-1-90 and subject to change without notice. All orders are shipped UPS Blue, C.O.D. unless other arrangements made at time of order.

Anapro's package for reading, writing, or initializing PC diskettes is called CPC, and supports standard 48-tpi XT format, plus 8- or 9-sector (MS-DOS 1.0 and 2.0), SS or DS, and 48- or 96-tpi. Prices for both these packages are very reasonable. And Anapro's programmer proprietor is now merging EMULATE and CPC into one utility package. For further information, contact:

Peter Shkabara
Anapro Corp.
P.O. Box 1987
Blythe, CA 92226
(619) 922-3919

Finally, you're quite correct when you note that a whole-disk copy program will also transfer the boot tracks to a diskette. I regularly run two, for Heath's standard SS SD hard-sector and for soft-sector formats. The former is public domain, and though it runs under HDOS 2.0, I have no trouble producing bootable CP/M disks with it. The latter program is available on the "Soft-Sector Support Package" disk (885-1127[-37]) available from the Zenith Users Group. This, too, is for HDOS. Though it has a few limitations (such as locking up the system when attempting to duplicate the 18 sector per track format that some HDOS floppy-disk device drivers produce), I've found it quite reliable even when reproducing CP/M diskettes. In my experience, the only thing you must bear in

mind when using these kinds of programs is that the *destination* disk drive *must* have the same number of sides and tracks per inch as the source drive. You can really do a number on the destination drive if you attempt to duplicate a 96-tpi disk in a 48-tpi drive!

Not all copy programs will transfer the boot tracks reliably. Zenith included one such with its CP/M, but this program *can't* be used to prepare bootable HDOS disks! Apparently there are some bytes on track 0 that are set by HDOS when a disk is SYSGENed, which aren't replicated by some of these programs. A word to the wise . . .

Sincerely,
Kirk L. Thompson
Editor, *The Staunch* 8/89'er
P.O. Box 548
West Branch IA 53258

Thanks, Kirk, for sharing this information. Jerry Davis and I are installing a Compaticard IV in my Tandem 6AX. This should give me the ability to write single-density and 8" formats. Once that is done, I will see whether Sydex' ANADISK program gives me the ability to duplicate CP/M boot disks. If it does on the machines I have, I'll be soliciting boot disks for various machines, so that I can supply CP/M boot disks for as many brands and models as possible. I'll keep you informed.

PERSONAL ADS

Eagle PC-2 for sale

Working Eagle PC-2 with MS-DOS, PFS:Write, Typing Tutor, and Tandy DMP106 printer. \$275. Call Terry Saxe, (408) 947-1975.

H89 for sale

Working H89 with Magnolia disk controller running three half-height soft-sector floppy-disk drives in an external box. The original full-height hard-sector floppy-disk drive is still in the case. Full documentation, software. \$50 or

best offer. Call Floyd Knapp, (408) 996-1444.

Communications program wanted

Looking for any telecommunications program that runs under the CP/M-86 operating system. Please call Jimmy Childers in Charlotte NC at (704) 399-8404 extension 17 during business hours East Coast time if you have one. [Note: this is 16-bit CP/M and will not run CP/M or MS-DOS programs! - DAJM]

MAGAZINE ARTICLES

Desktop goodies

Actually, the name of the magazine is *Desktop Communications*. But the May-June 1991 issue has so many interesting articles that it argues for the other title. These include *Working With EPS Files* by Ross Smith, *Getting From Here to There: Translating Graphic File Formats* by Rick Cook, and *Type It Right* by Albert Dommer. This last is a review of three books on typesetting on a computer, with emphasis that a computer and laser printer are not a typewriter. Hence certain tricks used on a typewriter, such as -- instead of -, should be unlearned. I don't agree with all their recommendations, but they're definitely worth thinking about.

Daniel Will-Harris has an article in this issue called *True Confessions: A Consultant Tells All*. In it he gives tips on dealing with salesmen in computer stores and with consultants. For example, asking questions to which you already know the answers is a good way to find out whether a salesman knows what he's talking about. Common sense is the rule here, as when he reminds you that every store to go to will tell you something different, depending on what they have in stock, what they know how to use, and what they make the most profit on. It should be no surprise that people selling stuff at a markup of 100% or more will try to talk you into buying what they make the most money from. The article ends with a quiz for computer salespeople and consultants. There is some humor in it, but even the deliberately silly questions have a point behind them. The author knows what he's talking about.

