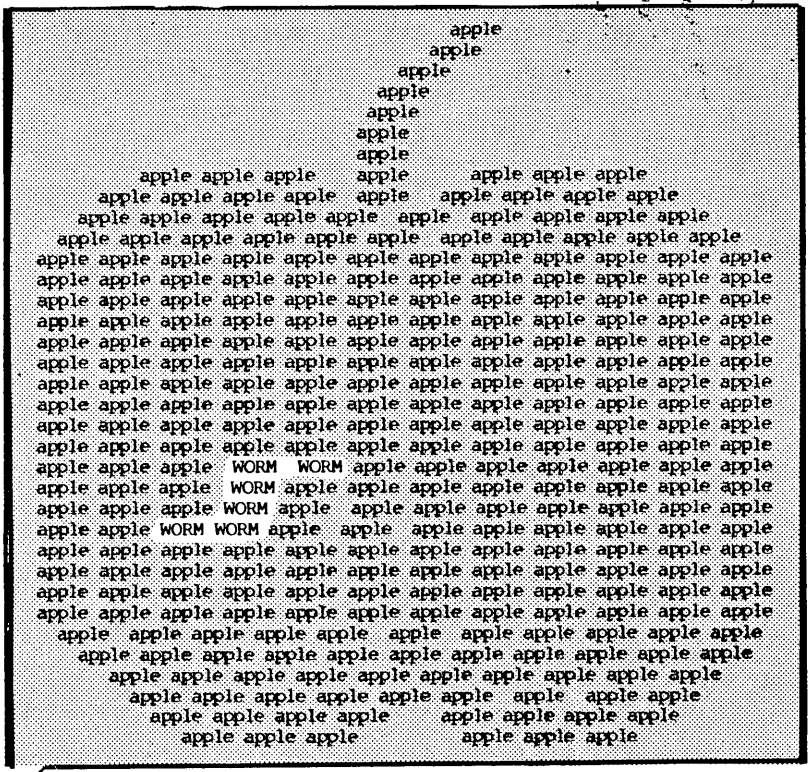
BAMDUA BAKUP News

Bay Area Micro Decision Users Association • Bay Area Kaypro Users and Programmers


apple
apple
apple
apple
apple
apple
apple
apple apple apple apple apple apple apple
apple apple apple apple apple apple apple apple apple
apple apple apple apple apple apple apple apple apple apple
apple apple apple apple apple apple apple apple apple apple apple
apple apple apple apple apple apple apple apple apple apple apple apple
apple apple apple apple apple apple apple apple apple apple apple apple
apple

BAMDUA-BAKUP NEWS is published bimonthly by **Bay Area Micro Decision Users Assn.** in association with **Bay Area Kaypro Users & Programmers,** both Calif. non-profit corps., for their members. Membership fee to join or renew is $20/yr. Both organizations were founded in the early 1980's to support users of Morrow MD and Kaypro computers. Both now include many users of other systems including MS-DOS. Both maintain libraries of CP/M public domain software.

Morrow Owners Review (MOR) provides some material for publication, purchases and separately mails copies of each issue to its members. Members of all three groups use the **MOR BBS** at 415-654-3798.

OFFICERS

President: Peter Campbell
Treasurer: Ron Jacobs
Secretary (BAMDUA): George Borys
Secretary (BAKUP): Jim Ullrey
Library: Gene Korte
At Large (BAMDUA): Sypko Andreae, Georgia Babladelis
At Large (BAKUP): Bob Athey, Dwight Chew

Newsletter:
Peter Campbell (BAMDUA),
Bob Athey (BAKUP)
Sypko Andreae (MOR)

Mailing Addresses:
BAMDUA: P.O. Box 5152, Berkeley, CA 94705
BAKUP: P.O. Box 8537, Berkeley, CA 94620
MOR: P.O. Box 5487, Berkeley, CA 94705

Current circulation: 400

MOR, P.O. Box 5487, Berkeley, CA 94705; (415) 658-0152

There was a time once when I was so busy organizing the Morrow users communities -- first the local club, BAMDUA, then the Morrow Owners' Review -- that I never had much time to play with all the software goodies that came my way. Although I was always involved in getting the public domain stuff together, first for the BAMDUA library, then for the MOR FLOBs, I never had time to play with it myself. I remember feeling a bit jealous of all my Morrow companions who seemed to have so much fun with all the new software, while I edited, phoned, wrote, and struggled with the finances and organization.

Now times are different. MOR doesn't take so much of my time anymore. One of my main chores is maintaining the MOR/BAMDUA BBS and I still I find I can't devote as much time to it as I'd like. Once in a while I get up real early, while everybody else is hopefully still asleep, and I rove around a few of the local big CP/M BBSs, hunting for new stuff. During the day on weekends it's all but impossible to get on these boards. Mike Allen is such a hunter of new PD software. I admire him for finding the real good stuff and writing about it in his TYROS column. An I admire Ilbert Butler for getting existing stuff in proper shape (IMP245), and George Borys for creating all those new goodies all the time. Among them I feel like a tourist.

BAMDUA/BAKUP GENERAL MEMBERSHIP MEETINGS DISCONTINUED

So far the regular monthly members meeting has been held every third Tuesday of the month at 7:30 PM in the Albany Senior Citizens Center, 846 Masonic Avenue, Albany, CA.

