the world of 68' micros
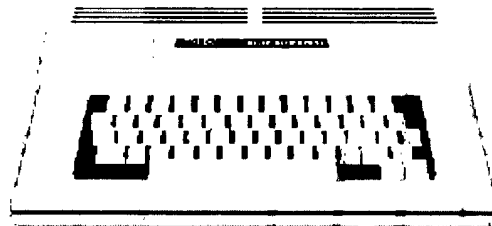
NEW EDITOR FOUND!

**Let's have a big welcome for Stephen Disney!
More info on "Editor's Page" and page 4.**

Sure this will mean changes -- hopefully for the benefit of the subscribers. Just remember to support the new editor in your fullest (I will be doing so!). After all, this magazine is really a co-operative effort between YOU, THE READERS, and the editor. Neither of you can exist without the other, please remember that!

Lately the changes have been leaning more toward those new to the CoCo or "rediscovering" it after a long hiatus. Is this acceptable? Is this what you want from YOUR magazine? If not, please let Stephen know -- he has a big job ahead of him and needs all the support and suggestions you can offer!

CONTENTS

If your address is incorrect, send me a postcard!

The Editor's Page

NOTE: *Okay, so this editorial is a little long. I explain a bit more than necessary, but at least skim through it so you see what is happening to YOUR magazine, and why. Besides, it will be one of the last that I write.*

ANOTHER NOTE: *There will only be FIVE issues in this volume due to the lack of time on my part (read on for an explanation). I will attempt to make up for this somewhat by making this and the next issue a bit larger. Sorry for the inconvenience, but couldn't be helped.*

Well, there it is on the front page! A new editor has finally been found. The choice was pretty easy -- only two people contacted me that were sincerely interested in keeping the legacy of the CoCo alive by continuing this publication (and felt they had the time to do so!). Stephen Disney is introduced on page 4.

Why a new editor?

Well, there are several reasons. The first is time on my part. It is just getting restricted to the point I had to take a week off work just to get this issue out, and it is way late! As most of you know, I am in the U.S. Air Force. My unit has a big ORI (Operational Readiness Inspection) coming up in late February. This inspection tests our war-time capabilities through a simulated excercise. In order to insure our best performance, we have been having our own base excercises since January 98. This has minimized my spare time every couple months, as the excercise takes at least a week and is 12 hour shifts (14-16 hour days or nights), 24 hours a day. Then there is the time preparing (at least two evenings or one full day on the weekend) for the excercise and recovering afterwards. And it takes about three days of rest to recover from six in the field.

Just what do we do "in the field"? I'm in a Civil Engineer Squadron, the 778th to be exact. Our job is to either augment an existing base or build an entire base from scratch at an airstrip somewhere. Because my CE unit is attached to a tanker squadron, we mainly augment an existing base or would build up military capability at a civilian airport. The tankers are built on large Boeing 707 or Douglas DC-10 aircraft -- they need lots of room for take-off and landing. Our team of 130 people can now set up a mini camp ready to use in about seven hours -- took a lot of practice to get there! After camp set-up, there is some camoflauging, hardening (mainly sand bags!), and setting up some "nice to have" items.

About 12 hours after arriving in the field, the deployment phase is over and the ATSO (Ability To Survive and Operate) phase sets in. In other words, the war starts! From then on it is chemical suits and gas masks, repairing runways, looking for injured people and unexploded bombs after an air attack, sweating for hours in a chemical suit and gas mask following a chemical attack, and many other tasks required of CE to maintain a functioning camp under such rugged conditions. Takes a lot out of you!

Under normal conditions we "play" games two or three times a year just to maintain capability. We have had four this year already, with two more scheduled before the February ORI.

While most of you don't have all that to go through, you do have other hobbies and a family. Those require some time also. The way it has been lately, I have to really work to have time for everything!

Okay, so after February things will be better. As long as we get an acceptable rating (and I'm positive we will!), that's true. Another ORI won't occur for this base for another four years.

BUT -- I'm trying to get a reassignment to a job (training instructor) that will allow me to progress in my career easier. The down-side is it also will mean I have a little less spare time to do things like "68' micros".

the world of 68' micros

Advertising Rates:
Contact publisher. We have scales to suit every type of business. Special rates for entrepreneurs and "cottage" businesses.

Contributions:
All contributions welcome. Submission constitutes warranty on part of the author that the work is original unless otherwise specified. Publisher reserves the right to edit or reject material without explanation. Editing will be limited to corrections and fitting available space. Authors retain copyright. Submission gives publisher first publication rights and right to reprint in any form with credit given author.

CoCo Interests....

Unfortunately, my interest in the CoCo has been waning, and time with it has been limited. I will continue to maintain a complete CoCo system (hardware) as well as keeping the CoCo3 emulator up on the clone. I just think it best to pass the magazine on before I get so swamped with work, other projects, and waning interests overwhelm me. That wouldn't be good for the magazine, for me, and most importantly for you.

I will still maintain a prescence in the CoCo community, and still have an interest in the CoCo. I will be starting a column on converting IBM GW-BASIC programs with this issue that I expect to last at least a year. I'll also be making other contributions and assisting Stephen in any way I can. I sincerely hope that all of you will do the same.

"The Deal"

I'm sure many of you are wondering just how Stephen will handle things. For starters, we are going to reduce the average page count to 20 (last few issues have averaged 20 pages, with some issues having more). Along with this comes a $2 reduction in subscription fees. There really shouldn't be any loss in content, as the previous subscription price was based on an average of 24 pages and I've had a hard time filling that many. Rather than struggle to fill pages to justify price, a reduction was deemed a better value for the customer.

Along with the slight price reduction, Stephen has agreed to produce the magazine for a minimum of two years (through July 2000). Longer depends on the amount of subscribers who remain faithful and supportive. After two years, the decision to continue is solely in Stephen's, and your, hands.

I'll maintain the subscriber database and fund Stephen until renewals start going in to him, which will start in January 1999. Until then, renewals sent to me will be passed on to him -- funds and database.

Stephen is taking over publication and I am paying for the outstanding issues owed each of you. **Renewals will continue to come to the current address for now. Don't wait until the "passing of the torch" to renew, you don't want to miss an issue! A complete database and funds will be passed on to Stephen at the right time. And continue to make checks payable to FARNA Systems until you read differently in this magazine.**

All in all, I think you, the readers, will be getting a good deal. A new editor with new ideas will invigorate the magazine and make it much better in the long run. So give Stephen Disney a chance by renewing, and give him your full support!

P.S. -- I have a limited number of back issues on hand, and these are on sale! If you want FOUR or more picked at random, they are $2.25 each. Over eight is only $2.10 (for number nine and up). I will pick before or after a certain date, volume, and/or number, but not scan for subjects. I'll fill in closest available if any are out. If you want me to scan for subjects, cost is $3.00 each. If I don't have four on a subject, the odd ones will be picked at random. If you want less than four, they are $3.00 each, screened for subject $3.50 each and I'll refund or pick at random any if subject not available -- your choice. Of 33 issues, only 17 are currently on hand. If you want all 17, act fast! You can get all 17 for $40. I only have two or three of some issues! I'll refund moeny of those I run out of. If you want certain issues, specify and cost will be $4.00 for the reprints. After the first of the year all back issues will be available from Stephen. **Another Deal: If you want an entire volume of back issues as a single bound book, the price will be $20 until December 15.** After this date they may not be available at that price again. Specify volume 1-5 (this is the last issue of volume 5).

My name is Stephen Disney. I am 22 years old and married. My wife's name is Melissa. We don't have any kids, yet. I was born in Newport News, Virginia, but am currently living in Statesboro, Georgia (home of the Allman Bros. "Statesboro Blues").

When I was a child I didn't have much in the way of video games. First, I had an Atari 2600. Then, my parents bought the family a Color Computer 2 with a tape recorder for saving data. Among the cartridges I had, my favorites were Downland and Audio Spectrum Analyzer. We had the latter hooked up to our living room sound system, and would frequently leave it on the kaleidoscope setting while listening to music. I learned to write BASIC programs at this time. Later on we moved on to the various Tandy 1000 models and the included Tandy Basic (and, of course, DOS). We also owned 286, 386, and 486 PCs. I currently own a 200 MHz pentium.

In my teen years I worked for my father and his company, Advanced Church Technologies. We started out doing photographic work (mostly song slides) and sound installations for churches (Can you imagine running PostScript on a 286 with a Tandy dot matrix? Don't try it. It's not fun and it takes forever to format pages.). Later, we branched out to include clubs, schools, military installations, theaters, etc. We also do a lot more lighting and A/V work in our installations now. I haven't done much with the company lately, but it is still booked solid with new installs.

Taking a step back, I can't remember what happened to our CoCo 2 to save me. In a fit of nostalgia last year, I bought a CoCo 3 from a pawn shop for $10. It was in great shape and was labeled "Tandy Keyboard" on the tag. Then, I looked the CoCo up on the internet. I discovered some sites on the web and the CoCo listserve. To my surprise, I knew a lot less about the CoCo than I thought. I had never heard of OS/9 and the use of floppy drives, hard drives, modems, etc. I hunted down Al Dages phone number in Stone Mountain, and bought the supplies I needed to get started again.

Currently, my CoCo 3 is 512k. I have a multipak, 2 floppy drives, a 40 meg. B&B hard drive setup, Orchestra 90 pak, MIDI pak, RS-232 pak, modem, and DMP-132 printer installed. I am currently using a composite monitor with sound, but plan to go to RGB soon. All this sits in the living room on the same desk as my pentium, and gets about the same amount of use too. I also have a CoCo 2 setup with a multipak in my bedroom for playing Downland, ASA, the Appliance and Light Controller pak, X-pad, and others.

I took the position as editor because I feel that it will give me a chance to contribute to the CoCo community. I am not the most knowledgeable person, and I am young, but I feel that I can keep the magazine going while providing pertinent info to the community and learning more myself. My email of choice is disneys@hotmail.com, but I can also be reached at disneys@mailexcite.com. My ICQ number is 9142149. My phone number is (912) 587-2042. My address is below. Feel free to write with your comments or submit material at any time. See you in the next issue!

68' micros
25213 U.S. 301 North, Lot #1
Statesboro, Georgia 30461

Editor's Note: As of January 1999 all correspondence, including renewals, should be sent to the address to the left of this box. Stephen will be assuming full responsibility of the magazine at that time. Any renewals or correspondence sent to the old address will be forwarded to Stephen.

Converting GW-BASIC to DECB Part 1   *Frank Swygert*

The making of "CoCo Family Recorder" - the key to your disk and MENU.BAS

Introduction...

Many of you with an interest in tracking your family genealogy have purchased a handy little program I wrote back in 1992 called "CoCo Family Recorder" (CCFR). I must admit that I am far from a good programmer, even in Disk Extended Color BASIC (DECB). But I can manage to write simple text code -- given enough time.

CCFR took me most of 1991 to code. It is a simple to use, yet complex and fast database program. It has some drawbacks, like creating records then filling in info, but these drawbacks make it fast and easy.

It would have taken me much longer to code if I had done everything from scratch. In fact, I most likely would have given up on the daunting task of writing a database program of such magnatude had I not a guide -- the GW-BASIC program "Genealogy on Display" (version 4.0).

My first thought was to get an MS-DOS based GW-BASIC genealogy program and convert it to the CoCo3. Sounds easy enough, and the BASIC languages are really quite similar -- both BASICs use up to 64K per module and have 80 column displays. The 32 column DECB display of the other CoCo models make converting or rewriting much more difficult -- almost impossible with text processing programs.

