

C.C.O.G.

COLOR COMPUTER OWNERS GROUP

NEWSLETTER AUGUST 1991

OFFICERS:

PRESIDENT **Bernie Patton**
VICE-PRESIDENT. **Chuck Goodwin**
SECRETARY **Marcine Glowicki**
TREASURER **Bob Waite**
CORRESPONDING SECRETARY . . **Robert Gault**
LIBRARIAN **Bob Humphrey**
BULLETIN BOARD. **(313) 292-4713**

INDEX:

1. Pres. Comments
2. Meeting Minutes
3. OS-9
8. Viewpoints
10. Experts Corner

PRESIDENTS COMMENTS:

My grandmother had a favorite saying: "You had better be careful about what you ask for, you just might get it!!!" Well we asked and we got it!! What did we ask for? Vendors to relpy to our offer to place their ad in our little newlsetter for free.

Well, thus far 11 vendors have responded (ads included in this volumn) and also most of them have opted to subscribe to our newsletter. I think this is great!! Many of them indicated that they liked reading our newsletter and had good things to say about the articles.

The september Rainbow has an article on "The Delmar 68000 based System IV". The headline is "A New Era". I hope that Frank Hogg and IMS will also send machines to the Rainbow for review. We have been two years waiting for these "NEW" machines to come to market. Many from the CoCo community have tired of waiting and gone on to other machines. Let's hope that those of us who have stuck it out will be rewarded

with machine capability that will surpass that of the current MS-DOS world.

This meeting it will be almost Labor Day. We only have three more meetings after this until we are into the holiday season. We have to give consideration to what we are going to do about a December meeting (If, where, and when). We also need to have more discussion at meetings about future topics and demonstrations. We need to have more of our membership step up and show us the software you are currently using on your COCO. It doesn't matter how new or old the software is, if it works, if it gets the job done, if you have fun with it then it's not obsolete.

It's soon going to be time for you to be thinking about running for office in YOUR club. We will be having nominations in October and elections in November. Be prepared to help to make this the BEST club anyone could belong to. Bernie

COLOR COMPUTER OWNERS GROUP

This newsletter is a periodic publication of the
COLOR COMPUTER OWNERS GROUP
of Metropolitan Detroit.

Copyright for all articles in this publication remains with the original author. Permission is given to reprint for nonprofit publications so long as original article is intact with credit to the Author. Requests can be made for downloading articles thru our BBS from other CoCo Clubs wishing to reprint articles.
Contact: Bernie Patton (313) 283-2474 BBS (313) 292-4713

C.C.O.G Meeting Minutes July 23, 1991

The meeting was called to order at 7:20 pm. Sixteen people were in attendance. The minutes from last month were accepted.

Old Business--Bernie went through past issues of COCO CLIPBOARD and RAINBOW to formulate a list of CoCo advertisers. He had a list of 57 advertisers. Next he composed a coverletter to send to each advertiser. The letter basically stated that our club, C.C.O.G., would accept advertising from them for inclusion in the club newsletter. There were several conditions but nothing overwhelming. C.C.O.G. would accept their copy of advertising for a time period of 3 months, and this is renewable if they continue to send an updated copy. Also for \$3 the advertisers would receive 3 months of our newsletter. \$6 would get them a one year's subscription to our newsletter. This fee is only to cover the cost of mailing the newsletter since we are a non profit organization. Bernie mailed the letters approximately 2 weeks ago. One letter came back "moved and no forwarding address". Three responses have already arrived, (see this month's newsletter for their advertising), Color System, MV Systems and Howard Medical. Howard Medical also sent a check for \$6. By next month there should be more responses.

Gus also asked if we were going to write a letter to the editor of RAINBOW stating what our club was doing. Bernie sited his closing statement in his coverletter to advertisers, "Accept this in the spirit it was intended." This idea of accepting advertising is not to usurp RAINBOW but it is an offer for those companies unable to pay the high advertising costs to get their product out to the public. Remember our newsletter goes out to other clubs, Radio Shack's and our members.

After all the mailings a postage bill totalling \$17.80 for advertising mailings, \$2.36 postage for last month, and probably another \$2.36 for this month. Bob was not at the meeting to give a treasurers report.

New Business--Robert Gault noticed that we have been dropped from RAINBOW'S Club List. He will be contacting RAINBOW to find out why.

Bob Humphrey resigned as the club librarian. Larry Schneider will be taking over. Thanks Bob for a job well done.

Demonstrations for the evening included Bob Gault working with random number generators and checkers to see how really random they are and Jim Snider showed off his latest hardware project. He used a mini tower AT case and put all his COCO toys in it.

RAINBOWS theme next month is graphics and normally it is games.

Both will be August's theme.

Next meeting Aug. 27, 1991.

OS-9 by Tom Napolitano

Robert Gault last month wrote a benchmark test program for comparison of the processing speed of the Coco3 running a basic looping routine under RSDos BASIC and under BASIC09. The results were interesting, and he suggested that I do the same for C. Actually I started the project then found there were quite a few different ways to express the same algorithm in the C language.

The first version ran through an empty loop 10000 times. This is inconclusive of anything, since it took about 1 second using the internal clock. Since the granularity of the clock is only on second, all this says is that it takes a "very short time". So back to the drawing board.