The last regular members meeting will be in May 1989. It will be a special event well worth attending: The 6th Birthday Party for BAMDUA!! Maybe there will be an appearance by a famous person. Be ready for a surprise.

After this meeting there will be no regular meetings, but only irregular ones, which will be announced by postcard about 3 weeks ahead of time. They may be happening every three months, or more often, depending on many factors. In any case you will be notified.

The BAMDUA/BAKUP Board will still meet regularly every 2nd Tuesday of the month, same place, same time. Members are welcome to attend.

APRIL MEETING NOTE

Banking by home computer will be the subject of our guest speaker Garrett Kirkstone, a vice president of the Bank of America. As usual the meeting will be at 7:30 p.m. on the third Tuesday of the month, April 18, at the Albany Senior Center, 846 Masonic Ave., just north of Solano Ave. in El Cerrito. Everyone is welcome.

The HANDY personal time management system: As a sideline of this tourism of mine I've decided to give you my personal experience of some of the public domain software I have encountered. That is how my war story on SAPP and FIRE came about. This time I'll tell you something about HANDY, the "Personal Time Management System," written in 1982 by Peter C. Hawxhurst. It's a neat little program and it works quite well within its limitations; it also has a nice documentation file. I have used it for a week now and I am pleased. It is actually useful.

HANDY comes wrapped in a file called HANDY201.LBR. The actual program is HANDYSYS.COM which uses no less than 9 overlay files (.OVL) for the 9 different functions it performs. HANDY uses the cursor commands of the terminal to give you a nice display, so you will have to adjust HANDYTRM.PRM to match it to the cursor commands of your particular terminal. I modified mine for the MT-70 terminal; it should work on the MDT-60 terminal as well. If you have an MT-70 and you want to try this program then call me on the BBS and I'll get you the MT-70 version of the HANDYTRM.PRM file. Lastly, all the files in HANDY201.LBR are crunched, so you have to uncrunch them: unpack with LT, or unpack with NULU and then use UNCR to uncrunch whatever crunched files there are (UNCR *.*).

What it can do: HANDY is largely self-guiding with its menus, but it helps to have the HANDY201.DOC handy to see what some of the more cryptic command choices stand for. Applications included in the HANDY system are:

- A personal appointment calendar for the display and maintenance in halfhour increments of daily appointments.
- A personal TO DO LIST for the display and maintenance of nonscheduled activities on a daily basis.
- A reminder list of important dates/events to remember on a monthly basis.
- A planning calendar for blocking out and displaying large portions of scheduled time.
- A calendar program for displaying monthly calendars a quarter of a year at a time.
- An action planning notebook for displaying and maintaining future activity records by categories.
- A notebook or log for the maintaining and displaying of miscellaneous notes, records or whatever by category.
- A quick reference phone number listing.
- A decision support routine for rational decision making assistance.

Each time-management program needs to know the time or at least the date. In HANDY's case you don't need a clock in your

COVER ART NOTE

This is an adaptation (with translation) of a cartoon (?) I saw in a German magazine some years ago. I put it together in Wordstar, and have used the EXACT scientific notation software to do some modification on it for other magazine(s).

--Bob Athey

computer because the first thing it asks for is the date which you then enter in the form: 03/31/1989 for March 31st, 1989. HANDY then goes ahead and creates a whole bunch of data files, one for each of the nine applications. The nice thing about the data files is that they are in ASCII, or readable (and editable) text. This makes it possible, in some cases, to circumnavigate the most inconvenient limitations of this system.

Error recovery works quite well. If you type something incorrectly, it quickly invites you to enter it in again. If you have navigated on the wrong path among the menus, you can nearly always, with some exceptions, bail yourself out by hitting ESC(ape).

Displays and Wrist-Slaps: The displays are nice -- neat and readable -- but there are a few bugs. For instance, in the Planning Calendar, where you can block out periods of days, if you start a period on a Sunday the resulting display (showing the blocked out days) is always off. (This program originated from an OSBORNE computer; maybe the program was never checked out on an 80 char. wide screen?) Just remember never to start a period on Sunday and you'll be fine. Other display bugs appear where a string is a bit longer than the author of this program had foreseen; when the string is cleared the "illegal" excess stays on the next screen confusing later displays. To clear the screen, go back to the main menu.

Another annoyance is that if you enter a string in the daily appointment calendar you can easily enter more characters than is allowed. Even though HANDY warns you ahead of time that you only have 28 characters for each appointment, it is really hard to eyeball whether you are past the limit or not. BBSs faced the same limited string length problem; there it has long since been resolved by displaying handsome little markers that clearly show how much space you have while you enter the string. Not so with HANDY: if you enter too many characters and you get a wrist-slapping error message and then you are punished by having to re-enter the entire string. Worse, if you want to delete a "Reminder" line (max 49 characters) HANDY makes you re-enter the complete reminder line before it allows deletion! You have to enter all the characters of the reminder string; if you are off by at least one, another wrist-slapping message. I find this sort of torture intolerable, although it might have been acceptable in 1982.

I soon found a way around these inconveniences: process the data files with NW or WS in non-document mode! I find myself doing this all the time now. Rather than retyping an appointment line with a spelling error in it, I go to NW and edit the APPOINTS.DAT file. I do the same with most of the other .DAT files (DOTHINGS.DAT, REMINDER.DAT and ACTPLANS.DAT). To make it easier to navigate with NW through my very long ACTPLANS.DAT (action plans file) I first sort it by date with the program SORTV: SORTV ACTIONPL.DAT <CR>.