I purchased a book entitled Basic Program Conversions" (HP Books, Box 5367, Tucson, AZ 85703; 602-888-2150; ISBN #0-89586-297-2; Copyright 1984) that had BASIC syntax and conversion notes for IBM, Commodore 64, Apple IIe, TRS-80 Model III & IV, and Tandy Color Computer BASIC and Disk Extended Color BASIC (check your local library for a copy). While it didn't cover Super Disk Extended Color BASIC (SDECB; CoCo3 BASIC), most commands were there and I had my CoCo3 programming manual. I could easily apply the extra CoCo 3 commands needed (mainly LOCATE for screen display) and if the GW program had a command not in the book I'd look through the CoCo3 manual and see if there was an equivalent. Simple right?

Well, the idea was simple, and for simple programs it works well. But a database is not exactly a simple program. For starters there is a lot of disk activity. For the CoCo this is simple, as disk access commands are built into BASIC. Not so for GW! All IBM Clone BASICs make calls to the operating system to handle disk operations.

I had originally tried to contact the author of "Genealogy on Display" (Melvin O. Duke), but was unable to locate him. The program had been last updated February 26, 1986, so it was five years old at the time. My intent was to simply make a CoCo version of the program. While a few modules (there are 14 used in CCFR, 25 on the origi-

nal disk -- some of which aren't necessary for operation and were eliminated) are nearly simple conversions, about 80% of the code had to be totally rewritten. And much of that is vastly different from the original.

Converting didn't happen, as the BASIC languages are just to different. But I had a great guide. I could see how a process was handled under GW, run it on my PC clone, and then duplicate the same process on the CoCo. More like reverse engineering, only I had to figure out complete routines for a single GW command at times. Some routines were in "BASIC Program Conversions", others I had to write on my own.

It will take some time, but the easiest way I could come up with to show readers the process is to print listings of some of the original GW programs alongside the CoCo3 listings. Where the code isn't self explanatory I've made notes at the end.

The Key to your CCFR disk...

What will the listings do for you, since the CCFR disk directory is coded? Well, the code is pretty simple. The directory wasn't coded to keep everyone out, just those who might change the program and not know enough about DECB to avoid problems. I didn't need the support hassles of having someone make a minor change in one module and screw up the entire program. But if you know how to scan a disk (using something like the "scan" function in ADOS) and a little DECB programming, the code was easy to break. My thinking was that if you could break the code, you knew enough to make changes or not expect support for your mistakes. And of course you'd make a backup and play with that, not the original disk.

The simple coding idea came from an old "Rainbow" magazine. I can't recall who came up with it or I'd certainly credit them here. The DECB program modules don't have readable character names, as one can tell from looking at the directory on an original CCFR disk. The names are in ASCII code, with some "invisible" characters (such as a space) used. The invisible codes are 03, 12, 13, 21, and 32 (End of Text, Form Feed, Carriage Return, Negative Acknowledgement, and Space, respectively). The remaining characters are in the extended character set, ASCII codes 128-143. These are all special symbols and pronunciation characters.

Each file name was coded with a four character name and a three character extension. To load the MENU.BAS program from an original disk, type the following:

```
LOAD        CHR$(130)+CHR$(32)+
CHR$(130)+CHR$(03)+"/"+CHR$(03)
+CHR$(12)+CHR$(13)
```

Then press the ENTER key. The MENU pro-

gram should load. List the program and the codes for the other program modules can be easily seen in the listing. The menu program should display "é é" from a standard CoCo3 DIR command, "é-é----.---" from a disk utility such as the ADOS menu, with "-" representing a spce with no character printed. A simple yet effective method of locking unknowledgeable users out.

A couple other tricks were used, but to make the program more user friendly and to protect the coded menu. I could hardly run the MENU program without exposing the coded module names, so a little utility program called "LOADER" was used. LOADER loads a BASIC program as a machine language program. It does this by writing a start, end, and execute address for the BASIC program and creating a binary file with only that information. The created file can then be loaded using "LOADM" and executed just like any other machine language program. I wrote a BASIC program to allow the user to setup program parameters and run the disk drives and program reliably at double speed, all transparent to the user(many people assumed the programs were machine language because of the speed). This program was loaded as a machine language program (BOOT.BIN on the CCFR disk) using LOADER.

The only way to view this program from the disk is to get the start and end addresses from the BOOT.BIN file and search the disk at those locations. For your convenience, the BOOT.BAS program is listed in this issue. The routines in the BOOT program may be used in your own, or you can use the entire BOOT.BAS program as long as you give me credit for it. The FARNA Systems information may be removed and your own added.

To make the program even easier, another simple utility called "DOSBOOT" was used. DOSBOOT writes a short program that will run a specified program unpon typing DOS. This works the same way that OS-9 boots. When the DOS command is invoked, the computer looks at a certain disk location and executes the instructions located there (if memory serves this location is track 0 sector 0). For CCFR, I specified BOOT.BIN.

Conclusion...

The listings are side by side beginning on the next page. Compare them line by line when possible. There are some obvious differences, such as the CoCo only alowing only two characters for variable names whereas GW allows up to 40. There are also the GW and MS-DOS setup parameters. Others will be noted within or after the code. Enjoy the coding excercise!

NOTE: All program parameter setup is accomplished within the GW-BASIC program. For the CoCo, a little more work was required for setup so a separate "BOOT" program was used. I wanted to make the program as fast as possible, so SDECB was patched to reliably perform disk I/O at double speed. Printer baud rates were modified for double speed also. The routines should be easy to pick out. As one can see, the differences in the MENU programs are many and a straight conversion was impossible. It was simpler to rewrite the menu program for the CoCo without following the GW code.

```
100 REM MENU Program
110 REM Menu of the available Genealogy Programs.
120 REM By: Melvin O. Duke. Last Updated 26 February 1986.
200 REM Screen Definitions
210 S1 = 0     'Set Text Mode
220 S2 = 1     'Enable Color
230 S3 = 0     'Active Page
240 S4 = 0     'Visual Page
250 WIDTH "scrn:", 80
260 SCREEN S1, S2, S3, S4
300 REM Color Definitions
310 K = 0      'blacK
320 N = 1      'blue (Navy or uNderline)
330 G = 2      'Green
340 B = 3      'cyan (light Blue)
350 R = 4      'Red
360 P = 5      'magenta (Purple)
370 O = 6      'brown (Orange)
380 W = 7      'White
400 REM Disk Definitions
410 DD.MENU$ = "a:"
420 DD.VERI$ = "a:"
430 DD.PROG$ = "a:"
440 DD.PERS$ = "a:"
450 DD.MARR$ = "a:"
460 DD.ORD$ = "a:"
470 DD.PCIDX$ = "a:"
480 DD.MARIDX$ = "a:"
500 REM Printer Definitions
510 FORM.FEED$ = CHR$(12)
520 PAP.SENS.ON$ = CHR$(27) + "9"
530 PAP.SENS.OFF$ = CHR$(27) + "8"
540 PAP.LONG$ = CHR$(27) + "C" + CHR$(66)
550 PAP.SHORT$ = CHR$(27) + "C" + CHR$(51)
560 COMPR.ON$ = CHR$(15)
570 COMPR.OFF$ = CHR$(18)
580 BOLD.ON$ = CHR$(27) + "E"
590 BOLD.OFF$ = CHR$(27) + "F"
600 REM Maximums
610 MAX.PER = 500
620 MAX.MAR = 200
630 OLD.MAX.PER = 0
640 OLD.MAX.MAR = 0
650 MAX.GEN = 4
660 MAX.LINES = 58
670 CHART.NOS$ = "n"
700 REM Titles
710 TITLE$ = "MENU of Programs in Genealogy"
720 TITLE$ = TITLE$ + " ON DISPLAY"
730 VERSION$ = "Version 4.0"
740 COPY1$ = "Copyright (c) 1983 thru 1986, by:"
750 COPY2$ = "Melvin O. Duke"
760 PRICE$ = "$45"
770 ADDR1$ = "Melvin O. Duke"
780 ADDR2$ = "P. O. Box 20836"
790 ADDR3$ = "San Jose, CA 95160"
800 REM Make sure that BASIC was invoked with /s:256
810 ON ERROR GOTO 870
820 OPEN DD.VERI$ + "verifile" FOR RANDOM AS #1 LEN = 256
830 ON ERROR GOTO 0
840 FIELD 1, 128 AS DUMY1$, 128 AS DUMY2$
850 CLOSE #1
860 GOTO 1000
870 IF ERR = 5 THEN 880 ELSE ON ERROR GOTO 0: GOTO 820
880 REM File Buffer less than 256 bytes
890 KEY ON: CLS : LOCATE 10, 1: COLOR R, B
900 PRINT "BASIC must be brought up with /s:256, for Genealogy Records."
910 PRINT "Program has been Terminated."
920 PRINT "Enter CONT to return to DOS"
930 COLOR W, K: STOP
```