Here is the program.

```
/* loop1.c */
/* Benchmark test of c compiler */
/* < 1 second @ 2mhz */

#include <stdio.h>
#include <time.h>

struct sgtbuf buff;
main()
{
int i;

    printime();

    for(i = 0; i < 10000; ++i)
        ;

    printime();
}
```

Notice that I declared the loop variable i as an integer. This should be much faster than RSDos BASIC using a float for all numbers. It is hard to tell, so I slowed the program down by stretching out the loop. By the way, printime() is a routine to print the current time on the os9 clock. I'll get into that at this time only to say that how it is written doesn't affect the results much.

```
/* loop2.c */
/* Benchmark test of c compiler */
/* 10 seconds @ 2 mhz */
main()
{
unsigned int i, j;

    for(i = 0; i < 10; ++i)
        for(j = 0; j < 65535; ++j)
            ;
}
```

In loop2 I extended the loop to 655350, in order to get a better measure of running times. From here on, I'll skip printing some of the "overhead" code, and just show the parts of the program that change. I had to unroll the loop with two variables. Why? Because an os9 C integer is only 16 bits of binary data. As you would expect, this loop takes ten times as long. But why use two loops? Why not use just one by declaring the loop variable as a long integer? Surprise, surprise. It takes six times as long!

```
/* loop3.c */
/* Benchmark test of c compiler */
/* 59 seconds @ 2 mhz */
main()
{
long i;

    for(i = 0; i < 655350; ++i)
        ;
}
```

The reason, when you think about it is that in order to compare two long ints, and to increment a long integer, it takes over twice as long. Just shows the inefficiency of C versus assembly. Now lets see if we can speed things up from the ten second time.

```
/* loop4.c */
/* Benchmark test of c compiler */
/* 6 seconds @ 2 mhz */
main()
{
unsigned int i, j;

    i = j = 0;
    while(++i <= 10)
        while(++j )
            ;
    printime();
}
```

Here I used a "while" loop instead of the "for" loop. It has the same effect, but runs faster. Sometime we should examine the resulting output from the c compiler to see how it handles the two constructs differently.

```
/* loop5.c */
/* Benchmark test of c compiler */
/* 10 seconds @ 2 mhz */
unsigned int i, j;
main()
{
    i = j = 0;
    while(++i <= 10)
        while(++j )
            ;
}
```

When you declare int i,j within a function, these are called "automatic" variables. Space is reserved on the temporary stack area of a function as the function is entered, and when the program leaves off executing the function, the automatic variables go away. Instead, in loop5 I declared i and j outside the function "main" as "global" variables. These keep their value as long as the program runs, irrespective of what function is being executed at any given time. It appears that accessing globals is slower than accessing data off the stack.

```
/* loop6.c */
/* Benchmark test of c compiler */
/* 7 seconds @ 2 mhz */
main()
{
double i;

    printime();
    i = 0;
    while(++i <= 10000)
        ;
}
```

As I mentioned, RSDos BASIC declares all numerical data as floating point numbers. So to test how this compares in C, I declared i as a double width floating point number. This slowed things down somewhat, but now I can go back in loop6 to only one loop variable, and a loop of 10000.

```
/* loop7.c */
/* Benchmark test of c compiler */
/* 6.5 seconds @ 2 mhz */
direct int i, j;
main()
{
    i = j = 0;
    while(++i <= 10)
        while(++j )
            ;
}
```

In loop7 we took advantage of a trick of the 6809 architecture, direct variables. What this means is that we are using special 6809 instructions that allow us to access direct data with one less byte of addressing. What this shows here is that it is faster access direct variables than regular globals, but still is faster to access off the stack. Another trick is to declare certain variables in C as residing in registers of the CPU. This only works for one variable on the 6809 processor, but we might as well take advantage of it. It shaves another half second from the previous best time.

```
/* loop8.c */
/* Benchmark test of c compiler */
/* 5.5 seconds @ 2 mhz */
```

```

main()
{
register unsigned int j;
unsigned int i;

    i = j = 0;
    while(++i <= 10)
        while(++j )
            ;
}

```

For those of you who are interested, here are two versions of printime(), with no comments from this person. See if you can decode the somewhat cryptic programming.

/* Version 1 of printime; obscure */

```

int printime()
{
    getime(&buff);

    putchar(buff.t_minute>9?(putchar(buff.t_minute/10+'0'), \
(int)buff.t_minute%10+'0') : (int)buff.t_minute%10+'0');

    putchar(' ');

    putchar(buff.t_second> 9 ? (putchar(buff.t_second/10 + '0'), \
(int)buff.t_second%10+'0') : (int)buff.t_second%10+'0');

    putchar('\n');
}

```

/* Version 2; neater */

```

int printime()
{
    getime(&buff);

    if(buff.t_minute > 9)
        putchar(buff.t_minute/10+'0');

    putchar((int)buff.t_minute%10+'0');

    putchar(' ');

    if(buff.t_second > 9)
        putchar(buff.t_second/10 + '0');

    putchar((int)buff.t_second%10+'0');

    putchar('\n');
}

```

Great OS-9 Software

VED, OS-9 Text Editor.....\$24.95

The best editor for OS-9 just got better. Version 2.0 of this best seller now includes 36 definable macros, case-switcher, and even more speed. See the review in Mar/Apr Clipboard. Works with 128 or 512K.

Upgrades to version 2.0 (with new 28 page manual) are \$12.00 with proof of purchase.

VPRINT, OS-9 Text Formatter.....\$29.95

An unbelievably powerful formatter. Features include complete proportional font support, multiple columns, footnoting, indexing, table of contents. And much more. Comes with 120 page manual, demo files and extensive macro file. (512K memory recommended.)

Ultra Label Maker 9.....\$19.95

Turns your printer into a printing press for labels. WYSIWYG previewing. Supports ALL printers. Useful and lots of fun. One of Rush Calley's "top 10." (Requires 512K CoCo3) CoCo2/3 version \$14.95.

Magazine Index System 9.....\$19.95

Now you can find those references fast. Comes with extensive CoCo magazine data files. File compatible with our RS-DOS version. Another one of Rush Calley's "top 10." (Requires 512K CoCo3.) CoCo2/3 version \$14.95.

Sorry, no credit cards--enclose check or money order plus \$2.00 S/H. Complete catalogue available--send \$1.00 (free with order). Most orders shipped next day!

Bob van der Pnel Software
P. O. Box 57 P. O. Box 355
Wyndel, B.C. or Porthill, ID
Canada V0B 2N0 USA #3853-0355

UltiMusE III

"What if...
all CoCo music programs were this good?"

UltiMusE III (the Ultimate Music Editor, CoCoIII) is a MIDI 'Notation Sequencer'. It lets you write and edit sheet music on a 640 x 192 graphics screen using the mouse, play it on ANY MIDI-equipped synthesizer(s), and print out the score... Written by an experienced computer professional who is also a serious amateur musician and composer. Copy a favorite piece of sheet music just as it looks! Why should your music sound like a machine????? **UltiMusE III** has a wide pitch range, from 4 octaves below Middle C to over 3 above.