Dates can kill you. When HANDY first comes up it asks for the date which it will store away in the APPOINTS.DAT file which mainly contains all the appointments for each day. It will then wipe out all appointments, things to do, and reminders with dates prior to the one you entered. In other words, it clears

out the complete past. HANDY is a forward looking program. If you were to type in a date, say, a year in the future, you risk wiping out all your appointments, things to do, and reminders for months to come! Before it wipes your slate clean for you it asks, politely, whether it should print out all the lists that it is about to delete. At that point check the date you entered, and if it is wrong then bail out with a deft ^C (CTRL/C). This will bring you back to the CP/M prompt and you can start all over again.

As the author himself warns you: Always back up your .DAT (data) files. Then if you loose all your data by entering a wrong (future) date, you can reload the backed-up .DAT files with NSweep and everything is again as before.

Every time you start HANDY you have to enter the date. If you run HANDY more than once a day -- as anybody probably would -- this becomes tedious. Why HANDY doesn't use the date previously entered and stored in APPOINTS.DAT is a mystery. It would be so much nicer if it would display what it thinks today's date is while asking you to confirm. Then if the displayed date is not correct you could enter the proper date. The way I work around this is by using KEY.COM. One key is assigned to say HANDYSYS<CR> so it will start HANDY for me, and the next key contains the date which I have previously edited in my KEY file as my first act of the day. (KEY.COM is a SMARTKEY-like program that runs on MD-HD and late model MD3 computers, but it only can use the function keys and the keypad keys if you enable the keypad.) Modern-day HANDY-like programs all run in the background, while you run your applications, so you don't have to start them up every time you want to consult them; they also use the built-in clock/calendar of the computer.

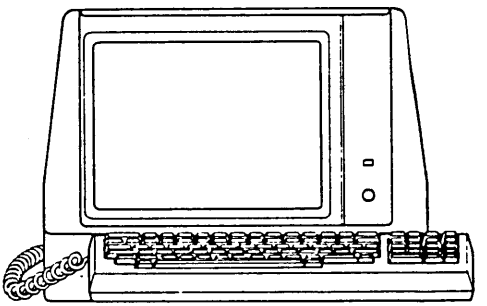
Daily Appointments and Things To Do Today: The appointment calendar screen shows one day at the time. You can make an entry on each appointment line which is indicated with a start time, one line for every half hour. It is oriented towards the workday, no evening hours are available! This doesn't work for me since I do have a lot of evening engagements. But there is no way around this one; the calendar day starts at 7:00 a.m. and ends at 6:00 p.m. and that's it. So I put my evening engagements in DOTHINGS.DAT (things-to-do-today file), adding the times myself. Not so neat. What I like about the appointment calendar is the easy way you can flip forward through the days. You can't flip backwards but you can specify a specific date, either in the future or in the past.

The things-to-do-today feature (main menu #2) allows you to make a TO DO LIST for the current day and any day in the future. Unlike the appointment calendar, you can't flip forward through the days and I miss that feature. There is also no easy way to carry forward to the next day things left undone today, a feature I need a lot! I use NW again to work around this one by globally changing today's date to tomorrow's date for the leftover items of the day. The last thing I should do before finishing up for the day.

Calendar Displays and Action Plans: The calendar program for displaying monthly calendars is fun to watch. You get four calendars displayed at the time; the second one is the current

MORROW

Our Job is to Keep You Working!



- REPAIRS
- CONSULTING
- SALES
- UPGRADES
- MICRO DECISION
- TERMINALS
- DECISION 1
- S100

Ship to us via UPS

JACOBS
computer services

(415) 525-9425
Call for information

CUT AND TAPE TO YOUR COMPUTER

month. Then you can change to any future or past month to be displayed as the second one on your screen. It goes back to about the year 1000 and forward to somewhere around year 2900. There is something fascinating about looking at dates far in the future and knowing you won't be around then. You can't enter data with menu; it's for display only.

The feature I find myself using most is ACTION PLANS. You can create up to 24 categories that each can be filled with up to 12 actions. Oddly enough, you can actually A(dd) more actions per category but the ones above twelve are not displayed. Instead, they appear as you delete a few of the visible actions. I learned that my intentions about what I want to do far exceed what I actually can do. My ACTIONPL.DAT file is unrealistically long indeed. It's a good thing there are no dates associated with these entries because that would be an overwhelming sight! Now I can just regard my long list of planned actions and put it off to the future at will. Not so with the things-to-do-today feature where I can keep moving undone things forward in time by changing the date fields with NW to tomorrow's date.

An admirable effort for 1982: With all its shortcomings and its few display bugs I like this program to the point that I felt tempted to get the source code (it's compiled BASIC) and rewrite it with up-to-date entry features and a few extended features, maybe doing it all in C. But I know better than to plan anything like this before retirement. Sure, HANDY has severe limitations, but you have to remember that this is an old program, last updated in 1982. In the meantime we have been spoiled and now take for granted all kinds of conveniences that were unheard of in 1982. For its time I would say that this is an outstanding public domain program, written and documented by an obviously well organized person with a clear head who was able to focus on what he most needed, leaving unimportant details out. An admirable effort. For a 1982 program this is really quite good and I heartily recommend you give it a try.