Incidentally, the inconsistency in which words in the articles' titles are capitalized is *Desktop Communications*, not mine. I quoted them exactly as they appeared.

Not much use

As part of the research for buying a new laser printer (more on that in future issues), I bought the May 1991 *Buyer's Guide to the*

Home Office. This proved to be a mistake when I got it home and began reading it more thoroughly than I had in the store. Basically, I paid \$4.95 for proof of a new adage. You've heard that a little knowledge is a dangerous thing? Know now that "A little knowledge is an excuse to publish a magazine."

I had expected that all the computers in this thing would be PCs. But I thought the section on laser printers would give me some information on the printers I was examining. No such luck! Most of the models I've been contemplating weren't covered. Those that were covered were covered so superficially that they might as well have just printed the manufacturer's brochures. Indeed, the manufacturers' claims were often repeated with no critical appraisal. The IBM LaserPrinter E, for example, is said to be impressive in its speed. Yet its output speed is a mere 5 pages per minute, compared to my LaserJet Plus' 6 ppm, or a LaserJet II's 8 PPM. In fact, nine of the fifteen printers reviewed had faster output speeds than the IBM printer. Other worthless buzz words like "DOS compatible" appear. Except for printers that can only be hooked up to a Macintosh, any printer can be hooked up to any computer. If this were not so, I would out of luck using a laser printer with my system.

In addition to including printers that are *not* laser printers in this section, this magazine gives no useful information on emulations. Most laser printers have an HP LaserJet emulation. But which model are they emulating? Are they emulating the original LaserJet, which couldn't use downloaded SoftFonts? The LaserJet Plus, which has limits on how large a font it can print, how many can be downloaded at once, and how many can appear on a single page? The LaserJet II, which has no limits except how much memory is installed? The LaserJet III, which also accepts scalable typefaces of HP design? The specifications here rarely say, and it's important. There is also no indication of how *well* these emulations work. If a particular feature you use is not implemented, or does

not work correctly, the emulation is not good enough to let you do your work.

I could go on and on, but it would be a waste of time. Don't buy this.

Software creeps in everywhere

Maybe it's my own orientation as someone who can barely swap one disk drive for another, but it seems to me that software is creeping into places it was never welcome before. An example of this is issue 19 (February/March 1991) of *Circuit Cellar INK*, a long-standing refuge (if you count all its years as a column in *Byte*) for engineers whose favorite programming language is solder. Most of the hardware articles in this issue deal with PC subjects: 80x86 instruction timings, using the PC's printer port for networking, etc. Exceptions are software articles that are either generic or specific to built-in microcontrollers: Chris Ciarcia's *Digital Image Processing*, on software image-processing techniques, for instance.

Then there's a section on software languages for the 8051 controller. M. Scott Martin, Tim McDonough, and Curtis Franklin, Jr. evaluate three C compilers in *Oh Say, Can You C?* Ed Nisley and Ken Davidson debate whether using high-level languages to program microcontrollers is efficient in two articles, with Ed turning thumbs down and Ken pleading a limited, qualified affirmative.

Finally, Scott Robert Ladd's *Making Hash* is an out-and-out software article on defining and using hash tables for fast, efficient data retrieval. His clear, concise example code is written in Modula-2.

Monthly delight

Computer Monthly continues to be the best magazine for the user of CP/M and Z-System computers, after *The Computer Journal*, *The Z-Letter*, and the newsletter put out by the best-organized user's group for your computer. It has regular columns on Apple IIs, Coleco Adams, Commodore 64 and 128, TRS-80, and other computers that can run CP/M, besides

covering Atari STs and the usual PC and Macintosh stuff. Another bonus is the profusion of ads from mail-order companies, which give you a good idea what's available (in printers, for instance) and what it's really selling for. The ads include CP/M vendors like Elliam and Sound Potentials, and Pickles & Trout selling CP/M for the TRS-80. The extensive user-group listings include what computers or software each group supports or specializes in.