NEW FEATURES INCLUDE: Part Copy, Percussion Table/Staves, MIDI Clock, Note Articulation, Note Translator, Note Joiner, FULL Point 'N Click mouse/keyboard menus! **UltiMusE III** is the perfect computer music program for both the professional & beginner musician.

Professional software should use a professional Operating System. **UltiMusE III** uses the advanced features of OS-9 Level 2 and does not interfere with its windowing and multi-tasking in any way.

SYSTEM REQUIREMENTS

CoCo 3 with at least 256K memory & OS-9 Level 2 Mouse or Joystick, Hi-Res Joystick Adapter recommended Synthesizer(s) with MIDI-In jack, plus serial cable, Dot Matrix Printer, a MIDI Interface Pak, & Multi-Pak interface are optional.

UltiMusE III \$54.95

Newspaper Plus

FINAL EDITION

DeskTop Publishing for CoCo 3 just got better! With the **ALL NEW NEWSPAPER PLUS-FINAL EDITION**, you can create complete and sophisticated Banners, Headlines along with Text Columns and Graphics. Bring in different pictures, fonts, fill patterns, and text from disk and create a publication with that pro-look to it. Comes complete with 22 fonts, 50 NewsArt pictures and fill patterns, 128k or 512k Disk

\$42.95

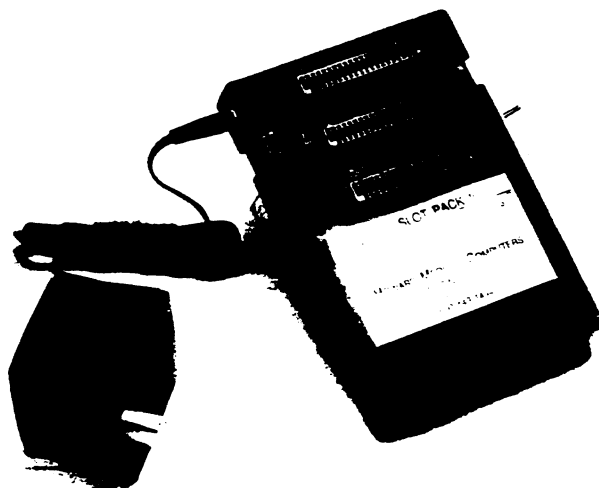
'FINAL EDITION' is just a news print slogan meaning the very latest published issue. In the case of Newspaper Plus - Final Edition, it means the latest up-grade is NOW available. Some of the added features are:

- Text Import with Left, Right, Centered & Justification
- Ram Disk Utility (512k) Stretch, Shrink & Compress picture utility
- A new 'Design Your Own' layout feature
- Full Font import ability
- Text to Picture wrap-around
- Disk Transfer Utility (512k)

ShellMate: By Eric Crichtow \$24.95
 'What Multi-Vue should have been...' NOW work OS-9 Level Two in a point-and-click environment similar to the **DiskMaster** program on the Amiga. **ShellMate** is one of the **FIRST** commercial programs to fully use the **Multi-Vue** windowing system that is both friendly and powerful. Commands like **COPY**, **DELETE**, **RENAME**, **LIST**, and **PRINT** are at your finger tip. Additional features like creating a directory, list the contents of a standard OS-9 archive file as well as create and dearchive a file. How about viewing VEF and GIF pictures... **NO PROBLEM!** **ShellMate** is a well-thought-out and useful program for OS-9. **System requirements:** OS-9 Level 2, Multi-Vue, Mouse or Joystick and at least 256k system memory.

Howard Medical Computers 800-443-1444

SLOT PACK III



Release III of Howard's replacement for the multipack interface combines a smaller case for more desk space and full compatibility with CoCo products

Place your disk controller in the middle slot, a B&B harddrive controller in slot one and an RS232C pack or modem pack in the unswitched 3rd slot:

ROM packs will work in the middle slot or slot 1 with the addition of our hard switch. OS-9 programs will fly with the new addition of the F (fast) chip on the data bus

The new 12 volt adaptor now supplies current for the MP-III instead of drawing it entirely from the Color Computer.

And the added leg underneath the slots add extra stability

Expand your CoCo to its fullest potential with three additional slots of Howard's slot pack III for only \$89.45 and power it with the AC-12, 12 volt adaptor for \$14.95

Howard Medical Computers 800-443-1444

Look what we have for you . . .

OS-9 Budget System

MV Systems is proud to offer this fine **BasicOS9** program written by Mike Dean. More than just a budget program, this system allows you to track and analyze your financial transactions and prepare helpful reports to fit your financial needs. Runs in text or graphics screens. **Requires Tandy Color Computer 3 or compatible w/128k disk drive, and OS-9 Level 2. Multi-Vue optional. Introductory price \$19.95!**

companions to **Multi-Vue's** Goal program, or by themselves to perform many handy scheduling tasks! **Requires Tandy Color Computer 3 or compatible w/128k disk drive, and OS-9 Level 2. Multi-Vue optional. Still just \$14.95!**

Special Offer! Purchase all three programs for just \$54.95!

Software Developers: MV Systems needs quality OS-9/OSK software to market. MV Systems is actively involved in helping software developers like you bring their products to market. We can help you put the finishing touches on your program or its documentation, if needed, and then help you market your product effectively. We can also assist you in porting software between OS9 and OSK (Version 2.4). So, if you have written (or are writing) software you would like to market, contact us for details. You'll be glad you did!

All products carry the Rainbow Certification Seal. VISA and MasterCard orders accepted. Please add \$2.50 (U.S.) or \$5.00 (foreign) for shipping and handling to all orders. Colorado residents please add 3% sales tax.