WHERE'D EVERYBODY GO?

654-3798

MOR BBS

MORROW MICRO DECISION SYSTEM FOR SALE

Morrow MD-3 (ROM rev. 3.1) CPU with 2 DS drives. Televideo 920C Terminal and MP1000 LQ Daisy Wheel printer. Software includes CP/M 2.2, NewWord, LogiCalc (a spreadsheet) and MBasic. BEST OFFER! Call John Larson at 415-845-3700 or 273-6413

[Bruce Gowens is a Personal Pearl Po-Bah, Computer Driver and Software Grease Monkey who lurks on the MOR RBBS (415/654-3798) and eats mail at P.O. Box 8068, Emeryville, CA 94608. His cat likes Emeryville because she wants to be near the seat of real power. Copyright 1989 Bruce M. Gowens]

PRETTY FUNNY? Those of you who read my column in the last issue of MOR-Atlanta (I can never figure out the schedule of BAMDUA, they're too regular) now know a lot about me that I generally don't care to reveal (the girls, wild parties, strange substances, what I **REALLY** do with my computer, etc.). So since I let my hair down (look at picture), I might as well reveal my deepest and most innermost secret.

I'm tired of "Getting Personal With Pearl." Sure, it was a cute column header back in the good old MOR days, but every welcome mat wears out. So I hereby rename this column to simply "Getting Personal." (Just don't ask me for the girl's phone numbers, that's **too** personal.) That means that I'll deal with Pearl every so often (she's turned into a grand old lady, and lost none of her beauty), but I'm going to talk about other things as well. I figure that all the people still using Personal Pearl have worked it out by now, and the others aren't interested. And if you are in a crunch, contact me by the means that should be printed above.

Good. We now understand each other. Maybe. Readers of the female persuasion might presume from the above that I am some kind of sexist. I certainly am, but I venomously deny that I am whatever kind they may think, other than latent. All this is introduction to the following.

I've picked up some text files from RBBS's purporting to be humor. The thing currently going the rounds is filename COMPUTER.SEX. It's clever enough, and I did get a laugh from some of the manipulations of our jargon--mainly because the puns illustrated how ridiculous our jargon can be. On the other hand it reached, and kept reaching, for the quick-and-dirty joke. This kind of smut was old to me when I was eight years old (as I remember, the seven-year-olds liked it). Kind of like a fraternity party in a fraternity where all members are from the sticks. And it treats woman as an object, which I find objectionable.

So much for the slam, now for the goodie. BEER.TXT, which I'm uploading to the MOR/BAMDUA board, is funny. It compares, from a male viewpoint, Beer with Women on 52 points. Sexist? I suppose so, but so what. I'm sure that some woman can reverse the ideas and list 52 reasons why Beer is better than Men, and I am sure I'll enjoy that list also. Please, please, whoever you are, send me the female response (funny, please, no polemics).

Unlike COMPUTER.SEX, which is an adolescent bit of clever smut, BEER.TXT gets its humor from normal male frustrations. And it knew when to quit. 52 is quite enough, thank you.

I mentioned jargon, above, and that is of interest to me. We have all these funny words: CPU, byte, bit, word (meaning two bytes), hack, CP/M, VAX, DOS, tweak, CH, RCH (last two inherited from engineering, meaning a relativity small distance and a relatively very small distance, and I won't explain the letters),

CMOS, call, overlay, ROM, RAM, chain, on-line, RBBS, software, data base (base?), firmware, hardware, interface, and on and on and on. Get into the MS-DOS world (MS-DOS?) and the list gets even longer (EGA, VGA ...).

Hey, can't these people speak English?

Well... not really. When we deal with computers, we are talking about something that has never before existed in the history of humanity. A machine that doesn't have a purpose, but can fulfill so many purposes. Strange concepts that deal with logical abstractions as if they were solid objects. Ideas made real, such as spreadsheets and data bases, without physical reality--yet nevertheless real. Since such things have never existed in the world before, we have to give them names--usually new--(the only alternative is to call them Ghosts) and that is one function of the jargon we use.

Sometimes we have to invent a new name simply because what we are doing is done so differently than in the past. Thus that amazing meaningless jargon phrase "word processing." You know it is not the same as typing. Just as writing is not inscribing letters with a quill pen, typewriter, or linotype. A writer is an author (as in Author! Author!) and a typist or secretary is not. There used to be a word, "transcriber," for someone who typed what was on an Dictaphone band. Well, word processing is something else, and what's more it is something else to a writer or secretary. It takes one to know one, but I know what the phrase means, I think. I'm waiting for the first paperback by "that famous word processor, Z. Zzzzz." But when all typists are "word processors" the distinction, and the phrase, will disappear. Just as you don't hear "transcriber" much any more (remember when "takes dictation" was a usual job description?).

What we are doing, of course, does have an older name. But it's pretty general, and doesn't specify the method. That old name is "Magic." We look into our crystal screens and see... And who says the old wizards didn't have to learn how to do it?

So there is a reason for some of our jargon. But there is another reason, and I recommend that we always be aware of it so that we can avoid it. I'm going to call it the "High Priest" reason, and after that I shouldn't have to explain (but will). It's not jargon (a specialized vocabulary for a trade) but is cant (a specialized vocabulary of a minority group). Webster's offers some other distinctions, but these will do for this discussion.