To give you a taste, the April issue's Adam News column by Faye Deere includes news about ADAMCON 03, to be held August 1-4 in South Bend, Indiana; sources for disk drives; a CP/M upgrade called T-DOS; the addition of software from Sound Potentials to the ECAUG library; and much more. Apple II hardware enhancements include the Curtis Keyboard Calculator, a small solar-power calculator for \$10 that could be added to any other computer's keyboard or monitor just as easily. Users of TRDOS will be glad to hear that System 1.5 is available for \$39.95 from TRSTimes Magazine. For details on these developments and more, get *Computer Monthly* at your local computer store, or write to the address below.

The May issue contained *SCSI: For More Than Just Mass Storage*, by Charles Emerson. Tidbits on SCSI-2, floppy-disk drives with embedded SCSI controllers, and 20Mb SCSI floppy disks, "floptical" drives, scanners, and laser printers are sure to start you drooling and make you think. The best thing about SCSI is that it isn't system-specific; our computers can use this stuff as well as PCs and Macs, if not better.

Other columns and articles in the May issue include programming Commodore 128 screen colors in CP/M, by Bill Roberson; tips on an Adam printer bug, more on ADAMCON 03, and Taxhelper 1990 by Faye Deere; a review of 4DOS, an MS-DOS COMMAND.COM replacement with some of the features of the Z-System, by John Fox; changing the Commodore BIOS, by Alan R. Dickey; and an excellent review of DosDisk by Samuel Vincent. DosDisk lets certain CP/M and Z-System computers use an MS-DOS floppy

without converting to a CP/M format first, an increasingly useful ability these days.

The current subscription price for *Computer Monthly* is 12 issues for \$15.95, 24 issues for \$22.25, for subscribers in the continental U.S. Since the cover price is \$23.40 per year in the U.S., or \$30 in Canada, that's quite a deal. The address is Computer Monthly, P.O. Box 7062, Atlanta GA 30357-9827. Foreign subscriptions are \$60 per year. Back issues are probably available, but I couldn't find anything in the masthead.

Tips for Epson owners

The Epson Lifeboat from NEUG (the National Epson Users Group) covers all models of Epson printers and computers, and all operating systems and software usable on Epson computers, including MS-DOS, CP/M, and Valdocs. Increasingly, however, the CP/M and Valdocs nuggets must be mined from huge tailings of MS-DOS dross. In the April-May issue my prospecting uncovered a list of QX-10 bulletin-board numbers from Jerry R. Ives (pg. 27), and a description by Dick Monroe of how he produces a newsletter using his QX-10, Peachtext 5000, and a Silver-Reed 550 daisy-wheel printer (pp. 40-41). Pages 46-49 seems to be the mother lode this issue, with a short reprint from a Minnesota EUG on enlarging the QX-10 monitor display, new products for the Geneva and QX-10 described by Greg Tarnowski, and an actual article on using the

QX-10 diagnostic disk.

NEUG also has a Co-Op, where members can buy stuff dumped by other members or by manufacturers. Used QX-10s are said to be available (no prices were quoted), as well as *lots* of software, Epson having sent all its software to liquidators. HASCI GraphPlan, HASCI MicroPlan, HASCI dBase II, and PeachText 5000 for CP/M are available for \$25 each. PeachTree Mail List Manager, PeachCalc, Tycoon (a money game), Acorn Logo Professor, and PeachText Word Processor are \$10 each. NEUG also offers a parts replacement service for QX users who have been a member for a continuous 13 months or more.

NEUG's address is Box 1076, Lemont PA 16851. Annual dues (which include *The Epson Lifeboat*) are \$26 in the U.S. Write NEUG for other rates and details.

***Futurebus+ Design* founded**

Jay Vilhena and company, the publishers of *SUPERMICRO*, have started a magazine for Futurebus+. So far issues 1 (Jan/Feb 1991) and 2 (Mar/Apr 1991) have appeared. This is heavy stuff for hardcore hardware hackers. If that's your meat, you may wish to write to Futurebus+ Design, 1275 N. University Ave., Unit 7, Provo UT 84604. Subscriptions are \$14 for 6 issues in the U.S., or \$18 per year in U.S. currency for orders outside the U.S.