page 6 the world of 68' micros

```
10 REM **** BOOT Program
20 POKE113,0:POKE65497,0:ONERRGOTO1270:ONBRKGOTO1280
30 POKE65497,0
40 REM Copyright (c) by F.G. Swygert, January 1993
50 IFPEEK(269)*256+PEEK(270)=32401THENGOTO610
60 WIDTH32:CLS:PRINT@100,"THE COCO FAMILY
RECORDER":PRINT@164,"(C) 1993, FARNA SYSTEMS"
70 PRINT@262,"AN 80 COLUMN MONITOR":PRINT@298,"IS
REQUIRED!"
80 PRINT@451,"(RGB WILL BE B&W UNDER 'C')"
90 PRINT@388,"RGB OR COMPOSITE (R/C)";:INPUTM$
100 IF M$="C"THEN130
110 IF M$="R"THEN140
120 GOTO60
130 FORX=0TO7:PALETTEX,63:NEXTX
140 WIDTH80:CLS1:LOCATE0,3:ATTR2,0,U:PRINTSTRING$(79,32);
150 LOCATE35,4:ATTR2,0,U,B:PRINT"        ";:ATTR2,0,B
160 LOCATE30,5:ATTR2,0,U,B:PRINT"  ";:ATTR2,0,B
170 LOCATE34,5:PRINT"/-ARNA Systems"
180 LOCATE30,7:ATTR2,7:PRINTSTRING$(13,32):LOCATE48,7:ATTR
2,0:PRINTSTRING$(32,32);
190 LOCATE30,8:ATTR2,7:PRINT"   /\—/\": LOCATE48,8:ATTR2,0:
PRINTSTRING$(32,32);
200 LOCATE30,9:ATTR2,7:PRINT"   < - - >":LOCATE48,9:ATTR2,0:
PRINTSTRING$(32,32);
210 LOCATE30,10:ATTR2,7:PRINT"     \ /":LOCATE48,10:ATTR2,0:
PRINTSTRING$(32,32);
220 LOCATE30,11:ATTR2,7:PRINT"      V":LOCATE48,11:ATTR2,0:
PRINTSTRING$(32,32);
230 LOCATE30,12:ATTR2,7:PRINTSTRING$(13,32):LOCATE48,12:
ATTR2,0:PRINTSTRING$(32,32);
240 ATTR2,0:LOCATE31,14:PRINT"(the FARNA Fox!)"
250 LOCATE35,17:PRINT"Presents:"
260 LOCATE25,19:PRINT"The CoCo Family Recorder 1.1 !"
270 LOCATE0,20:ATTR2,0,U:PRINTSTRING$(79,32);:ATTR2,0
280 PLAY"T6;O2;L2;G;L4;C;D;E;F;L2;G;C;P16;C"
290 PLAY"L2;A;L4;F;G;A;B;O3;L2;C;O2;C;P16;C;F;L4;G; F;E;D"
300 PLAY"L2;E;L4;F;E;D;C;L2;O1;B;O2;L4;C;D;E;D;L1;C"
310 LOCATE17,22:PRINT"(c) 1991... Please wait while system is set up";
320 CLEAR500,32400:RESTORE
330 ONERRGOTO1150:ONBRKGOTO1280
340 TL=0:LT=0:LN=250:T=32401
350 READA$
360 IFA$="**"THEN410
370 A=VAL("&H"+A$)
380 TL=TL+A:LT=LT+A:POKET,A
390 T=T+1:GOTO350
400 LN=LN+10:LT=0:GOTO350
410 A$=CHR$(142)+CHR$(126)+CHR$(145)+CHR$(191)
420 A$=A$+CHR$(1)+CHR$(13)+"9":A=VARPTR(A$):POKE32730,
PEEK(269)
430 POKE32731,PEEK(270):EXEC(PEEK(A+2)*256+PEEK(A+3))
440 GOTO640
450 DATA34,76,B6,1,57,81,BF,10,27,0,6,35,76,6E,9F,7F,DA,B6
460 DATA1,56,81,BF,10,27,0,6,16,FF,EE,60,0,86,86,FE,D6,6F,F7
470 DATA7E,AE,97,6F,12,12,12,12,12,12,86,D,AD,9F,A0,2,96,E7
480 DATA81,1,10,24,0,6D,8E,4,0,5F,A6,84,B7,7E,AF,86,AF,A7,84
490 DATAB6,7E,AF,81,80,10,24,0,48,81,1F,10,23,0,47,81,60,10
500 DATA24,0,46,1A,50,AD,9F,A0,2,B6,7E,AF,A7,80,CB,1,C1,20
510 DATA10,25,0,15,5F,A6,82,B7,7E,AF,86,AF,A7,84,86,D,AD,9F
520 DATAA0,2,B6,7E,AF,A7,80,8C,5,FF,10,23,FF,B3,F6,7E,AE,D7
530 DATA6F,86,FF,B7,1,57,16,FF,70,86,20,16,FF,BF,8B,60,16,FF
540 DATABA,80,40,16,FF,B5,8E,40,0,5F,1A,50,86,76,B7,FF,A2,A6
550 DATA84,B7,7E,AF,A6,1,B7,7E,B0,86,20,A7,84,86,C0,A7,1,B6
560 DATA7E,AF,AD,9F,A0,2,B6,7E,AF,A7,80,B6,7E,B0,A7,80,CB,1
570 DATA96,E7,81,1,10,27,0,1D,C1,50,10,25,0,D,5F,17,0,3F,86
580 DATAAD,AD,9F,A0,2,17,0,49,8C,4E,FF,10,23,FF,B2,16,0,1A,C1
590 DATA28,10,25,0,D,5F,17,0,22,86,D,AD,9F,A0,2,17,0,2C,8C
600 DATA47,7F,10,23,FF,95,F6,7E,AE,D7,6F,86,7A,B7,FF,A2,86
610 DATAFF,B7,1,57,16,FE,E0,A6,83,B7,7E,AF,A6,1,B7,7E,B0,86
620 DATA20,A7,84,86,C0,A7,1,39,B6,7E,AF,A7,80,B6,7E,B0,A7,80
```

GW-BASIC

```
940 SYSTEM
1000 REM Produce the first screen
1010 KEY ON: CLS : KEY OFF
1020 REM Draw the outer double box
1030 R1 = 1: C1 = 1: R2 = 24: C2 = 79: GOSUB 1300
1040 REM Find the title location
1050 TITLE.POS = 40 - INT(LEN(TITLE$) / 2)
1060 REM Draw the title box
1070 R1 = 3: C1 = TITLE.POS - 2: R2 = 6: C2 = TITLE.POS +
LEN(TITLE$) + 1: GOSUB 1460
1080 REM Print the title
1090 LOCATE 4, TITLE.POS: PRINT TITLE$
1100 LOCATE 5, 40 - INT(LEN(VERSION$) / 2): PRINT VERSION$;
1110 REM Draw the Contribution box
1120 R1 = 7: C1 = 18: R2 = 17: C2 = 61: GOSUB 1300
1130 REM Request the Contribution
1140 LOCATE 8, 20: PRINT "If you are using these programs, you are"
1150 LOCATE 9, 21: PRINT "expected to become a Registered User,"
1160 LOCATE 10, 20: PRINT "by making a contribution to the author"
1170 LOCATE 11, 23: PRINT "of the programs (" + PRICE$ + " sug-
gested)."
1180 REM Draw the Mailing Label
1190 R1 = 12: C1 = 28: R2 = 16: C2 = 52: GOSUB 1460
1200 REM Print the Name and Address
1210 LOCATE 13, 40 - INT(LEN(ADDR1$) / 2): PRINT ADDR1$;
1220 LOCATE 14, 40 - INT(LEN(ADDR2$) / 2): PRINT ADDR2$;
1230 LOCATE 15, 40 - INT(LEN(ADDR3$) / 2): PRINT ADDR3$;
1240 REM Draw the Copyright box
1250 R1 = 19: C1 = 21: R2 = 22: C2 = 59: GOSUB 1300
1260 REM Print the Copyright
1270 LOCATE 20, 40 - INT(LEN(COPY1$) / 2): PRINT COPY1$;
1280 LOCATE 21, 40 - INT(LEN(COPY2$) / 2): PRINT COPY2$;
1290 GOTO 1620
1300 REM subroutine to print a double box
1310 COLOR P
1320 FOR I = R1 + 1 TO R2 - 1
1330  LOCATE I, C1: PRINT CHR$(186);
1340  LOCATE I, C2: PRINT CHR$(186);
1350 NEXT I
1360 FOR J = C1 + 1 TO C2 - 1
1370  LOCATE R1, J: PRINT CHR$(205);
1380  LOCATE R2, J: PRINT CHR$(205);
1390 NEXT J
1400  LOCATE R1, C1: PRINT CHR$(201);
1410  LOCATE R1, C2: PRINT CHR$(187);
1420  LOCATE R2, C1: PRINT CHR$(200);
1430  LOCATE R2, C2: PRINT CHR$(188);
1440 COLOR W
1450 RETURN
1460 REM subroutine to print a single box
1470 COLOR B
1480 FOR I = R1 + 1 TO R2 - 1
1490  LOCATE I, C1: PRINT CHR$(179);
1500  LOCATE I, C2: PRINT CHR$(179);
1510 NEXT I
1520 FOR J = C1 + 1 TO C2 - 1
1530  LOCATE R1, J: PRINT CHR$(196);
1540  LOCATE R2, J: PRINT CHR$(196);
1550 NEXT J
1560  LOCATE R1, C1: PRINT CHR$(218);
1570  LOCATE R1, C2: PRINT CHR$(191);
1580  LOCATE R2, C1: PRINT CHR$(192);
1590  LOCATE R2, C2: PRINT CHR$(217);
1600 COLOR W
1610 RETURN
1620 REM ask user to press a key to continue
1630 LOCATE 25, 1
1640 PRINT "Have Program Diskette in place, then press any key to
continue.";
1650 A$ = INKEY$: IF A$ = "" THEN 1650
1660 KEY ON: CLS : KEY OFF
1670 REM MENU Program Starts Here.
1680 REM Draw the Menu itself.
1690 REM Draw the Outer Double Box.
1700 R1 = 1: C1 = 1: R2 = 23: C2 = 79: GOSUB 1300
```

SECB

```
630 DATA39,**
640 PCLEAR1:WIDTH80:CLS6:POKE282,0:POKE150,41
650 ONERRGOTO1150:ONBRKGOTO1280
660 IFPEEK(&HA282)=23THEN700
670 IFPEEK(&HC004)=215THEN690
680 POKE&HD6CD,0:POKE&HD723,20:GOTO700
690 POKE&HD7C0,0:POKE&HD816,20
700 CLS6:LOCATE34,4:PRINT"BOOT Program"
710 LOCATE28,6:PRINT"The CoCo Family Recorder"
720 LOCATE18,8:PRINT"Copyright (c) January 1993 by F.G. Swygert"
730 LOCATE28,10:PRINT"** ALL RIGHTS RESERVED **"
740 LOCATE3,12:PRINT"(Based on Genealogy ON DISPLAY, Copyright
February 1986, by Melvin O. Duke"
750 LOCATE11,13:PRINT"for IBM compatible computers with GW-BASIC
or IBM BASIC-A)"
760 IFPEEK(&HA282)=23THEN770ELSE790
770 SOUND150,4:LOCATE14,16:PRINT"Currently set up for 2400 baud
printer.":LOCATE14,17:INPUT"Do you wish to change this? (Y/N)";A$
780 IFA$="y"ORA$="Y"THEN970:IFA$="n"ORA$="N"THEN1090:
GOTO770
790 SOUND150,4:LOCATE14,16:PRINT"Currently set up for 2400 baud
printer and 6ms drive"
800 LOCATE14,17:INPUT"step rate. Do you wish to change this? (Y/N)";A$
810 IFA$="y"ORA$="Y"THEN840
820 IFA$="n"ORA$="N"THEN1110
830 GOTO800
840 CLS6:LOCATE6,4:PRINT"Select Drive Step Rate:":PRINT
850 PRINTTAB(8)"1 = 6ms    2 = 12ms    3 = 20ms    4 = 30ms"
860 PRINT:LOCATE6,8:INPUT"Selected Value";D$
870 IFPEEK(&HC004)=215THEN880ELSE920
880 IFD$="1"THENPOKE&HD7C0,0:POKE&HD816,20:GOTO980
890 IFD$="2"THENPOKE&HD7C0,0:POKE&HD816,21:GOTO980
900 IFD$="3"THENPOKE&HD7C0,0:POKE&HD816,22:GOTO980
910 IFD$="4"THENPOKE&HD7C0,0:POKE&HD816,23:GOTO980
920 IFD$="1"THENPOKE&HD6CD,0:POKE&HD723,20:GOTO980
930 IFD$="2"THENPOKE&HD6CD,0:POKE&HD723,21:GOTO980
940 IFD$="3"THENPOKE&HD6CD,0:POKE&HD723,22:GOTO980
950 IFD$="4"THENPOKE&HD6CD,0:POKE&HD723,23:GOTO980
960 GOTO840
970 CLS6
980 LOCATE6,10:PRINT"Select Printer Baud Rate:":PRINT
990 PRINTTAB(8)"1 = 600    2 = 1200    3 = 2400    4 = 4800    5 =
9600"
1000 LOCATE8,14:A=PEEK(65314):B=INT(A/2):C=A/2:IFC-
B=0THENPRINT"** PRINTER IS ON **"ELSEPRINT"** PRINTER IS OFF
**"
1010 LOCATE6,16:PRINT"Program will lock-up if you try to print without
printer on and online."
1020 PRINT:LOCATE6,18:INPUT"Selected Value";R$
1030 IFR$="1"THENPOKE150,180:GOTO1090
1040 IFR$="2"THENPOKE150,87:GOTO1090
1050 IFR$="3"THENPOKE150,41:GOTO1090
1060 IFR$="4"THENPOKE150,18:GOTO1090
1070 IFR$="5"THENPOKE150,4:GOTO1090
1080 GOTO980
1090 LOCATE20,23:PRINT"PRESS ANY KEY TO ACCEPT, BREAK TO
RESTART";:EXEC44539
1100 IFPEEK(&HA282)=23THEN1140
1110 IFPEEK(&HC004)=214THENA$="C0EED52AD6D1D6F1D727D57E"
ELSEA$="C101D617D7C4D7E4D81AD851"
1111 FORV=1TO24STEP4
1112 A=VAL("&H"+MID$(A$,V,4))
1113 POKEA,189:POKEA+1,240:POKEA+2,157:POKEA+3,18
1114 NEXT
1116 POKE&HF09D,52:POKE&HF09E,127
1120 POKE&HF09F,53:POKE&HF0A0,255
1140 LOADCHR$(130)+CHR$(32)+CHR$(130)+CHR$(03)+"/" +CHR$
(03)+CHR$(12)+CHR$(13),R
1150 CLS2
1160 LOCATE26,8:PRINT"Error Number ";ERNO;" Has Occured"
1170 LOCATE26,8:PRINT"Error Number ";ERNO;" Has Occured"
1180 LOCATE18,10:PRINT"ERROR NUMBERS:"
1190 LOCATE18,11:PRINT"3 = Out of Data       6 = Out of Memory"
1200 LOCATE18,12:PRINT"27 = Bad Record Number   17 = Bad File
Data"
```

FARNA *Systems*
Your most complete source for Color Computer and OS-9 information!