Cant has two social purposes. One is to identify outsiders. When I was a beat (**not** Beatnik) in San Francisco the way someone said "man" (as in "Hey, man") told me immediately whether he was one of "us" or one of "them." "Dig" was another such word. When too many outsiders learned the code, it changed. (I remember two young well-dressed drunks at a hole-in-wall place in NY, one saying "You want Orange Juice?" and the other replying "Far Out, Man!" The phrase had passed into the hands of outsiders, and obviously become meaningless.) This use of cant is typical of all minorities trying to find an identity, as well as criminal groups. It is also true of the very old rich.

The other purpose of cant is to make the speaker/author look good, smarter, wiser, or whatever. The "High Priest" bit. Academics do this a lot (always?), as do all the pseudo-execu-

tives in large corporations (spear-bearers, aka "yes-men"). I'd love to give examples, but might be sued if I did. Just listen to the CEO of that leading "alternative" computer company that seems to be suing others.

OK, we all want to make ourselves seem smarter and wiser. BUT IT HAS GONE TOO FAR! We are not a high priesthood with sacred language and symbols. We are just people getting by, and have found a nice little tool that helps us do so. We've learned the underlying truth of Magic: IGNORE THAT MAN BEHIND THE CURTAIN! We know that we are all slaves to the whims of the toolmakers (software writers) and like any other consumer have to beg, argue, fight, boycott, and whatever to get what we need and want from those alchemists of logic. So please, when you talk to your friends, the ones that don't want to know about computers, cut the jargon when you can; and always cut the cant.

And there is one other thing we can do: learn the magic. Twenty years ago you had to be some kind of weird teen-age genius to create the simplest of programs. Today anyone with moderate intelligence (and I know my readers are higher than that) can pick up enough to get their special jobs done using Turbo Pascal or Modula-2. All it takes is the ability to count, and sequential thinking abilities (this, then this, then this...). And once you see your first program on the screen, you are a real "High Priest" (just don't let it go to your head).

Nothing about Pearl this issue, except how did that reader track me down in my favorite bar? Well, I guess he was getting personal. "IGNORE THAT MAN..." is a quote from Frank Baum; who knew about computers before they were invented, Dorothy. And smut is in the eye of the beholder.

Next issue: How bondage changed my life!

BAMDUA-BAKUP News AD RATES

DISPLAY ADS

1/12 of a page (Business Card size)	$5.00
1/4 of a page	$25.00
1/2 of a page	$45.00
Full page	$85.00

Call for mechanical requirements.

CLASSIFIED ADS

Members: Free to members. 20-word maximum, normal rates apply thereafter.

Non-Members: 25 cents per word, with a $3.00 minimum. Please enclose your name, address, and phone number with your payment.

FREQUENCY DISCOUNT

2 months	10%
4 months	20%
6 months	35%

DEADLINE

Ad copy and payment must be received no later than the 15th of the month preceding publication. Send your check and ad copy to:

BAMDUA-BAKUP News
P.O. Box 5152
Berkeley, CA 94705
(415) 658-0152

Current Circulation:
400

P.O. Box 782, Ithaca, NY 14851; (607) 273-2132

Mail is starting to trickle in, perhaps because the holidays are over. We even have some answers this month.

Ben Cohen in Chicago offers some further help on using SmartKey with WordStar 4. The amount of memory used by SK has nothing to do with the number or length of your definitions; the amount of memory set aside for the definition file is fixed, and you can change it with SKPATCH. Meanwhile, you can use WSCHANGE to reconfigure WS4 to use less memory: from the WSCHANGE opening menu, select "C Computer," and from the next menu, "C Memory Usage." Mr. Cohen suggests: "Try setting WS4 to minimum memory allocations and see if it works that way. If it does, add to the undelete buffer until it doesn't and then back off again until it will. [This will tell you] the amount of memory you can allocate to WS4 features and still run with SK. Then you can allocate memory among the various features, such as the undelete buffer, number of dot commands allowed, header/footer size, etc. One candidate is WS shorthand; after all, if you have SmartKey, how much do you need to allocate to shorthand?" (As I pointed out last time, by using both you can get double use of the numeric keypad.)

Mr. Cohen also points out that people with Hard Disk MDs and CP/M+ have a lot more free memory and shouldn't have any trouble with memory resident programs like SK.

Bill Harrell in Utica, NY is using a public domain Z80 emulator called Z80mu to run Personal Pearl on an IBM-compatible Zenith 200. A friend loaded Pearl onto a CP/M partition on his hard disk using Uniform. Pearl runs fine, he says, but won't access his old data files on floppies. My first guess is that the emulator is set to read CP/M floppies in some format other than Morrow, so he must again use Uniform to copy the Morrow-format files onto the HD. However, when he enters a form name into Pearl, he gets an error message: "Unsupported BDOS function #1FH called from 6457H." BDOS (Basic Disk Operating System) is part of CP/M. Most CP/M software calls routines from BDOS to perform routine tasks like sending characters to the terminal or looking up files in a disk directory. From here on I'm guessing; I hope one of the CP/M gurus out there will write with clarification or correction. It appears that Pearl is calling a function that isn't supported by this emulator. The function at 1F is "Return physical disk information." Could there be be a patch to Pearl that would solve the problem?

If you want to try Z80mu, it was written by Joan Riff and is distributed by Computerwise Consulting Services, P.O. Box 813, McLean, VA 22101. They say, "No copyright notice, no financial contribution, no warrantee." You may find it on a PC-oriented BBS.