High Finance

Perform a variety of financial analysis calculations and create schedules of periodic data with this easy to use **Multi-Vue** application. Intelligent financial decisions are just a point and click away! Includes fantastic on-line tutorials that help you learn to use the program quickly and easily. Includes present/future value, sinking fund, loan amortization, depreciation, and much more! **Requires Tandy Color Computer 3 or compatible w/128k (256k or more recommended), disk drive, mouse/joystick, and OS-9 Level 2 w/Window module. Multi-Vue recommended. \$24.95.**

OS-9 Calendar Utilities

Ever wish you could display your schedule for the day automatically on start up? Or, perhaps you have wanted to perform repetitive maintenance tasks on your goal data files quickly and easily? Or, maybe you have given up on Goal totally because you couldn't print your calendar data. MV Systems' **OS-9 Calendar Utilities** will handle all these tasks for you, and more! You can use these popular utilities as

MV Systems

P.O. Box 818
Arvada, CO 80001

(303) 420-7777

Quality OS-9 Level 2 Software from ColorSystems

Specialty Programs	Games!
WPSHel	Variations of Solitaire
An OS-9 Level 2 Word Processing Shell.	Include five separate game programs, five variations of Solitaire.
Features:	Includes the following Variations:
<i>Pull-Down Menus</i> User Customizable Works with most any Editor and Text Formatter (Not Included)	Pyramid Klondike Spider Poker Canfield
<i>Requires Multi-Vue</i>	Complete package for only \$34.95
Complete with documentation for only \$22.00	<i>Multi-Vue is NOT required!</i>
MVBanner	Game Pack Special!
A Banner Generation Program for the Multi-Vue Environment.	Contains ALL of the following:
Features:	CoCobello CoCoYahtzee KnightsBridge Minefield Sea Battle
<i>Pull-Down Menus</i> On-Screen Banner Previewing Image Library and Editor	All five games for only \$34.95
<i>Requires Multi-Vue and an 80 Column Monitor</i>	<i>Multi-Vue is NOT required!</i> <i>CoCoYahtzee requires an 80 Column Monitor</i>
Only \$20.00	

To Order Send Your Check or Money Order to
ColorSystems
 P.O. Box 540
 Castle Hayne, NC
 (919) 675-1706 (Voice)
 (919) 675-1847 (BBS)

No Shipping Charges to the Continental US
 Add \$3 for Canada, Alaska, Hawaii and Mexico
 Add \$5 for other Oversea Areas

North Carolina Residents Please include 5% Sales Tax
 For Complete Product Descriptions Send For Our FREE Catalog!

The OS-9 and Multi-Vue specialists!

"Viewpoints" by Gus Korte"

Recently I contacted club member Jim Snider on the electronic bulletin board system (BBS) the CCOG club supports and arranged to install a FD502 disk drive and a FD500 disk drive into one unit. Not only did he do this with his usual efficiency but also replaced my disk controller ROM with a much more efficient ROM he has developed called an EPROM (electrically programmable read only memory). What a difference!! Now I have 3 disk drives compressed into one space-saving unit instead of 2 separate disk drive units as I had initially. In addition the EPROM does a lot of additional things the previous ROM did not do. When the disk drive is turned on, an 80 column screen appears automatically with white letters on a dark blue background instead of the 32 column green screen with black letters. When I type the command DIR to see the disk directory, instead of the contents scrolling by so fast that they are virtually unreadable, now the information pauses after the screen almost fills. This allows one to read the screen for a change at the reader's pace with a screenful of information each time. Previously I had to use club member Bob Gault's program (in the CCOG club library if you want to use it) to slow down the scrolling. Now each disk drive including a double-sided disk drive is available within one ROM. The step rate using the DSKINI command is now increased too. The VERIFY command is now constantly on which is very handy when copying other diskettes.

Before you can "kill" a file, you must confirm it first. There are many other items Jim has in the EPROM. It also was interesting to see Jim's electronic work space. He is well equipped to work on the COCO. Many thanks to Jim for his assistance in this situation and in others where he has given considerable assistance to us CCOG members. As has been mentioned previously, having a multiple disk drive system is a necessity to get full use of the BASIC09 software and related products.

It was interesting to view the VCR that was made at the 6/25/91 CCOG meeting. All of the comments were clear but for an unknown reason the video became unclear and could not be viewed. The sound portion was clear and any of you who missed or want to re-view the June meeting may want to run this VCR tape which Bernie Patton owns. Thanks to Bernie for producing VCR tapes like this of selected useful presentations at the CCOG meetings for us to view and hear especially where we missed the meeting or want to re-view them.

On page 11F of the Detroit News dated 7/22/91, appeared an article entitled "New software sparks a CD-ROM boom". It stated that "the home market for CD-ROM -- the computer version of the audio CD, or compact disc -- is in the early stages of a boom as personal computers move onto the shelves of mainstream retailers such as Sears." It also stated "Computer users can also play regular music CDs

on their system, controlling volume and other controls by clicking with a mouse on the picture of a CD player on the screen." "Among some of the new offerings: Microsoft Bookshelf, a collection of reference books including a dictionary, thesaurus, almanac and ZIP code directory. National Geographic's Mammals: A Multimedia Encyclopedia, with 700 color photographs, 600 pages of text, 155 animal vocalizations and 45 video clips."


That indicates the ever expanding items available to the computer hobbyist. The memory required for such CD's must be extremely larger than anything now available to us COCO users. Maybe the future COCO4 models will one day be able to include such items but at a very high price. This does indicate the future trend for those willing to invest more into such items.

Karl Sefcik's message in last month's club newsletter is again repeated here for those of you who wish to subscribe to the Rainbow magazine. You can use a free telephone number to order by credit card. It is 1-800-847-0309. Since some of you do not use the BBS yet, this information is included here once again for your information.

A reminder especially for new members of the CCOG, the VFW uses your money donations from our club in it's many charitable activities especially concerning hospitalized veterans. It's all going for a good cause. The Livonia VFW also collects used books and

magazines to give hospitalized veterans. So if any of you want to contribute such, which you would otherwise discard, give them to the VFW bartender to make some unfortunate veterans happy. Also we owe some thanks to the VFW bartender because he volunteers his time to open the VFW Post for us and serve us drinks. He gets nothing for providing us this service so let him know you appreciate his efforts for us whenever you can.