Advertisers this issue
Alphabetical by advertiser

The Computer Journal	11
Davidge Corporation	29
Elliam Associates	35
Herbert R. Johnson	19
Lambda Software Publishing	5
Sage Microsystems East	16
Socrates Z-Node 32	39
SOUND POTENTIALS	19
The Staunch 8/89'er	23

CP/M Software

*For most CP/M computers
including the Amstrad PCW*

Public Domain Software

100+ Page Catalog. Lists the contents of over 400 disks. \$8.50 plus \$2.00 S & H

- VDE - WordStar like Word Processor
- Reliance Mail List Program
- IMP Improved Modem Program
- Checks - Checkbook Program
- Touch Typing Tutor
- CHALK - beginning Spread Sheet Program

All of the above Public Domain programs come with printed manuals. Price \$15.00 each plus \$3.50 per order Shipping and Handling.

CP/M 2.2 MANUAL

Brand new CP/M 2.2 manual from Digital Research in 3" slip case (still sealed) with a registration card. Everything you wanted to know about CP/M and a lot more.

Price \$ 19.95 plus \$ 4.00 Shipping and Handling.

Commerical Software

BackGrounder - Task Swicher	74.95
BDS 'C' Language	89.95
Checks & Balances - Home Accounting	74.95
CPI Accounting - GL,AR, AP & PR per module	94.95
dBASE II - Database Program	150.00
MBASIC & CBASIC	59.00
Milestone - Project Planner	99.00
MITE - Communications Program	49.95
Nevada COBOL	39.00
Nevada Fortran	39.00
PC-FILE 80 - Database Program	49.95
SuperCalc - Super Spread Sheet	99.00
T/Maker - Integrated WP, S/Sheet, DB, etc.	120.00
TURBO Pascal	65.00
WordStar 2.26 - Word Processor	39.00
WordStar 4 - Word Processor	120.00
ZBASIC - Compiler BASIC	89.95
Z-System - Enhancement for 2.2 & 3.0	69.95
Z-80 Development System	49.95

Please add \$ 3.50 per order - Shipping & Handlg.



ELLIAM ASSOCIATES

ESTABLISHED 1975

Calif. Residence add 6% sales tax

Send orders to or call

Elliam Associates

P.O. Box 2664, Atascadero, CA 93423
(805) 466-8440

EAGLE COMPUTER USERS GROUP

The Eagle Computer Users Group is one of the few remaining support groups for users of Eagle computers, both the CP/M line and the later 1600 and PC models. Because Spellbinder was bundled with Eagle computers, ECUG is also a Spellbinder users group. Anyone who acquires an Eagle computer is urged to get in touch with ECUG, P.O. Box 3381, Saratoga CA 95070, phone (408) 972-1965.

Meeting place

ECUG meetings are held at Tandem Computers Incorporated, 10435 North Tantau Avenue, Cupertino. To get there, take 280 to the Wolfe Road exit; turn left at Vallco Parkway; turn left at Tantau; go over the bridge; and turn in where it says "Tandem Computers" on the left. There is no longer a guard at the lobby; ECUG members will check the front door at intervals to admit attendees. Be sure to sign in at the meeting.

Meetings are the second Saturday of every month, from 9 A.M. to Noon. The remaining 1991 meetings will occur on June 8, July 13, Aug. 10, Sep. 14, Oct. 12, Nov. 9, and Dec. 14.

May 11 meeting

Attendance was light at our May meeting. Not only was it the day before Mother's Day, but Tandem's annual Up-and-Running race was routed down Tantau Avenue right past the building in which we were meeting, and it was difficult to get to it. Nevertheless David McGlone, David Banoff, Kenneth Thomson, Dick Dethlefsen, Bob Vinisky, Jack Morse, Gene Chapin, and Dave Honkala managed to get in and hold the meeting.

Concrete evidence was presented that ECUG was not the only group that Tandem did not notify about a guard no longer being present at Building 200 on weekends. About thirty members of an SJSU class or club had to stand around outside until a phone call to security brought a guard over from another building to let them in.

Bob Vinisky talked about and demonstrated the features of LSH, a Z-System command-line history shell, with remarks to indicate which features were different or absent in EASE, HSH, and CLED. I, for one, learned a lot. Thanks, Bob.

June 8 meeting

9:00 Meeting begins.

9:30 Jerry Davis will discuss what's involved in installing Micro Solutions' MatchPoint and Compaticard IV floppy-disk controllers in a PC.

12:00 ECUG meetings ends.

ECUG library

The contents of the ECUG Library reside at the editor's house. Members may borrow them between one meeting and the next. Either call me evenings at (408) 293-5176 and ask me to bring them to a meeting, or phone to arrange a time to come over and borrow them.