ADD $3 S&H, $4 CANADA, $10 OVERSEAS

BOOKS:

Mastering OS-9 - $30.00
Completely steps one through learning all aspects of OS-9 on the Color Computer. Easy to follow instructions and tutorials. With a disk full of added utilities and software!
SALE $20 ONLY 5 DEC THRU

Tandy's Little Wonder - $25.00
History, tech info, hacks, schematics, repairs,... almost EVERYTHING available for the Color Computer! A MUST HAVE for ALL CoCo aficionados, both new and old!!! This is an invaluable resource for those trying to keep the CoCo alive or get back into using it.

Quick Reference Guides
Handy little books contain the most referenced info in easy to find format. Size makes them unobtrusive on your desk. Command syntax, error codes, system calls, etc.
CoCo OS-9 Level II : $5.00
OS-9/68000 : $7.00

Complete Disto Schematic set: $15
Complete set of all Disto product schematics. Great to have... needed for repairs!

CHECK OUT THE NEW LOW PRICES ON NITRO PRODUCTS!

SOFTWARE:

CoCo Family Recorder: Best genealogy record keeper EVER for the CoCo! Requires CoCo3, two drives (40 track for OS-9) and 80 cols.
DECB: $15.00   OS-9: $20.00

DigiTech Pro: $10.00
Add sounds to your BASIC and M/L programs! Very easy to use. User must make simple cable for sound input through joystick port. Requires CoCo3, DECB, 512K.

ADOS: Best ever enhancement for DECB! Double sided drives, 40/80 tracks, fast formats, extra and enhanced commands!
Original (CoCo 1/2/3) : **$10.00**
ADOS 3 (CoCo 3 only) : **$20.00**
Extended ADOS 3 (CoCo 3 only, requires ADOS 3, support for 512K-2MB, RAM drives, 40/80 track drives mixed) : **$30.00**
ADOS 3/EADOS 3 Combo: $40.00

Pixel Blaster - $12.00
High speed graphics tools for CoCo 3 OS-9 Level II. Easily speed up performance of your graphics programs! Designed especially for game programmers!

Patch OS-9 - $7.00
Latest versions of all popular utils and new commands with complete documentation. Auto-installer requires 2 40T DS drives (one may be larger).

TuneUp : $10.00
Don't have a 6309? You can still take advantage of Nitro software technology! Many OS-9 Level II modules rewritten for improved speed with the stock 6809!

Thexder OS-9
Shanghai OS-9 : $10.00 each
Transfer your ROM Pack game code to an OS-9 disk! Requires ownership of original ROM pack.

Rusty : $10.00
Launch DECB programs from OS-9! Load DECB programs from OS-9 hard drive!

NitrOS-9:
Nitro speeds up OS-9 from 20-50% depending on the system calls used. This is accomplished by completely rewriting OS-9 to use all the added features of the Hitachi 6309 processor. Many routines were streamlined on top of the added functions! The fastest thing for the CoCo3! Easy install script! 6309 required.
Level 3 adds even more versatility to Nitro! RBF and SCF file managers are given separate blocks of memory then switched in and out as needed. Adds 16K to system RAM... great for adding many devices!
NitrOS-9 V.2.0: $10.00
NitrOS-9 Level 3: $10.00

The AT306 OS-9 Single Board Computer

AT306 Motherboard Specs:
16 bit PC/AT I/O Bus (three slots)
MC68306 CPU at 16.67MHz
Four 30 Pin SIMM Sockets
IDE Hard Drive Interface
Floppy Drive Interface (180K-2.88M)
Two 16 byte Fast Serial Ports (up to 115K baud)
Two "Terminal" Serial Ports (no modem)
Bidirectional Parallel Port
Real-time clock
PC/AT Keyboard Controller (five pin DIN)

Included Software Package:
"Personal" OS-9/68000 Vr 3.0
        (Industrial with RBF)
MGR Graphical Windowing Environment
        with full documentation
Drivers for Tseng W32i
        and Trident 8900 VGA cards
Drivers for Future Domain 1680
        and Adaptec AAH15xx SCSI cards
Many PD and customized utilities and tools

The AT306 is a fully integrated single board computer. It is designed to use standard PC/AT type components. Sized the same as a "Baby AT" board (approximately 8" square). Compact and inexpensive enough to be used as an embedded controller! Use with a terminal (or terminal emulation software on another computer) or with a video card as a console system. Basic OS-9 drivers are in ROM, making the system easy to get started with.

> **HACKERS MINI KIT: Includes AT306 board, OS-9 and drivers, util software, assembly instructions/tips, T8900 1MB video card. Add your own case, keyboard, drives, and monitor!   ONLY $550!**

Call for a quote on turn-key systems and quantity pricing.
Warranty is 90 days for labor & setup, components limited to manufacturers warranty.

Microware Programmers Package -
Licensed copies of Microware C compiler, Assembler, Debugger, and many other tools!
With system purchase: $65.00 Without system: $85.00

GW-BASIC

```
1710 REM Draw the Heading Separator.
1720 R1 = 3: C1 = 1: R2 = 3: C2 = 79: GOSUB 2780
1730 REM Draw the Vertical Separators.
1740 R1 = 1: C1 = 6: R2 = 23: C2 = 6: GOSUB 2870
1750 R1 = 1: C1 = 17: R2 = 23: C2 = 17: GOSUB 2870
1760 REM Attach the intersections
1770 COLOR P
1780 LOCATE 3, 6: PRINT CHR$(197);
1790 LOCATE 3, 17: PRINT CHR$(197);
1800 COLOR W, K
1810 REM Print the content of the menu.
1820 COLOR K, W
1830 LOCATE 2, 3: PRINT "No";
1840 LOCATE 2, 8: PRINT "Name"
1850 LOCATE 2, 19: PRINT "Function of the Program"
1860 COLOR W, K
1870 COLOR K, W: LOCATE 4, 3: PRINT " 1"; : COLOR R, K
1880 LOCATE 4, 8: PRINT "CREATPER";
1890 LOCATE 4, 19: PRINT "Creates (FORMATS) a Persons File.";
1900 COLOR K, W: LOCATE 5, 3: PRINT " 2"; : COLOR R, K
1910 LOCATE 5, 8: PRINT "CREATMAR";
1920 LOCATE 5, 19: PRINT "Creates (FORMATS) a Marriages File.";
1930 IF DD.ORD$ = "no" THEN 1970
1940 COLOR K, W: LOCATE 6, 3: PRINT " 3"; : COLOR R, K
1950 LOCATE 6, 8: PRINT "CREATORD";
1960 LOCATE 6, 19: PRINT "Creates (FORMATS) an Ordinances File.";
1970 COLOR K, W: LOCATE 7, 3: PRINT " 4"; : COLOR G, K
1980 LOCATE 7, 8: PRINT "UPDATPER";
1990 LOCATE 7, 19: PRINT "Updates Information in the Persons File."
2000 COLOR K, W: LOCATE 8, 3: PRINT " 5"; : COLOR G, K
2010 LOCATE 8, 8: PRINT "UPDATMAR";
2020 LOCATE 8, 19: PRINT "Updates Information in the Marriages File."
2030 IF DD.ORD$ = "no" THEN 2070
2040 COLOR K, W: LOCATE 9, 3: PRINT " 6"; : COLOR G, K
2050 LOCATE 9, 8: PRINT "UPDATORD";
2060 LOCATE 9, 19: PRINT "Updates Information in the Ordinances File."
2070 COLOR K, W: LOCATE 10, 3: PRINT " 7"; : COLOR R, K
2080 LOCATE 10, 8: PRINT "INDEXPC ";
2090 LOCATE 10, 19: PRINT "Prepares a Parent/Child Index. (For 13,
16, 18 and 19.)";
2100 COLOR K, W: LOCATE 11, 3: PRINT " 8"; : COLOR R, K
2110 LOCATE 11, 8: PRINT "INDEXMAR";
2120 LOCATE 11, 19: PRINT "Prepares a Marriages Index. (For 15, 16,
17, 18 and 19.)";
2130 COLOR K, W: LOCATE 12, 3: PRINT " 9"; : COLOR O, K
2140 LOCATE 12, 8: PRINT "PRINTPER";
2150 LOCATE 12, 19: PRINT "Prints Detailed Information about
Persons.";
2160 COLOR K, W: LOCATE 13, 3: PRINT "10"; : COLOR O, K
2170 LOCATE 13, 8: PRINT "PRINTMAR";
2180 LOCATE 13, 19: PRINT "Prints Detailed Information about
Marriages."
2190 COLOR K, W: LOCATE 14, 3: PRINT "11"; : COLOR O, K
2200 LOCATE 14, 8: PRINT "LISTPER ";
2210 LOCATE 14, 19: PRINT "Prints a List of the Persons in the Persons
File."
2220 COLOR K, W: LOCATE 15, 3: PRINT "12"; : COLOR O, K
2230 LOCATE 15, 8: PRINT "LISTMAR ";
2240 LOCATE 15, 19: PRINT "Prints a List of the Marriages in the
Marriages File."
2250 COLOR K, W: LOCATE 16, 3: PRINT "13"; : COLOR O, K
2260 LOCATE 16, 8: PRINT "LISTPCI ";
2270 LOCATE 16, 19: PRINT "Prints a List of the Parent/Child Index.";
2280 COLOR K, W: LOCATE 17, 3: PRINT "14"; : COLOR O, K
2290 LOCATE 17, 8: PRINT "ALPHAPER";
2300 LOCATE 17, 19: PRINT "Prints an Alphabetical List of Persons."
2310 COLOR K, W: LOCATE 18, 3: PRINT "15"; : COLOR O, K
2320 LOCATE 18, 8: PRINT "ALPHAMAR";
2330 LOCATE 18, 19: PRINT "Prints an Alphabetical List of Marriages."
2340 COLOR K, W: LOCATE 19, 3: PRINT "16"; : COLOR B, K
2350 LOCATE 19, 8: PRINT "DISPLAY ";
2360 LOCATE 19, 19: PRINT "Displays Genealogical Information on the
Screen."
2370 COLOR K, W: LOCATE 20, 3: PRINT "17"; : COLOR G, K
2380 LOCATE 20, 8: PRINT "PEDIGREE";
```