PUT YOUR COCO TO WORK . . .



SALE!

CHECK WRITER. Use your printer and standard bank checks to help you monthly with your accounts. Saves files, releases, edits and pauses printer to let you write in amounts. Has Index screen. Manual. \$19.95

BANKMAN. Continuously updates your check book. Save, edit and print files. Lets you analyze spending; reconciles, summarizes banking transactions. With manual. \$29.95

INVENTORY MANAGER from Ferret Enterprises. Writes invoices, keeps track of stock, saves time and money. . . . \$24.95

GENERAL STATISTICS. Computes over 60 stats, including Standard deviation & Std. error, Covariance, Correlation, Chi-Sq. & ANOVA. Includes screen tables and manual. Edits, Saves & Prints data. User friendly. Great for H.S. & College students. \$29.95

ENVELOPE WRITER. Prints return and mailing addresses directly on the envelope with a DWP or on tractor envelopes with a DMP. Saves on labels! Memorizes return addresses. Prints custom forms with all printers. Perfect for small or large jobs. . . \$17.95

BRAINBUSTERS. Multiple Choice/True-False Test Writing program. Lets you Save, Edit, Select, Randomize & Print. Many more options. Use for business or fun. . . \$29.95

THE EXAMINER. Use with above program to take "On Screen" tests. Grades and Saves results. For home or classroom. . . \$19.95

TRIVIA 1-3. Try these BRAINBUSTER quizzes. History, Sports, Sci., etc. Each \$5.95

BASEBALL STATS. Keeps track of 20 stats. Continuously updates. Full of optional! User friendly. Includes manual. . . . \$29.95

QUICK STATS. A baseball scoreboard! Keeps track of over a dozen stats. Lets you Save, Edit, and Print out data. Manual. . . \$22.95

BALL CARDS. A database for all sports card collectors. Super fast find and retrieval of record files. Manual. . . \$19.95


COCO LABELS. Prints fancy forms with all printers. Select names from alphabetical screen index. Prints duplicates. . . \$19.95

CALCULATOR. One of a kind. Prints out! Simulates adding machine. All standard functions plus 7 memory banks. . . \$14.95

Specify COCO 1, 2 or 3.
Disk Only. Send amount of
money order. Add \$3.00
Shipping/handling.
Illinois Residents add
6.25% Sales Tax.

TRADING POST

DISTRIBUTORS OF
S-STAR SOFTWARE
P.O. BOX 3622
CAMPBELL, IL
61822



Expert's Corner by Robert Gault

Here are some opinions I have about the C language. They probably are controversial but I will be glad to field counter arguments.

There are few (very few) advantages to programming in C versus using in a quality high level language like Basic09. In most cases, Basic09 (for example) is faster, easier to learn and use, and has more built in features than C. But there are reasons for using C. For me, the most important one is the ease with which assembly code can be incorporated into a C program. In fact, I DON'T THINK YOU CAN WRITE QUALITY C PROGRAMS UNLESS YOU KNOW ASSEMBLY LANGUAGE!

I am going to present a program in C which will demonstrate some of C's strong and weak points. The program is my version of "Demon" a graphics algorithm presented in the August, 1989, Scientific American in "Computer Recreations."

First we will look at a pure C version which probably can't be made faster by a rewrite; it takes about 1.4 seconds per pass. Then we will look at techniques for hand optimization that remove some of the horrible code produced by the Tandy C compiler. The modest improvement to 1.3 seconds per pass would have been much greater but for the emphasis on speed in the original C code. Finally, I will present the code for an assembly version of the main loop. This caused a boost in speed to about .33 seconds per pass resulting in a watchable program.

Here is the jist of the algorithm. Picture an array, say 100x100 pixels. It is filled with 15 different colors in a random pattern. Then at each clock pulse the old pattern is tested to give a new pattern as follows:

```
repeat until break
  for i = 1 to 100
    for j = 1 to 100
      for each neighbor pixel(k,l) of (i,j)
        if old(k,l) = old(i,j) + 1
          then new(i,j) = old(k,l)
        else
          new(i,j) = old(i,j)

  for i = 1 to 100
    for j = 1 to 100
      display new(i,j)
      old(i,j) = new(i,j)
```

The values of the old/new array are just the palette color values. One important point. Both the palette colors and the array itself wrap around in a circle. That is both color and location are modulus values. For example with 15 colors, 15+1=0 and old(100+1,0) means old(0,0).

The above algorithm produces some amazing patterns ending in three possible states. Some patterns lock up and go nowhere. Some eventually become all one color. Some reach a "demon" state. Read the Sci.Am. article or type in the program.