README.DOC, Journal of the Orange Coast IBM PC User Group. April 1991 issue, Version 7.4. Donated by David Banoff.

ECUG software libraries

ECUG has two software librarians. Anyone seeking CP/M or Z-System software should contact Ken Thomson, 71 Rosenkranz Street, San Francisco CA 94110, phone (415) 648-7550. For PC (MS-DOS) software, our librarian is Jack Morse, 7390 Rainbow Drive, #1, Cupertino CA 95014, phone (408) 252-6103.

Please note that, as ECUG is no longer a corporation, the officers of the club do not wish to handle money (except those, like Shirley Welch and myself, who have businesses and do so as part of the business). Therefore, when you request disks from Ken or Jack, send them the floppy disks and the postage to mail them. They will copy the software you request onto your disks and mail them back to you. That

way no money changes hands, as would be the case if they continued to charge a fee per disk.

No PC software was received this month.

EAGLE HARDWARE BULLETIN

Eagle keyboard vs IBM keyboard

Editor's note: Back when Eagle Computer was still in business, they maintained a tech support bulletin board, on which were posted a number of Eagle hardware bulletins such as this one. I print them here as a service to ECUG members with Eagle 1600s and Eagle PCs. We will print the others in months to come, thanks to David Banoff, who gave them to me on disk.

The Eagle 1600 and PC keyboards offer several advantages over the IBM PC keyboard:

Number of keys: The Eagle PC has a 105-key keyboard while the IBM has only 83 keys on its keyboard.

Number of function keys: The Eagle PC has 24 programmable keys; the IBM PC has 12 programmable keys.

IBM Selectric compatible: The Eagle PC has the return and backspace keys in the same position as the IBM Selectric typewriter, whereas the IBM does not.

English command function: The Eagle PC has marked its function keys with the appropriate commands for ease of use. For example, an Eagle user wanting to search a document can strike the search key to perform this function. An IBM user must strike the F9 key to perform the same function. A user must refer to the User's Manual each time a task must be performed.

Ability to emulate IBM keyboard: The Eagle PC can emulate the num lock and scroll lock functions of the IBM keyboard. For example, Lotus 1-2-3 can be run with the Eagle PC by emulating the num lock and scroll lock functions of the IBM PC.

Easier Touch of keyboard: The Eagle PC has an easier touch, giving the user an advantage

in typing speed and productivity.

Many of the programs created for the IBM PC computer will also run on your Eagle 1600 or PC computer with little or no modifications. However, since there are several differences between the Eagle and the IBM keyboards, you may encounter situations where the program or its documentation refers to an IBM key that is not evident on the Eagle. To alleviate this and to assist you in operating your IBM programs, the following chart compares the special function keys. To use the chart, simply read down the IBM column until you locate the key or combination of keys in question, then look across to the Eagle side to find the correct Eagle equivalent. Keys separated by a plus sign + are pressed simultaneously.

EAGLE	IBM
[RETURN]	[ENTER]
[ESCAPE]	[ESC]
[DELETE]	[DEL]
[INSERT]	[INS]
[ALT]	[ALT]
[CONTROL]	[CTRL]
[F1]	[F1]
[F2]	[F2]
[F3]	[F3]
[F4]	[F4]
[F5]	[F5]
[F6]	[F6]
[F7]	[F7]
[F8]	[F8]
[F9]	[F9]
[F10]	[F10]
[F11]	[SHIFT] + [F1]
[F12]	[SHIFT] + [F2]
[F13]	[SHIFT] + [F3]
[F14]	[SHIFT] + [F4]
[F15]	[SHIFT] + [F5]
[F16]	[SHIFT] + [F6]