SECB

```
1210 LOCATE18,13:PRINT"20 = I/O Error        21 = Bad File Mode"
1220 LOCATE18,14:PRINT"25 = Disk Full        26 = File Not Found"
1230 LOCATE16,16:PRINT"<<** Check for correct disk(s) in drive(s)!
**>>"
1240 LOCATE25,18:PRINT"Place Program Disk in Drive 0"
1250 SOUND150,4:LOCATE20,20:PRINT"PRESS ANY KEY TO END,.
BREAK TO TRY AGAIN"
1260 EXEC44539
1270 CLOSE:POKE113,0:EXEC35867
1280 CLS2
1290 LOCATE20,12:INPUT"Do you wish to halt the program, Y or N";R$
1300 IFR$="Y"ORR$="y"THEN1270
1310 IFR$="N"ORR$="n"THEN1330
1320 GOTO1290
1330 CLOSE:CLS6:GOTO640

10 REM **** MENU Program
20 REM Copyright (c) by F.G. Swygert, September 1991
30 IFPEEK(269)*256+PEEK(270)<>32401THENGOTO1040
40 PCLEAR1:WIDTH80:CLS6:POKE282,0
50 SOUND150,4:SOUND150,4:SOUND100,4:SOUND100,4:SOUND
150,4:SOUND150,4
60 ONERRGOTO710:ONBRKGOTO840
70 LOCATE28,10:PRINT"The CoCo Family Recorder"
80 LOCATE18,12:PRINT"Copyright (c) September 1991 by F.G. Swygert"
90 LOCATE28,14:PRINT"** ALL RIGHTS RESERVED **"
100 LOCATE3,16:PRINT"(Based on Genealogy ON DISPLAY, Copyright
February 1986, by Melvin O. Duke"
110 LOCATE11,17:PRINT"for IBM compatible computers with GW-BASIC or
IBM BASIC-A)"
120 SOUND150,4:LOCATE27,21:PRINT"Press any key to continue"
130 EXEC44539
135 CLS6:ONERRGOTO710:ONBRKGOTO840
140 PRINT:PRINT:PRINT"        The CoCo Family Recorder Version
1.0":PRINT
150 PRINT" No  Name      Function"
160 PRINT" 1  CREATPER  Creates (FORMATS) a Persons File."
170 PRINT" 2  CREATMAR  Creates (FORMATS) a Marriages File "
180 PRINT" 3  UPDATPER  Updates Information in the Persons File."
190 PRINT" 4  UPDATMAR  Updates Information in the Marriages File."
200 PRINT" 5  INDEXPC   Prepares a Parent/Child Index. (For 11 and 12)"
210 PRINT" 6  INDEXMAR  Prepares a Marriages Index. (For 12)"
220 PRINT" 7  PRINTPER  Prints Detailed Information about Persons."
230 PRINT" 8  PRINTMAR  Prints Detailed Information about Marriages."
240 PRINT" 9  LISTPER   Prints a List of the Persons in the Persons File."
250 PRINT" 10  LISTMAR  Prints a List of the Marriages in the Marriages
File."
260 PRINT" 11  LISTPCI  Prints a List of the Parent/Child Index."
270 PRINT" 12  DISPLAY  Displays Genealogical Information on the
Screen."
280 PRINT" 13  WORKSHT  Prints information gathering worksheet."
290 PRINT:PRINT"      Press CTRL-F1 to print screen in any program!"
300 IF(PEEK(65314)AND1)=1THENSOUND150,4:SOUND150,4:PRINT"
(PRINTER NOT READY!)"
310 B=PEEK(150)
320 IFB=180THEND=600
330 IFB=87THEND=1200
340 IFB=41THEND=2400
350 IFB=18THEND=4800
360 IFB=4THEND=9600
370 PRINT"        Printer Baud Rate :";D:PRINT
380 PRINT" Type a Program Number, and press the ENTER key."
390 INPUT"  (Q to quit, 14 to change baud rate, 15 to restart)";R$
400 IFR$="q"ORR$="Q"THEN590
410 R=VAL(R$)
420 IFR<1ORR>15THEN135
430 IFR=1THENLOAD"CREATPER:0",R
440 IFR=2THENLOAD"CREATMAR:0",R
450 IFR=3THENLOAD"UPDATPER:0",R
460 IFR=4THENLOAD"UPDATMAR:0",R
470 IFR=5THENLOAD"INDEXPC:0",R
480 IFR=6THENLOAD"INDEXMAR:0",R
490 IFR=7THENLOAD"PRINTPER:0",R
500 IFR=8THENLOAD"PRINTMAR:0",R
510 IFR=9THENLOAD"LISTPER:0",R
```

```
2390 LOCATE 20, 19: PRINT "Prints Pedigree Charts (Family Trees)."
2400 COLOR K, W: LOCATE 21, 3: PRINT "18"; : COLOR G, K
2410 LOCATE 21, 8: PRINT "FAMILY ";
2420 LOCATE 21, 19: PRINT "Prints Family Group Sheets."
2430 COLOR K, W: LOCATE 22, 3: PRINT "19"; : COLOR B, K
2440 LOCATE 22, 8: PRINT "DESCEND";
2450 LOCATE 22, 19: PRINT "Displays (and Optionally Prints) Descen-
dents Charts."
2460 COLOR W, K
2470 REM Now obtain User Response
2480 LOCATE 25, 2: PRINT "(0 to quit, 20 to restart the MENU)";
2490 LOCATE 24, 1: INPUT "Type a Program Number, and press the
'enter' key."; REPLY$
2500 IF REPLY$ = "" THEN 1660
2510 IF REPLY$ = "0" THEN 2960
2520 REPLY = VAL(REPLY$)
2530 IF REPLY < 1 OR REPLY > 20 THEN 1660
2540 IF REPLY = 1 THEN KEY ON : CHAIN DD.PROG$+"creatper",,ALL
2550 IF REPLY = 2 THEN KEY ON : CHAIN DD.PROG$+"creatmar",,ALL
2560 IF DD.ORD$ = "no" THEN 2580
2570 IF REPLY = 3 THEN KEY ON : CHAIN DD.PROG$+"creatord",,ALL
2580 IF REPLY = 4 THEN KEY ON : CHAIN DD.PROG$+"updatper",,ALL
2590 IF REPLY = 5 THEN KEY ON : CHAIN DD.PROG$+"updatmar",,ALL
2600 IF DD.ORD$ = "no" THEN 2620
2610 IF REPLY = 6 THEN KEY ON : CHAIN DD.PROG$+"updatord",,ALL
2620 IF REPLY = 7 THEN KEY ON : CHAIN DD.PROG$+"indexpc",,ALL
2630 IF REPLY = 8 THEN KEY ON : CHAIN DD.PROG$+"indexmar",,ALL
2640 IF REPLY = 9 THEN KEY ON : CHAIN DD.PROG$+"printper",,ALL
2650 IF REPLY = 10 THEN KEY ON : CHAIN DD.PROG$+"printmar",,ALL
2660 IF REPLY = 11 THEN KEY ON : CHAIN DD.PROG$+"listper",,ALL
2670 IF REPLY = 12 THEN KEY ON : CHAIN DD.PROG$+"listmar",,ALL
2680 IF REPLY = 13 THEN KEY ON : CHAIN DD.PROG$+"listpci",,ALL
2690 IF REPLY = 14 THEN KEY ON : CHAIN DD.PROG$+"alphaper",,ALL
2700 IF REPLY = 15 THEN KEY ON : CHAIN
DD.PROG$+"alphamar",,ALL
2710 IF REPLY = 16 THEN KEY ON : CHAIN DD.PROG$+"display",,ALL
2720 IF REPLY = 17 THEN KEY ON : CHAIN DD.PROG$+"pedigree",,ALL
2730 IF REPLY = 18 THEN KEY ON : CHAIN DD.PROG$+"family",,ALL
2740 IF REPLY = 19 THEN KEY ON : CHAIN DD.PROG$+"descend"
,,ALL
2750 IF REPLY = 20 THEN KEY ON: RUN DD.MENU$ + "menu"
2760 REM Improper Response
2770 GOTO 1660
2780 REM Subroutine to draw a single horizontal line. Attach to double.
2790 COLOR P
2800 FOR J = C1 + 1 TO C2 - 1
2810  LOCATE R1, J: PRINT CHR$(196);
2820 NEXT J
2830 LOCATE R1, C1: PRINT CHR$(199);
2840 LOCATE R1, C2: PRINT CHR$(182);
2850 COLOR W
2860 RETURN
2870 REM Subroutine to draw a single vertical line. Attach to double.
2880 COLOR P
2890 FOR I = R1 + 1 TO R2 - 1
2900  LOCATE I, C1: PRINT CHR$(179);
2910 NEXT I
2920 LOCATE R1, C1: PRINT CHR$(209);
2930 LOCATE R2, C1: PRINT CHR$(207);
2940 COLOR W
2950 RETURN
2960 KEY ON: CLS : KEY OFF: LOCATE 21, 1
2970 PRINT "End of Program"
2980 END
```

```
520 IFR=10THENLOAD"LISTMAR:0",R
530 IFR=11THENLOAD"LISTPCI:0",R
540 IFR=12THENLOAD"DISPLAY:0",R
550 IFR=13THENLOAD"WORKSHT:0",R
560 IFR=14THEN910
570 IFR=15THENLOAD"BOOT:0",R
580 GOTO135
590 CLS2
600 LOCATE21,12:INPUT"Do you wish to end the program, Y or N";Q$
610 IFQ$="Y"ORQ$="y"THEN640
620 IFQ$="N"ORQ$="n"THEN700
630 GOTO600
640 CLS1:CLOSE:LOCATE14,8:PRINT"Don't forget to make NEW
backups of your DATA Disks!"
670 FORX=1TO1000:NEXTX
690 POKE65496,0:POKE113,0:EXEC35867
700 GOTO135
710 CLS2
720 LOCATE26,8:PRINT"Error Number ";ERNO;" Has Occured"
730 LOCATE26,8:PRINT"Error Number ";ERNO;" Has Occured"
740 LOCATE18,10:PRINT"ERROR NUMBERS:"
750 LOCATE18,11:PRINT"3 = Out of Data        6 = Out of Memory"
760 LOCATE18,12:PRINT"27 = Bad Record Number   17 = Bad File Data"
770 LOCATE18,13:PRINT"20 = I/O Error        21 = Bad File Mode"
780 LOCATE18,14:PRINT"25 = Disk Full        26 = File Not Found"
790 LOCATE16,16:PRINT"<<** Check for correct disk(s) in drive(s)! **>>"
800 LOCATE25,18:PRINT"Place Program Disk in Drive 0"
810 SOUND150,4:LOCATE15,20:PRINT"PRESS ANY KEY TO RETURN
TO MENU, BREAK TO TRY AGAIN"
820 EXEC44539
830 CLS6:GOTO135
840 CLS2
850 LOCATE20,12:INPUT"Do you wish to halt the program, Y or N";R$
860 IFR$="Y"ORR$="y"THEN890
870 IFR$="N"ORR$="n"THEN900
880 GOTO850
890 CLOSE:GOTO590
900 CLOSE:CLS6:CLEAR500:GOTO135
910 CLS6:LOCATE27,8:PRINT"Select Printer Baud Rate:":PRINT:PRINT
920 PRINT"    1 = 600    2 = 1200   3 = 2400   4 = 4800   5 = 9600"
930 LOCATE27,16:A=PEEK(65314):B=INT(A/2):C=A/2:IFC-
B=0THENPRINT"PRINTER IS ON"ELSEPRINT"PRINTER IS OFF"
940 LOCATE3,18:PRINT"Program will lock-up if you try to print without
printer on and online."
950 PRINT:LOCATE27,13:INPUT"Selected Value";R$
960 IFR$="1"THENPOKE150,180:GOTO1020
970 IFR$="2"THENPOKE150,87:GOTO1020
980 IFR$="3"THENPOKE150,41:GOTO1020
990 IFR$="4"THENPOKE150,18:GOTO1020
1000 IFR$="5"THENPOKE150,4:GOTO1020
1010 GOTO910
1020 GOTO135
1040 CLOSE:POKE113,0:EXEC35867
```

Program Notes:
It would be very helpful to pick up a BASIC reference for GW-BASIC (almost any MS-DOS BASIC reference, except QBASIC and later, should do). But for those who don't have one, here are some of the GW commands used that you may not be familiar with AND that caused trouble:

CHAIN - Calls a program and passes variables to it from the current program. SDECB doesn't allow passing of variables. That is one reason some setup was accomplished using a separate program. Rather than specifying the number of records desired and passing the variable to each sub program, the maximum number was used and the variable permanently set in each SDECB subprogram. One should be able to create a data file and store golbal variables, modifying the variables with one program and having the other programs read this file upon starting. I took the simplest route.