Here's another approach to running CP/M software on an IBM or compatible PC: someone sent me an ad for a product called UniDOS, a Z-80 coprocessor card that plugs into a half-size ac-

cessory slot on a PC. According to the ad, the card includes a Z-80 chip running at 8 Mhz, about twice as fast as the clock speed on a Morrow. This doesn't necessarily mean it will run twice as fast in practice, since everything also has to run in and out through the PC's CPU; disk calls might also be slower. As noted above, not all "emulators" emulate perfectly. If anyone out there has tried this product, send in a review. UniDOS is available from Central Computer Products, 330 Central Ave., Fillmore, CA 93015, 800-456-4123. The price listed is $175, including a free copy of the PC version of Uniform.

It should be noted that these and any other CP/M emulators are still limited to using 64K of memory, no matter how much you have on your PC. There still ain't no free lunch.

And finally back to "real" CP/M. **Gary Rosenshield** in Madison, WI has an MD-16 running Z3PLUS, which will only allow 16 directories per logical drive. He'd like to know if there is a way to partition his disk into two drives, allowing twice as many directories. I know this can be done under MS-DOS. Anyone know how to do it under Z3PLUS or CP/M 3.0?

Tools for Tyros

by Mike Allen

301 North Roadrunner Parkway, #109, Las Cruces, NM 88001

Most of us probably bought our computers for the things that they could already do such as word processing or data base management. But sooner or later there comes a time when every computer owner wants to write a program. It may be because we are just curious or want to learn something new or we want to do something with our computer that the programs we have don't do.

Programming for the first time can be an intimidating experience. There are lots of decisions to make before we even start to program. The first thing we find is that we have to decide which language to use.

What is a computer language? Very broadly it is a method for making a set of instructions that a human can give a computer to accomplish some task. Again very broadly, languages fall into two categories, assembly and higher level.

Assembly languages are processor specific. They are a set of instructions given directly to the processor in the machine being used. The microprocessor in the Morrow machines is a Zilog Z80. An important thing to remember is that the "instruction set" for a Z80 is a "super set" of the Intel 8080. This means that a Z80 can execute programs written for an 8080. The converse is not true. The Z80 has many more and more powerful instructions than does the 8080.

Assembly language programs are normally written using some sort of an editor. The file created by the editor is then fed to a special program called an assembler which, after one or more steps, creates a program that the computer can execute.

All Morrows came with two editors which will work for creating programs for the assembler. The first editor came with CP/M itself from Digital Research. It is called ED.COM. It is a "line editor" and is rather primitive and cumbersome to use.

Since either WordStar or NewWord came with our Morrows and these are much better editors than ED, I suggest that they be used, in the non-document mode, to create your "source" files for assembly language programs. I tend to use the public domain editor VDE since it is quicker and smaller than either WordStar or NewWord but has all the features needed for assembly language programming.

The Morrows also came with an assembler. For the floppy based MDs running CP/M 2.2 there is ASM.COM. The MDHDs using CP/M+ came with MAC.COM and RMAC.COM. These assemblers don't create .COM files directly but intermediates files of either the and LINK.COM does the same thing for a .REL file.

These assemblers were designed for the 8080 microprocessor and therefore don't take full advantage of the Z80. There are assemblers available for the Z80 from MicroSoft and SLR. The SLR assembler is getting rave reviews from its users as being fast, versatile and inexpensive as well as being well supported by SLR. The same cannot be said of Macro-80 from MicroSoft. For a long time, however, M80 was the only game in town. There is a lot of PD software around written for M80. M80 can occasionally be found at bargain prices. I got my version from a HeathKit store for $18 when they were liquidating their stock. There are also some PD Z80 assemblers around, but I have had only limited experience with them and so hesitate to comment.

I enjoy assembly language programming. It gets into the "guts and feathers" of the machine. I get a real appreciation of what goes into making my Morrow work and I also learn a lot about interfacing with CP/M. There is a real feeling of accomplishment in getting an assembly language program running. As a rule, assembly language programs execute faster than higher level language programs.

Higher level languages are designed more for the programmer than the machine. The goal of higher level languages is to generate programs that are "machine independent." That means that a program written in a higher level language could be used on many different machines using different operating systems. Like most design goals this hasn't really been reached, but it is surprisingly close.

Higher level languages can be broken into two general categories, interpreters and compilers. Interpreters take the source code and interpret and execute it instruction by instruction. Compilers are much like assemblers. They take the source code and generate a machine executable program. Both systems have advantages and disadvantages. Interpreters are easier to troubleshoot and can execute as you compose your program. They are much slower than compilers and are limited in memory available since both the interpreter and the source code have to be resident. Compiled languages, after the program is written and debugged, are faster and use less memory.

There are many different higher level languages available for Morrows. I'll comment on a few of the more popular ones: BASIC, PILOT, Pascal and C. It should be noted that any given language is not inherently interpreted or compiled. This is dependent on the particular implementation. BASIC, for example, can be either interpreted or compiled.

Most Morrow machines came with two interpreters, MicroSoft Basic (MBASIC.COM) and Morrow Pilot (PILOT.COM). Some also came with BaZic which I have not used enough to comment upon.