[F17]	[SHIFT] + [F7]
[F18]	[SHIFT] + [F8]
[F19]	[SHIFT] + [F9]
[F20]	[SHIFT] + [F10]
[F21]	[CTRL] + [F1]
[F22]	[CTRL] + [F2]
[F23]	[CTRL] + [F3]
[F24]	[CTRL] + [F4]
[CONTROL] + [F5]	[CTRL] + [F5]
[CONTROL] + [F6]	[CTRL] + [F6]
[CONTROL] + [F7]	[CTRL] + [F7]
[CONTROL] + [F8]	[CTRL] + [F8]
[CONTROL] + [F9]	[CTRL] + [F9]
[CONTROL] + [F10]	[CTRL] + [F10]
[ALT] + [F1]	[ALT] + [F1]
[ALT] + [F2]	[ALT] + [F2]
[ALT] + [F3]	[ALT] + [F3]
[ALT] + [F4]	[ALT] + [F4]
[ALT] + [F5]	[ALT] + [F5]
[ALT] + [F6]	[ALT] + [F6]
[ALT] + [F7]	[ALT] + [F7]
[ALT] + [F8]	[ALT] + [F8]
[ALT] + [F9]	[ALT] + [F9]
[ALT] + [F10]	[ALT] + [F10]
[ALPHA LOCK]	[CAPS LOCK]
[SCRL BREAK]	[SCROLL LOCK]
[TAB]	[⇔]
[SHIFT] + [*]	[SHIFT] + [PRTSC]

This asterisk key is located directly to the right of the calculator keypad.

The keypad on the IBM keyboard is designed to act as both a standard calculator keypad and as a means of moving the cursor around the screen. The keypad is switched from one function to the other with the [NUM LOCK] key.

The keypad on the Eagle can be made to function in exactly the same fashion by pressing the Eagle equivalent of [NUM LOCK]. When set to the calculator mode, the keys enter the numbers printed on their faces. When in the cursor movement mode, they

function as shown in Part 1. Shown in Part 2 are a series of special Eagle movement keys not found on the IBM that function without the need of the [NUM LOCK] equivalent. Thus, the Eagle offers you the choice of either using the keypad for both functions and switching back and forth, or leaving it in the calculator mode and moving the cursor with the special movement keys.

PART 1:

EAGLE	IBM
[CONTROL] + [ALT] + [N]	[NUM LOCK]
[1]	[END]
[2]	[←]
[3]	[PgDn]
[4]	[↓]
[5]	
[6]	[↑]
[7]	[HOME]
[8]	[→]
[9]	[PgUp]

PART 2:

EAGLE	IBM
[ALT] + [BACK SPACE]	[↓]
[→]	[→]
[←]	[←]
[↑]	[↑]
[SHIFT] + [↑]	[END]
[HOME]	[HOME]
[SHIFT] + [→]	[PgUp]
[SHIFT] + [←]	[PgDn]

You can also emulate the PAUSE function of the IBM keyboard (activated by pressing [ALT] + [NUMLOCK] on the IBM) by pressing [ALT] + [CONTROL] + [P].

Socrates

Z - N O D E 3 2

POST OFFICE BOX 12, SOUTH PLAINFIELD, NJ 07080

modem: (201) 754-9067 24 hours daily

Serving Today's CP/M and ZCPR Users

With the best in 8-bit computing

CP/M Support

You have been told that CP/M is dead. Don't you believe it. To paraphrase a famous quote, reports of our demise are premature. We haven't died; we have transformed!

Since the earliest days, CP/M has attracted the best in programming talent. Some sold their work commercially but many more donated their efforts to the public domain. Today, most commercial programmers have moved on. But the others remain, as active as ever. This presents you with an interesting dilemma: Most of the new programs are free or nearly so, but stores won't carry them! How can you get support?

There are four avenues of support for today's CP/M user:

- Remote Access Systems (BBS's)
- User Groups
- Mail Order Companies
- Magazines and Newsletters

REMOTE ACCESS SYSTEMS

Remote Access Systems (RAS or sometimes called Bulletin Boards), are computers set to automatically answer the telephone. You can send and receive messages, programs and files on a RAS. You need a modem and a communications program.

There are literally hundreds of systems that support CP/M. Most are free to the caller. *Socrates Z-Node 32* is such a system. Some of the most active are listed at the end of this paper. Find one that appeals to you and call. You should be able to find the North American listing of remote CP/M systems on any of these. Ask the sysop if you need help.

One problem with getting support by modem is the cost of the calls. Galaxy Starlink offers an "after hours" service through Tymnet that allows you to make modem calls to some 200 cities for as little as \$1.50 per hour plus \$10.00 per month. Call 1-505-881-6988 for more information.

USER GROUPS

User groups are excellent sources of help. A comprehensive list of groups is printed in the back of *The Computer Shopper* each month. Groups vary considerably based on their size and the relative experience of their members, so check out the group before you join. Since a group is only as good as its members, you should plan to take on some responsibilities. You will find your efforts amply repaid.