The subroutines in the GW program at line 1300 and 1460 draw boxes using the IBM extended character set.

micro notes

Motorola's Blackbird set-top box
See the article about Motorola's new Blackbird set-top box. It runs OS-9 (DAVID) and also hooks it into Project X from VM Labs, which you may have seen in WIRED and other publications as the next biggest thing to happen.

http://www.eetimes.com/news/98/1024news/blackbird.html

An excerpt from the article for those without web access:

BEGIN QUOTE: *Although Motorola declined to describe Blackbird in advance of its planned Sept. 12 launch, sources who have been closely working with Motorola on the project said the company is betting big on the success of what they said will be a highly flexible platform. "Considering some industry forecasts showing an Internet set-top and a game platform as an ideal combination, this [Blackbird] strategy does make sense," said Abhishek Gami, an analyst at William Blair & Co. (Chicago), an investment-banking firm.*

At the heart of Blackbird is the Project X media processor from startup VM Labs. Blackbird uses it to decode digital audio and video streams, to process graphics and as a main processor in configurations for standalone DVD or game players. Versions of Motorola's PowerPC will be used in some high-end configurations.

The Blackbird platform stems from the mid-'90s, when Motorola, Microware and a forerunner of VM Labs jointly bid on a request for proposals (RFP) issued by Tele-TV.

That ambitious interactive TV joint venture — now defunct — comprised Bell Atlantic, Nynex and Pacific Telesis. Industry sources close to Motorola said the whole NIM concept for the Blackbird platform came from the Tele-TV RFP. END QUOTE.

(editors note: Remember the hype on using DAVID and OS-9 for set-top boxes just a couple years ago?)

I was suprised when I saw this article -- I'd been following Project X (the one that does real time raytracing and other things and plans to be the heart of high performance low cost DVD players and game machines), and seeing it linked with OS-9 makes me feel fuzzy inside.
- Allen Huffmen

Microware Positions Available!
Microware Systems Corporation has the following positions available. All require a minimum of a BS in Computer Science or related field. Entry and Senior level positions are available.
Vice President, Professional Services
Manager - Consulting
Product Manager - Consulting
Product Manager - Marketing
Software Engineer - R&D Tools
Support Engineer - Training and Support
For more information, contact:
Human Resources Administrator
Microware Systems Corporation
1500 N. W. 118th Street
Des Moines, IA 50325
Fax: (515) 327-5528
E-mail: recruiting@microware.com

Order Projector 3 On-Line!
Projector-3 views and converts all CoCo picture formats plus GIF, Atari, Macintosh, Amiga, and more! Superbly displays graphics normally not viewable by the CoCo, and allows converting ANY rendered image to many other formats, making P-3 an aid in desktop publishing and other applications. Renderings can also be edited and printed. The url for ordering and information is:

https://www.regnow.com/softsell/nph-softsell.cgi?item=1583-1

Delivery is done through a vendor generated registration code that will be sent to you via email.
-Brian Tietz

The CoCoFest Chronicles
Allen Huffman's typeset and metal spiral bound book, CoCoFest Chronicles, contained updated versions of his reports on all the post-Rainbow CoCoFests, from the 1990 Atlanta Fest on up to the 1997 Pennsylvania Fest. Special introductions by people such as Ken Fish, Tony Podraza, James Jones, Carl Boll, Brother Jeremy, Tim Johns, Mike Knudsen, and others. New behind the scenes commentary explaining many "inside" items that never made the released reports. This is possibly one of the last major efforts to document Tandy Color Computer history.

https://www.regnow.com/softsell/nph-softsell.cgi?item=1567-1

The product will be shipped to you within one week of ordering. If you don't have web access, send $29.95 (postpaid in US) to:
Allen C. Huffman
P.O. Box 22031
Des Moines, IA 50325
-Allen Huffmen

BlackHawk Enterprises online
BlackHawk Enterprises, Inc. is pleased to announce the opening of their very website. This site on the Internet will host all current information on BlackHawk products as soon as it becomes available. Also there is a very intensive 'Links' section, where web surfers can find information on such topics as OS-9 related:
Companies
User Groups
User Homepages
FTP sites
Magazine Information
Books
Newsgroups
Email Lists
There is also current priceing on software/hardware and if there is enough interest there may soon be a way of securely ordering products online!

http://www.home.pon.net/kf6ntg/blackhawk

- Brian Tietz

The World of 68' Micros online
Farna Systems is also happy to announce the opening of our recently renovated web site.
This site will sport new features such as:
Monthly News
Subscription Info.
Back Issue Info.
How to contribute to 68' micros online
Contact 68 micros
The url for our main page is:
http://www.home.pon.net/kf6ntg/68micros

operating system nine
Getting the OS-9 Attitude

Author Unknown & Rick Ulland

Introduction

This is a limited peek into a powerful, complex system. It is not intended as a "how to" manual, rather to help you "get the feel". Some specifics are given and every attempt has been made to be accurate in these cases.)

Getting the Attitude...

From the very outset I will make it clear that, as it comes, OS9 for CoCo is a rather crippled system because of its packaging and lack of support from the producer. There are several things you can do, as soon as you are able, that will make it vastly superior to what it is in its original form.

First, we have Level 2 on a CoCo3. There is no comparison between a 32 or 40 column screen and an 80 column screen, but since Tandy doesn't know what kind of monitor or disk drive(s) you may have, OS9 arrives in a low resolution package and on single sided, 35 track disks. Personally, I wouldn't even consider using it that way. It is like having a powerful new car with special full time brakes installed and all but one window painted black.

So, you have an 80 column monitor; how do you get OS9 to use it? There is an article by Rick Ulland entitled "The First Step" that will be included in this series. It tells you how to switch to 80 columns, double sided 40 track disks and high speed drive stepping. It is a fairly long tedious procedure, but when it's done, you'll be able to see what you're doing and you'll have more than double the disk space.

Another real aid in using OS9 is in the Delphi System Modules database (also found on RTSI FTP site -- ftp:os9archive.rtsi.com). It is written by OS9 programmer Kevin Darling and is called SCF EDITOR PLUS - LEVEL TWO. Again it is a task and a half to install, but is WELL worth it. It is a command line editor that makes life in the OS9 world of long command lines greatly easier, especially if you're not an expert typist. If you do not have a friend who can install this feature for you the guys on the SIG will help you through it like they did me. I will talk more about this later.

There are many other improvements and additions to OS9 that you will get as you go along, but these are tops in my opinion. Next in line would be the new shell, SHELL+, with several improvements. The sooner you can arrive at an updated system the better off you'll be. (editor's note: all of these improvements are included in the available packaged enhancements, such as "Patch OS-9", "Tune-Up", and "Nitro" already. These were designed to make the task of updating OS9 less daunting.)

Coming to OS9 from CoCo Disk Extended Color BASIC involves something besides the software and dreams of owning a hard drive - if you don't already have one. No doubt you've heard about the power and versatility of OS9. It is, indeed, several steps ahead of DECB in many respects. But the change in environments can be almost overwhelming if you don't grasp a few essentials - a few of the differences between DECB and OS9. You can't maintain a DECB attitude while learning OS9.

DECB is a BASIC language with some of the power of a DOS (disk operating system). OS9 is a disk operating system with some of the power of a language thrown in. This much is mundane. But getting into the actual differences is both interesting and essential.

A simple concept that sheds light on the move to OS9 can be illustrated this way. If you had all the money you wanted, but all audio/video systems were alike, it would take you minutes to buy a complete system. But with all the hundreds or thousands of TVs, CDs, amps, tuners, speakers, stereos, tape players, recorders, multi media systems, etc., etc., in the equation, it is a major job deciding on an intelligent system purchase.

The more options an operating system has, the more decisions you must make, the more you must remember, and the more occasions you have to make mistakes. What might involve 3 factors in a simple system might involve 10 in a more powerful system. The 3 would give you 9 paths to choose from. The 10 would give you 100 choices - for starters.

The importance of this point can't be over estimated. And it directly relates to one of the things that causes the most trouble and consternation to new OS9 users - multiple directory disk organization. That will be our starting point and our prime consideration.

For all practical purposes an OS9 system might as well have 6 or 8 (or more) different drives. You can imagine how much you could do by organizing them all in creative order! But you can also imagine the headaches of remembering which drive had what on it if they weren't organized. This dilemma would ideally be solved by arranging them in a logical heirarchical order.

First you would divide everything up into files that do work and files that are worked on - executable files and data files. If you have a great many executable files you might have to divide them into 2 or more sets. The data files must be divided into many more sets and subsets. There are data files like messages, there are tables, lists, subroutines, logs, graphics, music, controls, etc. Some files even defy catagorizing!

Which drives would get which files? Remember, we're talking about much power and large numbers of files! And, to make it interesting most of the commands you would use like "deldir" aren't usually in memory, but on disk! To get all the power, THERE ARE SO MANY commands they'd choke the computer if you put them all in memory at the same time! And you'd go crazy looking for the right one on the disk - except for the organization.

(Pay close attention to the use of words like "might" and "could". This is not a "how to" article. I will go out of my way to avoid setting down "how to" rules while hypothesizing.)

A level of organization is USUALLY, but not always (another variable) indicated by a "/" mark. Let's look at some levels.

Within the disk in a drive there might be a section of commands. Listed under commands (besides scores of things we usually think of as commands) might be WP, a word processor, and SS a spread sheet. Let's see what this starts to look like. Then I'll back up and make things better (and relieve those who just KNOW I'm doing this wrong.).
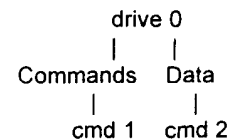
```
            drive 0
         |          |
     Commands     Data
         |          |
       cmd 1      cmd 2


You start with a "/" for the top level
                /
    You add a drive number
               /d0
```

Get Rid of STATIC

Rick Ulland

What causes the static, and how to clear it up.

Okay, if you've had a CoCo3 for any length of time you probably don't use a TV any more. But there are still lots of users out there who do, and I'm sure they would like a better screen. What causes this static?

When dealing with rf, there are all sorts of constraints, especially computers where the motherboard puts out roughly the same strength signal as the video by wire. What happens is the rf signal isn't a simple in to out, but bounces back and forth along the wire connecting reciever to antenna, or CoCo to TV.

A SWR (standing wave ratio) thats pretty low decribes a transmission line than's 'in tune'- imagine a water trough with waves.If it's the right length, the echo of a stale wave bouncing end to end will match up with the fresh wave moving through, and they actually help each other through the output. Partly.

In the real world, the transmission time through the cable (plus wire in TV up to input amp) is not an even multiple of the wavelength of the signal (closer isn't always better). Normally, this means slightly weak throughput but in computerdom some random noise from the motherboard couples with the errant echos (at the effective transmission line length) so noise becomes more than itself, even more than the driving signal. Really nice herringbone, if thats what you want to see. And I'm sure you've all noticed that the static intensifies with disk I/O -- that long unshielded

ribbon cable adds its own noise (unless you have an FD-502 with shielded cable - - it should produce less noise, though I doubt it produces none).