We are going to use a version of the random number generator from last month. We will also again use a VDG screen because then we can directly poke pixels for maximum speed.

```
/* Demon: see Scientific American, Aug. 1989 for info on algorithm
   This program must be run from a VDG screen. */
```

```
#include <stdio.h>
#include <time.h>
```

```
#define I_SetStt 0x8e
#define I_GetStt 0x8d
#define SS_AScrnn 0x8b
#define SS_DScrnn 0x8c
#define SS_FScrnn 0x8d
#define SS_ScTyp 0x93
```

```
struct registers{          /* used by gettime */
    char rg_cc,rg_a,rg_b,rg_dp;
    unsigned rg_x,rg_y,rg_u;
};
```

```
direct long next;          /* used by random number */
```

```
direct char i,j,k,l,set,*screen,seq1,seq2;
direct char keypress;
char rnd();
direct int n,sig,temp,temp2,temp3;
int interrupt();
```

```
/* RGB color scheme */
char palette[]={1,8,9,2,16,18,6,48,4,32,36,34,54,62,63,0};
```

```
/* set palette */
char pal_set[]={0x1b,0x31,0,0};
```

```
/* default palettes */
char default_pal[]={0x1b,0x30};
```

```
struct registers reg;
```

```
/* work space for screen data */
direct char *new,*old; /* pointers to data space */
char data0[64][64],data1[64][64];
```

```
main()
{
    unsigned getscreen();
    char ans;
    struct sgtbuf buffer;
    intercept (interrupt);
    seq1=14;
```

```

puts("\xcWelcome to the world of");
puts("Griffeath's Demons.\n");
puts("If you want instructions");
puts("or information press \"y\"");
puts("\n<hit any key to continue>");
while (!read(0,&ans,1));
if(tolower(ans)=='y'){
    puts("\xcGriffeath Demons were described");
    puts("in \"Computer Recreations\"");
    puts("Scientific American, Aug., 1989");
    puts("\nThis version written in C by");
    puts("Robert Gault.");
    puts("\nOnce a 64x64 array is made,");
    puts("resulting patterns depend on");
    puts("the number of colors used.\n");
    puts("<<shift>><<esc>> = restart");
    puts("<<esc>> = quit\n");
    puts("<hit any key to continue>");
    while (!read(0,&ans,1));
}

puts("\xcPlease wait while I initialize\nthe screen with a
random");
puts("pattern....\n\n");

/* get random seed for rnd from system time */
gettime(&buffer);
next=buffer.t_year*buffer.t_month*buffer.t_hour;
next*=buffer.t_minute*buffer.t_second;

if(next<0)
    next=-next;

/* test screen type for VDG screen */
reg.rg_a=1;
reg.rg_b=SS_ScTyp;
if(_os9(I_GetStt,&reg)==0)
{
    puts("Sorry, this program requires\na VDG screen to run.");
    exit(0);
}

/* map high res screen into VDG space */
screen=getscreen();

restart:

keypress=0;

/* clear graphics screen to palette 15 which will be black */
for(n=0;n<0x3c00;++n)
    *(screen+n)=0xff;

```

```

/* set palette colors */
for(i=0;i<16;++i){
    pal_set[2]=i;
    pal_set[3]=palette[i];
    write(1,pal_set,4);
}

/* display the graphics screen */
reg.rg_b=SS_DScrn;
reg.rg_y=1;
_os9(I_SetStt,&reg);

/* set screen pointer to start of graphic image */
screen+=5064;

/* setup a random screen pattern */

seq2=seq1-1;
old=&data0[0][0];
for(j=63,n=5040;j>=0;--j,n-=80){
    temp=j<<6;
    for(i=62;i>=0;i-=2){
        if((k=rnd()&0xf)>=seq1)
            k=seq2;
        *(old+temp+i)=k;
        if((l=rnd()&0xf)>=seq1)
            l=seq2;
        *(old+temp+i+1)=l;
        *(screen+(i>>1)+n)=(k<<4)+l;
    }
    if(keypress) /* break out if shift esc indicated */
        break;
}

/* setup for main loop */
set=1;
for(;;){

    if(keypress){
        screen-=5064;
        goto restart;
    }

    set=set+1&1;
    if(!set) /* this test must be included in the main
loop */
    {
        /* for the ml. version; see text */
        old=&data0[0][0];
        new=&data1[0][0];}
    else /* swap data banks */
    {
        old=&data1[0][0];
        new=&data0[0][0];}
}

```

```

/*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!*/
#include mainloop
/*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!*/

```

```

}
}

interrupt(sig)
int sig;
{
    char ans;
    int temp;
    if(sig==2)
        sig=0;
    reg.rg_a=1;
    reg.rg_b=SS_DScrn;
    reg.rg_y=0;
    _os9(I_SetStt,&reg);
    write(1,default_pal,2);
    puts("\x0c");
    if(sig==3){
        keypress=1;
        puts("Shall I change the number");
        puts("of colors? <y or n>");
        while (!read(0,&ans,1)||ans==3);
        if(tolower(ans)=='y'){
            temp=0;
            while(temp<5||temp>15){
                printf("\x0cCurrent selection: %d\n",seq1);
                puts("Enter the number of colors:\n");
                printf("    4< n <16    ?=>");
                scanf("%d",&temp);
            }
            seq1=temp;
        }
        puts("\x0cPlease wait....");
        return;
    }

    reg.rg_b=SS_FScrn;
    reg.rg_y=1;
    _os9(I_SetStt,&reg);

    exit(sig);
}

```

```

unsigned getscreen()
{
    reg.rg_a=1;
    reg.rg_b=SS_AScrn;
    reg.rg_x=2; /* 160x192 16 colors 16K screen */
    if(_os9(I_SetStt,&reg)!=0)
        interrupt(reg.rg_b);
}

```

Special prices for CCOG members!

One For Me by Val Burke is a fabulous example of what a professional can do with CoCo MIDI Pro. It proves that CoCo MIDI Pro controlling a synthesizer can produce music that doesn't have to sound like a computer at all! Catchy tunes, foot stomping rhythms, and a pleasant variety from slow blues to reggae and latin rhythms is here. This is a tape that you'll want to listen to over and over again! Order now while the price is only \$7.00

A Little Computer Music by Mark Steele is a brand new tape that you won't want to miss! If you liked *Switched On Bach*, you're going to love this one! Mark's music sounds like Bach and Mozart mixed into a bouquet of vibrant synthesizer sounds and spiced with a little rock and jazz. Recorded with the assistance of CoCo MIDI Pro and Lyra. Hurry and get your copy now for only \$7.00

CoCo MIDI Pro is a complete 16 track MIDI recording sequencer and hardware MIDI interface that gives you all the recording and editing features you need to make great music. Find out why musicians love this program! 512K CoCo 3, Multi-Pak/Y-cable required. (while available) \$80.00

Lyra is an easy to use music editor that lets you compose music on a graphics music staff with a mouse. Includes the 100 page Lyra Companion. Synthesizer optional. \$25.00

MusicWare Hamlet Rt. 1261, Seaside, OR 97138 Ordering information: payment by check, money order, or credit card (Visa or Mastercard) is welcome. Please add \$3.00 for shipping on all orders.
 1-800-321-1490 (orders only, please!)
 1-503-738-0119 (inquiries)

```

return(reg.rg_x);
}