MAIL ORDER COMPANIES

There are many companies that continue to specialize in CP/M. *Chicago's First Osborne Group* publishes a file named CPMSRC-I.LZT which lists most such firms. You can get this list from your user group or on a RAS. Or write to CFUG at Box 1678, Chicago IL 60690. Enclose a couple dollars to pay expenses.

Special mention should be made of the *Z-System Software Update Service*. Users of ZCPR can subscribe for regular updates of the best Z-System programs. Special editions are also available. For example, there is a collection of some 360 command files in a set of six disks for just \$36, or over a full megabyte of help files for \$20. Send \$2 to *Sage Microsystems East*, 1435 Centre Street, Newton Centre MA 02159-2469 to receive a catalog disk. Be sure to tell them what format you have!

MAGAZINES AND NEWSLETTERS

If you want 300 slick pages of full page advertising and press releases disguised as product reviews, you will be disappointed. Our publications reflect the tradition of user involvement in CP/M. You may even find yourself asked to write an article on something you've learned! Some publications to look at:

THE COMPUTER JOURNAL is the top-of-the-line in CP/M periodicals. Topics range from tutorials to advanced work in the operating system. Heavy emphasis on Z-System. Published six times a year. Subscriptions are \$18 per year. *The Computer Journal*, 190 Sullivan Crossroads, Columbia Falls MT 59912.

PIECES OF EIGHT is published by the *Connecticut CP/M Users Group*. Great degree of user involvement reflects the diverse interests of the members. Recent articles include making Basic 'Z-Smart', and installing a RAM disk. Membership including newsletter is \$15 per year. *CCP/M*, c/o Tom Veile, 26 Slater Avenue, Norwich CT 06360.

SLKUG NEWS is the newsletter of the *St. Louis Kaypro Users Group*. Don't let their name fool you — they support all kinds of CP/M machines. Articles include reviews of new products, offers for group purchases and tips on using applications. Membership including newsletter is \$18 per year. *SLKUG News*, 5095 Waterman Avenue, St. Louis MO 63108.

By the way, we are compiling a listing of all CP/M newsletters and periodicals. If you know of one that you feel is of value, let us know! A sample copy would be most appreciated.

CAN WE HELP?

One last avenue of support is *Socrates Z-Node 32*. We will copy any files you want from our system or extensive off-line archives for \$6 per disk (\$10 for foreign orders). Send \$2 for a listing of available files. Please make checks out to Chris McEwen, Sysop, and tell us what format you need. Proceeds go to the support of Z-Node 32.

Chris McEwen

Try one of these Remote Access Systems for support via modem:

Z-Node Central, Los Angeles	CA 213-670-9465	Z-Node 36, Pasadena	CA 818-799-1632
Z-Node 3, Newton Centre	MA 617-965-7259	Z-Node 45, Houston	TX 713-937-8886
Z-Node 4, Salem	OR 503-370-7655	Z-Node 50, Alice Springs, N.T.	
Z-Node 5, Montreal QC CANADA	514-324-9031	AUSTRALIA 5750	61-089-528-852
Z-Node 6, Drexel Hill	PA 215-623-4040	Z-Node 58, Oklahoma City	OK 405-943-8638
Z-Node 9, San Diego	CA 619-270-3148	Z-Node 62, Perth,	
Z-Node 10, Mill Creek	WA 206-481-1371	Western AUSTRALIA	61-9-450-0200
Z-Node 11, Chicago	IL 312-764-5162	Z-Node 65, Cheyenne	WY 307-638-1917
Z-Node 12, Newington	CT 203-665-1100	Z-Node 66, Costa Mesa	CA 714-546-5407
Z-Node 15, Manhattan	NY 212-489-7370	Z-Node 73, Ballwin	MO 314-821-1078
Z-Node 20, Burnaby, BC CANADA	604-299-0935	Z-Node 77, Austin	TX 512-444-8691
Z-Node 21, S Plainfield	NJ 201-757-1491	Z-Node 78, Olympia	WA 206-943-4842
Z-Node 32, S Plainfield	NJ 201-754-9067	Z-Node 81, Lancaster	CA 805/949-6404
Z-Node 33, Enid	OK 405-237-9282		