MBASIC is probably the most popular and widespread language in the CP/M community. Basic was designed as a language that would be easy to use, supplying most of what's needed to write useful programs. MBASIC certainly meets those goals. The documentation is fair. If you never want to write very complicated or fast programs, MBASIC is for you. However if you are serious about learning programming I suggest that you never mess with Basic and go directly to Pascal or C. If you get too comfortable in Basic you will have too much to unlearn when you progress to those languages.

Morrow Pilot is a surprising language that probably has a lot of applications that have never been explored. The PILOT language was designed as a programmed learning language. Its math capabilities are limited at best but it can really handle words. Morrow designed their Pilot primarily for their menu system. The Morrow Menu system is indeed a Pilot program. Take a peek at the .PIL files to see what a Pilot program looks like. [PILOT manual available from MOR, PO Box 5487, Berkeley, CA 94705, $12.00 post paid. Ed.]

If you're serious about programming, I suggest that you try either Pascal or C. There are several versions of each available (or at least there were at one time) for CP/M.

Following MBASIC probably the most popular language for CP/M is Turbo Pascal from Borland. When Borland first came out with their Turbo Pascal they not only made an impact with its speed and ease of use but shattered the price structure in the market place and put such firms as MicroSoft on notice.

Turbo Pascal is not just a compiler, but an entire system for program development. When you invoke Turbo Pascal you don't execute the compiler, but call up an editor for creation of your program. You can either start creating your program or call up a program source file that you already have on disk. You may then continue work on your source code or compile the source.

One of the hardest parts of getting a compiled program working is debugging. Turbo Pascal took some of the pain out of that process. When the compiler finds an error it throws you back into the editor and shows you the line of source code that the compiler choked on. This makes debugging much easier. The bad news is that Borland no longer markets Turbo Pascal for CP/M. I have heard that Alpha (present marketers of the Z System) is negotiating with Borland to handle Turbo Pascal for CP/M. Turbo Pascal can occasionally be found at swapfests and liquidations. The documentation is very good as far as the Borland implementation of Pascal goes, but other references are needed for Pascal itself.

There used to be another inexpensive Pascal on the market known as JRT Pascal. The program was released to Public Domain when the company went under. One reason the company went under was that JRT Pascal was extremely "buggy." Even though the price is right (free) I recommend that you stay away from it.

Pascal was originally designed by Nikolas Wirth as a tool to teach programming. It gained so much popularity that it soon

became widespread as an actual language. Turbo Pascal is actually "non-standard" but its widespread acceptance has made it a de facto standard.

We now come to my favorite language, C. To me C has many advantages not the least of which is that there is one accepted standard. That standard is the book The C Programming Language by Kernighan and Ritchie. If you are going to learn C, invest the twenty-some-odd bucks for that book.

C assumes that the programmer knows what he is doing and therefore lets him do it. This can be very frustrating when learning C since you might do something like tell your program to clobber the operating system and it will do so, but this approach gives C its flexibility. You can do things in C that are almost impossible in any other language. C source code is very terse and can be very difficult to read. In almost any language it is important that you "comment" your source code. This means making little comments in your program so that you, and other readers of your code, can figure out what you're doing. This is especially true of C and assembly programming.

The best buy for C compilers is MIX C. I believe it is still available and for about $25. You can add their editor to the compiler for a total package price of $40. It is not the best C compiler on the market by any standard. Others are faster and generate more compact machine code. But the MIX C seems to adhere closely to the K&R standard; it works and is much cheaper than the others. The documentation is outstanding. You can almost (but not quite) do without the K&R book.

There are other languages that might interest you. There are many adherents of Turbo Modula II. Modula II was Wirth's effort at applying what he had learned with Pascal to make an ideal language. I have not used it but many people whom I respect are in love with it. Turbo Modula II was developed by Borland but never released. Again I believe that Alpha is marketing it.

There are versions of the old standbys such as Fortran, Algol and Cobol available. I don't see too many people using them. There are also PD versions of the "Artificial Intelligence" languages Prolog and Lisp as well as a language called Forth which is neither a compiler nor an interpreter but something in between.

If you are interested in seeing what comparable programs in various languages look like, my file MDCLCK13.LBR has programs for reading and setting the MACK Clock in assembly, BASIC, Pascal and C. These are **NOT** meant as examples of proper programming but you can get an idea of what the programs look like.

Correspondence ■ Queries ■ Complaints ■ Editorial Submissions	**For Kaypro-related** articles, advertisements, or memberships in BAKUP ($20/yr), write to P.O. Box 8537, Berkeley, CA 94707-8537; or call Bob Athey at 415-526-3541. **For Morrow-related** articles, ads, or memberships ($20/yr) in BAMDUA, write to P.O. Box 5152, Berkeley, CA 94705, or I. Butler at 415-526-8655.

COMPUTER CLOCKS AND DATE STAMPERS

To: Mike Allen
From: Bill Dobbs, Tucson, AZ

I saw the MOR mail order message about parts being available for your design of a clock (the MACK).

Since I recently converted my MD2 to a MD3 I feel compelled to continue to upgrade what I have. This is driven by the fact that I cannot afford to purchase a newer model of anything.

What can be gained from having a clock? I know I should have taken advantage of the many BBS messages on the subject, but it seems that I have missed something along the way.

Your explanation would be appreciated.

To: Bill Dobbs
From: Mike Allen

What you can gain from the clock will depend on what software you are running.