```

```

char rnd()          /* returns psuedorandom numbers 0-134456 */
{
    long m=134456, a=8121, c=28411;
    next=(a*next+c)%m;
    return((char)(seq1*next/m));
}

```

This is a pure C version of the main loop.

```

for(i=63;i>=0;--i){
    temp=i<<6;
    temp2=(i-1&63)<<6;
    temp3=(i+1&63)<<6;
    for(j=63;j>=0;--j){

        if((l==*(old+temp+j))==seq2)
            k=0;
        else
            k=1+1;

        if(k==*(old+temp+(j-1&63)))
            l=k;
        else if(k==*(old+temp+(j+1&63)))
            l=k;
        else if(k==*(old+temp2+j))
            l=k;
        else if(k==*(old+temp3+j))
            l=k;

        *(new+temp+j)=1;
    }
}

for(i=63,n=5040;i>=0;--i,n-=80){
    temp=i<<6;
    for(j=62;j>=0;j-=2){
        *(screen+(j>>1)+n)=(*(new+temp+j)<<4)+*(new+temp+j+1);
    }
}

```

Weak point #1 for C is that the random number code should return a value from $0 \leq \text{rnd} < \text{val}$. Most of the time it does but Tandy C does not reliably do integer divide math operations; sometimes it rounds off, then rnd equals val.

Problem #2 involves arrays. It would be nice to say new = old and have the entire array be changed without a loop as is done in Basic09. C does not allow either elements of an array nor the entire array to be swapped by readdressing because the name of an

array is not a variable. It is also NOT nice to use a loop (too slow) nor to duplicate code using a new variable name (too much code.) I have chosen to declare the arrays char data0[64][64], data1[64][64] and access them via the pointers char *new, *old.

Each time through the main loop, *new and *old get reassigned:

```
if(!set)
{
    old=&data0[0][0];
    new=&data1[0][0];}
else /* swap data banks */
{
    old=&data1[0][0];
    new=&data0[0][0];}
```

Then I can use *(old+i<<6+j) as equivalent to data0[i][j]. In fact, the code produced by the C compiler is slightly better with the former version even if it is harder for humans to read. The choice of a 64x64 array is deliberate as it allows the i<<6 rather than i*64. This can't be done unless the array size is a power of two.

Now let's look at the main loop. Everything there has been chosen for speed. We need all the speed we can get because we have at least 64x64x4 operations to perform each pass; and that does not include language overhead!

The loops count down to zero instead of up. The assembly code BGE (branch if greater than or equal to) takes 3 cycles while CMPA #64, BLO (compare A to 64 and branch if less than) takes 5 cycles.

Syntax such as:

```
if((l=*(old+temp+j))==seq2)
```

which says let L equal old[i][j] then if L equals seq2, is faster than separate statements although harder to read. It is faster because C won't STA L,y and then LDA L,y but only do the former. After all why load a register if the value is already there? Calculations based on i are done before entering the j loop, ie. temp=i<<6.

Our next group of problems are the worst ones. When C decodes i<<6 which means shift the value of i to the left 6 times, ie. i x 2**6, the code produced is:
PSHS D, LDB #6, LBSR CCASL where:

* CCASL means C language char arithmetic shift left

```
ccasl tstb
      beq exit
ccasl2 asl 3,s
      rol 2,s
      decb
      bne ccasl2
```



```

exit  ldd 2,s
      pshs d
      ldd 2,s
      std 4,s
      ldd ,s
      leas 4,s
      rts

```

which is truly ugly code, but that is the price you pay for flexibility.

I compiled the source code with CC1 -ac demon.c which left the output with comments at the assembler source level. Then I removed all pshs d, ldd #6 and changed lbsr ccasl to lbsr casl. Then I wrote my own version as:

```

casl  lslb
      rola
      lslb
      rola
      lslb
      rola
      lslb
      rola
      lslb
      rola
      lslb
      rola
      rts

```

The above and a few more odds and ends at the assembly level (ex. LDB SEX CLRA to LDB CLRA) gave a some increase in speed, but not enough.

What techniques should be used if the MAINLOOP is written in pure assembly? Well get rid of as much y relative addressing, unnecessary stacking, and unnecessary subroutine calls as possible for starters. Also get rid of integer operations when byte operations will do and use as many direct page variables as possible for speed. Some code is duplicated, sacrificing compactness for speed. Usually this code would be made into a subroutine.

The largest speed increase seemed to result from converting the screen update routine to assembly. With the C version, you can see the color changes scroll. Changes appear instantaneous in the ml. version. I attribute this to the use of multiple register variables; Tandy C allows only one per function.

The code for the main loop is shown below. It is instructive to compare the code generated by the C compiler against the version below.

```

#asm
      tst <set          * see note above for why this
test was not *

```

```

    bne flop
flip leax data0,y
    stx <old
    leau data1,y
    stu <new
    bra process
flop leax data1,y
    stx <old
    leau data0,y
    stu <new

process pshs y    * save the C data index *
    ldy #64    * number of rows *
m11 ldb #64    * number of columns *
m12 lda ,x
    cmpa <seq2    * largest palette color; 15-black is background
*
    blo m1211
    lda #\$ff
m1211 inca    * set for next higher color *
    cmpb #64    * first column ? *
    beq m121
    cmpa -1,x
    beq store
    bra m1212
m121 cmpa 63,x
    beq store
m1212 cmpb #1    * last column ? *
    beq m122
    cmpa 1,x
    beq store
    bra m1222
m122 cmpa -63,x
    beq store
m1222 cmpy #64    * first row ? *
    bne m13
    cmpa 4032,x
    beq store
    bra m14
m13 cmpa -64,x
    beq store
m14 cmpy #1    * last row ? *
    bne m15
    cmpa -4032,x
    beq store
    bra m152
m15 cmpa 64,x
    beq store
m152 lda ,x    * no change; get old data *
store sta ,u+    * update data array *
    leax 1,x
m16 decb    * next column *
    bne m12
    leay -1,y    * next row *

```

* left as part of the C code *

JWT Enterprises

Attention! MM1 programmers wanted. Many prospective projects already planned. Write for more information.