This RF interference can be greatly reduced by using a shielded RF cable long enough to roll a few times in a loop before it reaches the TV. Coils work because you randomize the wave pattern. Transformer effects hit both the noise and the signal (so much for sheilding), and at some point the wanted electrons minus damping overrides the crap minus damping for a nice picture.

If you had a really tiny microwatt SWR meter might do even better cutting cable. Pulling tiny coils around a ferrite toroid (ok, donut) makes 'em dozens of times smaller, small enuogh for a cute plastic cover....

I liked the "Clive solution" -- my old Sinclair ZX81 put out enough RF to cause sunburn (okay, I'm exagerrating greatly!), but supplied a rock steady picture way beyond what the 'so-much-better' Timex TS1000 (or CoCo, 99/4a, Commodore, etc.) could. It sent it's video on channel 36 uhf!

But then the FCC stepped in. They really had to -- computers of the late 70's and very early 80's put out enough RFI (radio frequency interference) that your neighbor in the next apartment (or even next house in some town neighborhoods) couldn't watch their own TV or listen to a clear radio braodcast. Dennis Kitsz (remember him?) got into computers with the old TRS-80 Model 1. He had his work room

covered with "chicken wire" and that grounded to the computer chassis, in effect shielding the entire room! That was done to reduce (but not eliminate) complaints from neighbors in his apartment complex.

There are simpler things to help clear the signal. The easiest is described above, since you already have a shielded cable that came with the computer -- loop it if long enough. This can be taken one step further by getting a better shielded cable -- the loops shouldn't be necessary then, though they may help.

A better cable is easy enough to find -- in fact, you may already have one! Get a cable TV patch cable and a female F plug to RCA adapter from good ol' Rat Shack (#278-252, $1.99;or gold plated #278-290, $2.49). If you don't have a cable ready TV, you'll also need a matching transformer (cable adpater, RS #15-1253, $2.99). TV leads flat cable) are 300 ohm, coax cable 75 ohm (as used for cable TV and video equipment). Most thin patch cables (RF cables) are no better than the original CoCo cable, no point in buying those! You need a good short length of RG-6 or better yet RG-6Q cable (RS#15-1541, 3 ft, $6.99; 15-1542, 6 ft, $7.99). Make sure it says "RG-6" on the cable or it is most likely the cheap RF cable.

---

operating system nine

You add another "/" for a level within d0
/d0/
You add the commands identifier
/d0/cmds
You add another level indicator
/d0/cmds/
Finally the name of the word processor
/d0/cmds/wp

In BASIC you might enter RUN"WP". In OS9 you might enter /d0/cmds/wp. A three level command. (Nothing magic about the "three". There can be more levels - or less.)

Let's do a tiny bit of computing - deleting WP from the disk. That will add another dimension to remember. If there were only one level you could enter "del wp." But the computer wants something like

this: del /d0/cmds/wp
The real beginners OS9 system usually starts with 2 drives at the top level of organization. So let's put a backup copy of WP on the other drive - drive 1. Let's see how complicated this can turn out:
copy /d0/cmds/wp /d1/cmds/wp
Already you can see something that is similar to DECB - spaces separate items within a statement. Something different is that both the word "copy" and the word "WP" are executable and considered commands on the disk. Some people separate these kinds of files making yet another thing to remember.

Now remember that you have tons of stuff on your disk - commands and data. The authors of OS9 came up with a way to cut down on a lot of the typing you need to do to slice through all the levels of organization. You can set the system to know what drive and/or level you're working with and whether it is executable or data. Voila,

the commands CHD (data) and CHX (executable).

(Let's switch to computer talk for these levels of organization. What they are called is directories and sub-directories. Like a general index for an encyclopedia then a different index for each book.)

If you enter "chx /d0/cmds" you set the executable commands pointer to that drive and directory. Then you can leave off that part from a typed command! Since WP is in the cmds directory, to get it you just enter "wp"! At this point you could delete it by entering "del wp." Simplifies things, doesn't it?

(Let's do some more computerese. After the command itself, a line like we've been using is called a pathlist. It indicates the path down through the levels of directories to a given item you're after. Remember - the effective name [pathlist] of a file includes the names of the drives and directories above the filename itself.)

These helpful commands (CHD and CHX) are perhaps even more time saving if you're dealing with data. For example, if you have to decompress several files that are located in the ARC directory, you could enter "chd /d1/arc". Then you could call all the files without the full command line and all the files you decompress (dearchive) would be stored in that same directory.

(Still more. When you set chx to a directory it is then called the "execution" directory and the system will look there for any command you use that doesn't have a pathlist preceding it.)

CHD and CHX are a lot like connecting to two single directory drives, one for data and one for commands. That simplifies things doesn't it? But are you thinking what I'm thinking? Chd and chx are also two more things to remember!

So what happens if you forget? You get an error!! If you set "chx /d1/cmds" and then call for a command that is on drive 0 you will get an "error 216" - pathname not found. That is like an NE error in DECB. But you can't just blame things on CHD or CHX, because the system will have to look in SOME specific place for what you command. And if you ask the wrong thing, you get the wrong thing.

Note that the error message uses "pathname", not "pathlist". Consider a pathname a segment (between slashes) of a pathlist. Let's look closer at errors. Forwarned is indeed forarmed.

Usually errors are the bane of the OS9 beginners existance. With so many things to remember it is easy to make multiple mistakes. And as with other systems the actual error number doesn't always tell you what's wrong - even if you look up the number.

But there is something you can keep in mind that will ease your recovery from such things with the least frustration and time. Knowing it in advance will at least allow you to analyze problems in an intelligent way. Remember that the file name is called "pathlist" because it includes the path of drives or directories through which the file is accessed.

Any part of the pathlist can give you the error - not just the file name at the end of it. Let me describe just one error problem. If the command you start a line with is not in the directory of executable commands that you indicate, you can get a 216 error there. If the drive you name, has the wrong disk in it, you can get an error from that part of the pathlist. If the directory you name is not on the indicated disk, that will give the same error, 216 - pathname not found. And, of course, if the file you want turns out not to be in the indicated directory, same thing. Of course, misspelling can effect any of these things. At least you now know to look in different places for trouble.

All this really involves a state of mind. To get at all that power you have to think differently (and more) with OS9 than you did in DECB.

Adventures in Assembly, Part 2

Art Flexser

Easy sorting using a Bubble Sort - in DECB and Assembly!

Here is another exercise in assembly language programming. It involves a routine to sort data in order of size. The technique to be used is called a BUBBLE SORT. It is not generally the fastest possible sorting algorithm, but it is the easiest and most straightforward to implement.

In this example, we are going to sort the bytes that make up the text display memory region—locations $400 to $5FF. This will allow a graphical demonstration of the bubble sort, since we will be able to watch its progress on the screen.

To help you understand how the bubble sort works, here is a Basic program that will do a bubble sort of the bytes contained in $400-$5FF, after filling the screen with random values.

```
10 FOR I=&H400 TO &H5FF:POKE I,
RND(256)-1:NEXT
20 FOR X=&H400 TO &H5FE
30 FOR U=X+1 TO &H5FF
40 IF PEEK(U) < PEEK(X) THEN T=
PEEK(X):POKE X,PEEK(U):POKE U,T
50 NEXT U:NEXT X
60 GOTO 60
```

Here's how the program works. First, the value in location $400 is successively compared to the values in locations $401-$5FF. If the contents of one of these locations is less than the value in location $400, the contents of this location and of location $400 are interchanged with one another (line 40). The new value in $400 is now used for new comparisons to other locations, until a still smaller value is found to take its place.

After the comparison of $400 with the last screen location, $5FF, has taken place, $400 must contain the smallest byte in screen memory. Then (NEXT X) the same process takes place except that $401 is compared with $402

through $5FF, and when this set of comparisons is finished, $401 will contain the second smallest byte value. And so forth, until the whole screen is arranged in ascending order of byte values. This takes quite a large amount of time, as you will see if you run this program. Incidentally, if you want the bytes arranged in descending rather than ascending order, all that is necessary is to change the "<" in line 40 to a ">".

Note, by the way, that the sort is according to a byte's display code, which is not the same as its ASCII code, except for uppercase letters. For example, lowercase "a" has a display code of 1, but an ASCII code of $61. If you put a letter on the screen by poking a byte directly into screen memory, rather than by using Basic's CHROUT routine (JSR [$A002]), you must use the display code rather than the ASCII code.

Now here's the assembly language exercise: simply (?) translate lines 20-50 of the Basic program above into a machine language routine, so that we will have a sort that zips along at a bit more satisfying speed. I've used X and U as variables in the Basic program in order to correspond to the use of registers X and U in the assembly language routine I've written. You'll find that routine posted in TUTA2.SRC. (Can you modify it to sort in descending order? A single poke will do it.) But, again, I encourage you to try to write one of your own before looking at that code.

In my next contribution to this tutorial series, I intend to use a bubble sort for something much more practical—a fast string sort routine. This will be a program of genuine practical value, not just a demonstration of programming technique. So, stay tuned!

```
00100 *TUTA2.SRC
00110 *ART FLEXSER
00120 *
00130 *THIS ROUTINE SORTS THE BYTES
CONTAINED IN THE TEXT
00140 *SCREEN LOCATIONS $400-$5FF
INTO ASCENDING ORDER.
00150 *THE FOLLOWING BASIC PRO-
GRAM WILL DEMONSTRATE
00160 *THIS ROUTINE IN OPERATION:
00170 *  10 FORI=&H400 TO &H5FF: POKE
I,RND(256)-1:NEXT
00180 *  20 EXEC &H3E00
00190 *  30 GOTO 30
00200 *(THIS ASSUMES THE ML HAS
BEEN LOADM'ED FIRST.)
00210      ORG    $3E00
00220 START LDX    #$400  START OF
TEXT SCREEN
00230 NEXTX LEAU   1,X    INITIALIZE
U=X+1
00240 NEXTU BSR    SWAP   INTER-
CHANGE BYTES IF NEC.
00250      LEAU   1,U    INCREMENT U
00260      CMPU   #$5FF  END OF TEXT
SCR?
00270      BLS    NEXTU  NO, NEXT U
00280      LEAX   1,X    INCREMENT X
00290      CMPX   #$5FE  LAST X?
00300      BLS    NEXTX  NO, NEXT X
00310      RTS
00320 *THIS ROUTINE COMPARES THE
CONTENTS OF THE BYTES
00330 *POINTED TO BY X AND BY U.  IF
THE BYTE POINTED TO BY X
00340 *EXCEEDS THAT POINTED TO BY U,
THE TWO BYTES ARE SWAPPED.
00350 SWAP LDA    ,X     GET X BYTE
00360      CMPA   ,U     COMPARE TO U
BYTE
00370      BLS    OUT    EXIT IF < OR =
00380 *SWAP THE BYTES
00390      LDB    ,U
00400      STB    ,X
00410      STA    ,U
00420 OUT   RTS
00430      END    START
```