The big thing you **can** gain is the date/time stamping of your files. You can see when you created, accessed and/or updated a particular file. To use this feature to its full advantage you should be running an operating system that can use a clock. Vanilla CP/M 2.2 cannot. For your MD3 (nee MD2) you'd probably be best off using something like Carson Wilson's Z80DOS, which is public domain and includes the files to use the MACK.

There is also DateStamper from Plu*Perfect Systems which will stamp your files with plain ol' CP/M 2.2. Rick Charnes can give you some information on that.

To be perfectly honest I would not go with the MACK at this time. George Borys has come up with a scheme to use the "Smart Watch" or "No-Slot Clock" on the MDs complete with a nice selection of programs. The big advantage to the Smart Watch is its ease of installation and no construction. Pull out your ROM, plug in the Smart Watch and plug your rom back in on top of the Smart Watch. I believe there may be a couple of jumpers that may have to be put in the Smart Watch itself. The most expensive version of the Smart Watch that I have seen is $49. Shop around and I'm sure you can find it cheaper.

If you want to go with the MACK, Sypko has available some of the stranger parts, most importantly the Printed Circuit board and the trimmer capacitor. Check with him to see what he still has.

Again the Smart Watch is definitely the easier way to go and probably not too much more expensive than the MACK parts. George's info on the Smart Watch is contained in the files SMARTWAT.LBR and SMARTWAT.ART (MOR AND BAMDUA BBS). George does nice work. Any further questions, please ask.

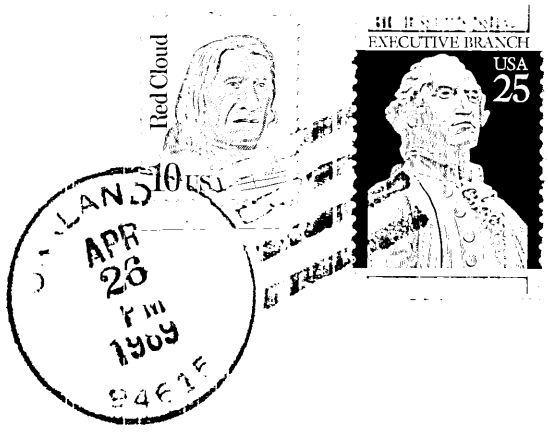
MOR, 415-654-3798

VOLUNTEERS WHO ANSWER HELP CALLS OF MEMBERS

PERSON TO CALL IF YOUR PRIMARY CP/M EXPERIENCE IS WITH KAYPRO	PERSON TO CALL IF YOUR PRIMARY CP/M EXPERIENCE IS WITH MORROW
re: Beginners & General	
Athey - Folsom (beginners SIG)	Campbell - Charnes - Korte
Van Sickle	Oechsli
re: CP/M, MS-DOS & Other Operating Systems	
Fowler - Pugh - de Castro	Borys(hardware) - Butler(Mac)
Winyard - McPheeters(hardware)	Campbell(Mac & Atari) - Korte
re: Other User Groups, Newsletter	
Athey - Winyard	Butler(legal)
re: WordStar & NewWord Word Processing	
Athey - Buck - Peeples	Butler - Campbell - Charnes
	Naparst - Oechsli
re: Database Programs	
Pugh(dBASE) - Cole(Pearl)	Campbell(dBASE) Gowens(Pearl)
re: Perfect Writer / Calc / Filer	
Bruner - de Castro - van Oosten	(not common among Morrow)
Athey(Calc) - Willett(Filer)	
re: Programming Languages	
Bruner(MBasic) Winyard(Pascal)	Borys(MBasic) McKusick(Pascal)
re: ZCPR3 Systems & MEX Modem Program	
Charnes - Fowler	Charnes - Korte
re: Other Programs by Name or Type	
Lautenberger(spreadsheets)	Charnes(BackGrounder)
Becker(Handyman) Bruner(XtraKey)	Johnson(Quest & spreadsheets)
Uzzell(FrameWork)	

415 AREA CODE PHONE NUMBERS & RULES FOR ALL CALLERS

PRIMARY CP/M EXPERIENCE KAYPRO		PRIMARY CP/M EXPERIENCE MORROW	
Bob Athey	BBS or 526-3541	George Borys	til 11 582-7615
Dennis Becker	625-3868	Ilbert Butler	til 11 526-8655
Bob Bruner	BBS or 528-1065	Peter Campbell	527-3387
John Buck	268-9541	Rick Charnes	BBS or 826-9448
Ted de Castro	581-8862	Bruce Gowens	268-9450/845-8002
Leonard Cole	527-2110	Wesley Johnson	til 9 444-0568
Anne Folsom	843-5168	Gene Korte	525-8944
Ken Fowler	222-0830	Lee McKusick	am only 849-9053
Walt Lautenberger	283-2266	Stan Naparst	525-2086
Woody McPheeters BBS	548-3126	Frank Oechsli	527-6089
Chris Peeples	til 11 655-4438		
Steve Pugh	527-7272		
David Uzzell	465-3013		
Jeanne van Oosten	547-4792		
George Van Sickle	682-3188		

RULES FOR ALL CALLERS:
Time: Unless noted, weekdays 7-10 pm, weekends 10am-10pm
Long Distance & Toll Calls: Returned **COLLECT**
No Criticising Unpaid Help!

Morrow Owners Review
P.O. Box 5487
Berkeley, CA 94705