Optimize Utility Set 1 **NEW!**

Optimize your disks by eliminating fragmented files and compacting your directories for faster file access. Also includes a utility to assess file fragmentation and directory fragmentation as well as excess directory padding.

Optimize Utility Set 1 \$29.95 Foreign postage, add \$3.00

Nine-Times: Each issue contains: 9 helpful and useful programs to help build your OS-9 library * Instructions, examples, and samples of Basic09 procedures and subroutines to help with your own programs and your understanding of Basic09 * C programs and programming examples * Hints, Help columns, and informative articles to advance your knowledge of OS-9 * Supplied totally of 5.25" disk * Bound manual sent to each new subscriber for help in getting *Nine-Times* up and running, as well as tips on using it with a ram disk or hard disk * All graphic/joystick interface for ease of use.

1-Year Subs. \$34.95 Foreign postage, add \$8.00

Back Issues: Available for the *May 1989* through *May 1991* issues. Please write for information on Back issue contents.

Back Issue, ea. \$7.00 Foreign postage, add \$2.00 ea.

Magazine Source: Due to many inquiries, the source code for the magazine graphic shell is being provided as an informational tool. Included is the actual Basic09 source code and compiled modules on disk, as well as documentation and a printed copy of the source code.

Source, \$25.95 Foreign postage, add \$5.00

Foreign postage excludes U.S. Territories and Canada. New rates eff. May 31, 1991

To order, please send U.S. **JWT Enterprises**
check or money order to: **5755 Lockwood Blvd.**

Technical Assistance
(216)-758-7694

Youngstown, OH 44512



All products per Level 2 OS-9
Sorry, no C.O.D.'s or credit cards. Foreign & Canadian orders, please use U.S. money orders.
U.S. checks, allow 3-4 weeks for receipt of first issue/back issue.

Copyright (C) 1991

OS-9 is a trademark of Microvare Systems Corp. and Motorola, Inc.

```

    bne ml1

endml ldu <screen    * display new data on screen *
    ldx <new
    ldy #64
show1 ldb #32
show2 lda ,x+
    lsla                * two pixels per byte *
    lsla
    lsla
    lsla
    ora ,x+
    sta ,u+
    decb                * next column *
    bne show2
    leau 48,u
    leay -1,y          * next row *
    bne show1
    puls y
#endasm

```


One unexpected idiosyncrasy of the Tandy C compiler showed up during this project. If the main loops are written as separate files and added to the mainline using the #include "mainloop" call, there had better not be an IF statement preceding the #include for assembly versions. C causes a FALSE IF statement to branch to the first next C statement present. #Asm does not qualify as a C statement for PASS1 nor PASS2 so your entire assembly source code will be skipped over during False IF conditions during run time.

That is why the assembly version moved the if(set==0) into the assembly code.

Club members with OS-9 but not the C language can get a compiled copy of the ml. version from the BBS and eventually our library.

6809s FOREVER!

the Rulaford Research BBS



Introducing a new MIDI Music BBS!

- Use your computer, a terminal program, and a modem to communicate with other users nationwide
- Download public domain programs for the CoCo and IBM
- Try out new music programs by downloading demos
- Order products on-line
- Set your modem for 300, 1200, or 2400 baud, 8 bits, no parity (call (619) 690-3648 if you need assistance)
- No connect fee!

Call (619) 662-2428 today!

C.C.O.G.

COLOR COMPUTER OWNERS GROUP

Oldest COCO Club in Metro Detroit
Meetings the ~~THIRD~~^{FOURTH} Tuesday of each month

7:00 PM — 9:30 PM

LIVONIA VFW — 7 MILE & MIDDLEBELT
SIDE DOOR — "THE FOXHOLE"

Join other COCO owners for Hardware and Software demos,
Public domain Library, Discounts, and general knowledge exchange.

1991 Meetings

Jan 15 **Jul ~~16~~ 23**

Feb 19 **Aug ~~20~~ 27**

Mar 19 **Sep ~~17~~ 24**

Apr 16 **Oct ~~15~~ 22**

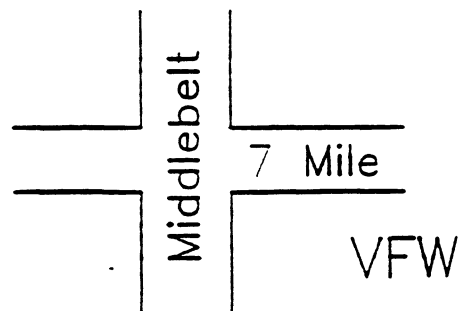
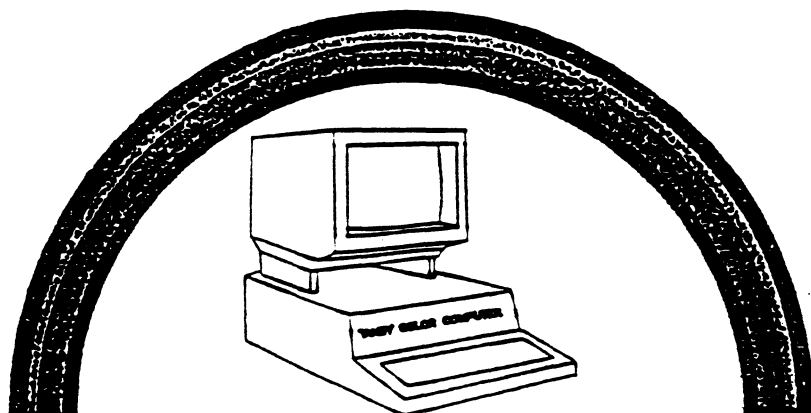
May ~~21~~ 28 **Nov ~~19~~ 26**

Jun ~~18~~ 25 **Dec ~~17~~ ?**

MEETINGS INFORMAL AND OPEN TO THE PUBLIC

Contact: Bernie @ (313) 283-2474

BBS @ ^{or} (313) 292-4713



COLOR COMPUTER OWNERS GROUP
C/O BERNIE PATTON
388 EMMONS BLVD.
WYANDOTTE, MI 48192