Classic BASIC Game Series
SEEKER BY JOHN BARRETT

This game was originally written for a tape based CoCo 1 in 1984. It will run on a CoCo 2 or 3. Defend your planet from invaders by launching rockets from your orbiting moonbase. Contains lo-res graphics. Instructions are within the program.

```
DELPHI HEADER:
Name: SEEKER.BAS
Type: PROGRAM
Date: 25-FEB-1988 12:37
       by JBARRETT
Size: 5894
Count: 133 (as of 10 SEP 1998)
(Count is the number of times down-
loaded since the upload date. I'm not sure
if it includes times the program was listed
to the screen and "captured" that way --
which is how I "downloaded". Since this is
a straight BASIC listing, many people
may have captured their screen buffer
instead of downloading.)

10 CLS0
20 IF PEEK(33021)<>50 THEN110
30 PALETTECMP
40 POKE65497,0
50 PRINT:PRINT"COCO3 VERSION"
60 PRINT"RGB OR
COMPOSITE?":SCREEN0,1
70 A$=INKEY$:IFA$=""THEN70
80 IFA$="R"THENPALETTE RGB
90 IFA$="C"THEN PALETTECMP
100 FORX=1TO4:PALETTEX-1,X*5:NEXT
110 :PRINT@64,"DO YOU WANT
INSTRUCTIONS?":SCREEN0,1
120 A$=INKEY$:IFA$=""THEN120
130 A$=LEFT$(A$,1):IF A$="Y" THEN
1530
140 POKE340,0:IFPEEK(340)=254
THEN140
150 CLS0:PRINT@32*5,"
SEEKER":PRINT@32*11,"(C)1984 BY
J.H. BARRETT":PRINT"PLACED IN PUB-
LIC DOMAIN  25 FEB88":SCREEN 0,1
160 GR=1:YE=2:BL=3:RE=4
170 PRINT@0,"SELECT
LEVEL":PRINT@32," EASY":PRINT@64,"
HARD":SCREEN0,1
180 A=JOYSTK(0):A=JOYSTK(1):IF A<3
THEN Y=32 ELSE Y=64
190 X=0:HO=PEEK(Y+1024)
200 POKEY+1024,127+16*RND(8):
POKE1 024+Y+5,PEEK(1024+Y):FOR
G=1TO100:NEXT:POKEY+1024,HO:
POKE1024+Y+5,HO
210 IF PEEK(340)<>254 THEN 180
220 IFY>34 THEN MM=5 ELSEMM=1
230 FORF=0TO3:PRINT@32*F,STRING$
(32,128);:NEXT:SCREEN0,1: F=RND(-
TIMER)
240 GOSUB1030
250 PMODE1,1:SCREEN1,0:PCLS:
GOSUB1190:SCREEN1,0
```

```
260 '
270 'FORS=1TO4:PCOPY S+4 TO S:
NEXT
280 PCOPY3TO1:PCOPY4TO2
290 SM=MM+2
300 TIMER=0
310 GOSUB 1310
320 GOSUB 410
330 POKE340,0:IFPEEK(340)=254 THEN
GOSUB 490
340 GOSUB 620
350 IF RND(10-MM)=1 THEN GOSUB
1190
360 GOSUB 780
370 MC=MC+1:IFMC=100THEN MC=0:
MM=MM+1:IF MM>6THENMM=6
380 IFTIMER/60<1 THEN380
390 IF DM>29 THEN 1390
400 GOTO260
410 '
420 X=JOYSTK(0)*4:
430 Y=JOYSTK(1)*3
440 IFX<3THENX=3ELSEIFX>252THEN
X=252
450 IFY<3THENY=3ELSEIFY>188THEN
Y=188
460 COLORYE:DRAW"BM"+STR$(X)
+","+ STR$(Y)
470 DRAW"U4D4R4L8R4D4
480 RETURN
490 '
500 Q=0
510 FORF=1 TO 4:IF FR(F)=1 THEN Q=
Q+1
520 NEXT
530 IFQ=4 THEN RETURN
540 F=RND(4):IF FR(F)=1 THEN 540
550 FR(F)=1
560 TT(F)=0
570 PLAY"V1L255T255O1"
580 FORG=1TO3:PLAY"AV+AV+AV+V
+V+BBBCV+
590 NEXT
600 X(F)=MX:Y(F)=MY
610 RETURN
620 '
630 XX=INT(X):YY=INT(Y)
640 FORF=1TO4
650 IF FR(F)=0 THEN 760
660 X=INT(X(F)):Y=INT(Y(F))
670 IF X<XX THEN X=X+SM
680 IF X>XX THEN X=X-SM
690 IF Y>YY THEN Y=Y-SM
700 IF Y<YY THEN Y=Y+SM
710 IF RND(88)=1 THEN Y=Y+1
720 'PSET(X,Y,YE):
730 COLORYE:CIRCLE(X,Y),2
```

```
740 TT(F)=TT(F)+1:IF TT(F)> 30-MM*2
THEN CIRCLE(X,Y),10,YE:FR(F)=0
750 X(F)=X:Y(F)=Y
760 NEXT
770 RETURN
780 '
790 FORF=1TO4
800 IF AL(F)=0 THEN 910
810 AX=AX(F):AY=AY(F)
820 IFAX>127 THEN AX=AX-MM ELSE
AX=AX+MM
830 IFAY<96 THEN AY=AY+MM ELSE
AY=AY-MM
840 IF PPOINT(AX,AY)=BL THEN GOSUB
930:DM=DM+10:AL(F)=0:IF DM>15 THEN
PMODE1,3:COLORRE: LINE(0,0)-(255,
191), PSET,B:PMODE1,1
850 PSET(AX,AY,RE)
860 FORG=1TO 4
870 IF FR(G)<>1 THEN 890
880 IF AX<X(G)+10 AND AX>X(G)-10 AND
AY>Y(G)-10 AND AY<Y(G)+10 THEN
GOSUB 980:AL(F)=0:FR(G)=0: SC=SC+10:
G=100
890 NEXTG
900 AX(F)=AX:AY(F)=AY
910 NEXT
920 RETURN
930 'EX
940 FORQ=1TO10:PLAY"L255T255V
3101GO2GO1G":
950 CIRCLE(AX,AY),Q,YE
960 NEXT
970 RETURN
980 PLAY"V1":FORRQ=1TO10STEP4
990 CIRCLE(AX,AY),RQ,YE:PLAY "L234T
23301GO5G#V+V+V+"
1000 SC=SC+MM
1010 NEXT
1020 RETURN
1030 '
1040 PMODE1,3
1050 PCLS
1060 COLORBL:X=127:Y=96
1070 IFMM=5THENCIRCLE(X,Y),40 ELSE
CIRCLE(X,Y),29
1080 PAINT(X,Y)
1090 COLORRE
1100 RA=57.29577951
1110 FORD=1TO 360STEP 10
1120 DD=D/RA
1130 R=RND(20):IFMM>4THENR=RND(34)
1140 X=SIN(DD)*R+127:Y=COS(DD) *R+96
1150 LINE(X,Y)-(X+3,Y+3),PSET,BF
1160 NEXT
1170 COLORYE:LINE(0,0)-(255,191),
PSET,B
```
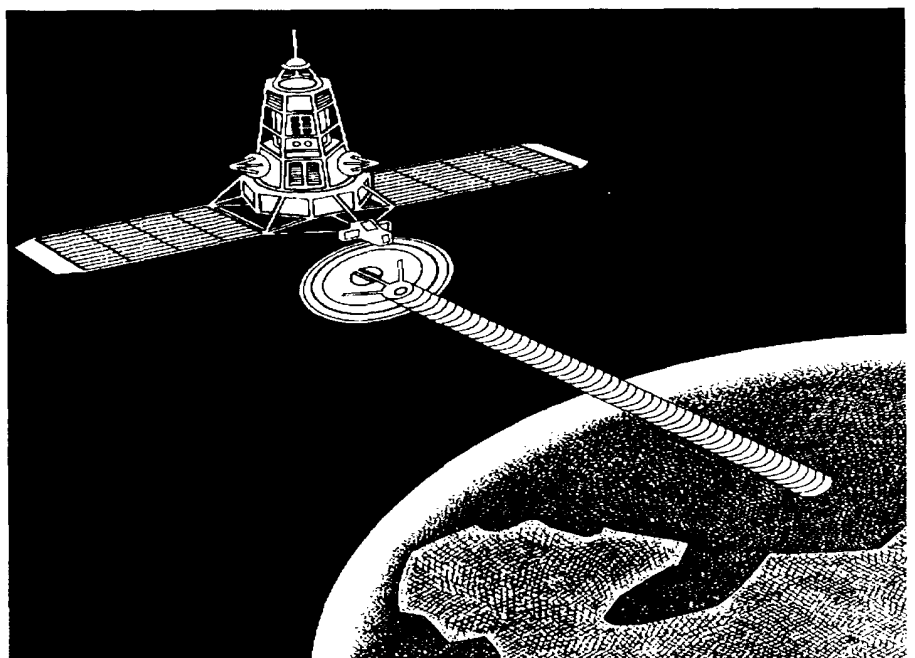
```
1180 RETURN
1190 'AL
1200 Q=0:FORF=1 TO 4
1210 IFAL(F)=1 THEN Q=Q+1
1220 NEXT
1230 IF Q=4 THEN RETURN
1240 Q=RND(4):IF AL(Q)=1 THEN 1240
1250 AL(Q)=1
1260 IF RND(2)=1 THEN AX(Q)=1 ELSE
AX(Q)=254
1270 AY(Q)=RND(190)
1280 SOUND133,1
1290 IF RND(3)=1 THEN AX(Q)=RND
(255):IF RND(2)=1 THEN AY(Q)=1 ELSE
AY(Q)=190
1300 RETURN
1310 'MOON
1320 COLORRE
1330 MD=MD+9:IF MD>359THEN MD=1
1340 DD=MD/RA
1350 MX=127+SIN(DD)*50:MY=96+
COS(DD) *50
1360 IFMM=5 THENMX=127+SIN(DD)
*80:MY=96+COS(DD)*50
1370 CIRCLE(MX,MY),5
1380 RETURN
1390 '
1400 COLORYE:SCREEN1,0
1410 FORR=1TO60 STEP2
1420 CIRCLE(127,96),R,YE:CIRCLE
(127,96), R,GR
1430 SCREEN1,0:NEXT
1440 CLS0:PRINT@33*2.1,"YOUR
WORLD IS NOW DUST!"
1450 SC=SC*100000/842
1460 PRINT@32*5,USING"SCORE
##,###,### ";SC
1470 PRINT@480,"PRESS BUTTON TO
PLAY AGAIN";:SCREEN0,1
1480 '
1490 POKE340,0
1500 :IFPEEK(340)<> 254 THEN 1480
1510 :CLSRND(8):FORK=1TO1000:
NEXT
1520 RUN
1530 '
1540 '
1550 '"INST
1560 CLS:
1570 READA$:
1580 IFA$="END"THEN 1650
1590 IFA$="@"THENA$=STRING$(32,
127+RND(8)*16)
1600 IFLEN(A$)<32THENA$=A$+" ":
GOTO1600
1610 PRINTA$;:PLAY"V22L255T255O4A
1620 L=L+1:IFL=15THENPRINT" press
any key";:L=0:FORFF=1TO100:FF=4: IF
INKEY$=""THEN NEXT ELSE CLS
1630 GOTO1570
1640 DATA@,@,@,@,@
1650 RUN
1660 DATA"      S E E K E R",,"(C) 1984
BY JOHN BARRETT
1670 DATA"@","YOU ARE THE COM-
MANDER OF STARION","1, WHICH
ORBITS THE PLANET AND-
```

```
1670 DATA"@","YOU ARE THE COM-
MANDER OF STARION","1, WHICH
ORBITS THE PLANET AND-","ROSSIA
IN A FIXED ORBIT."
1680 DATA"EVIL BORONS ARE ATTACK-
ING YOUR","PLANET. THE STARION 1
IS THE","ONLY DEFENSE."
1690 DATA@,@
1700 DATA"YOUR SCREEN SHOWS
THE LONG RANGE","RADAR, WITH
ANDROSSIA AT THE","CENTER. THE
STARION 1 ORBITS ","COUNTER-
CLOCKWISE. THE BORONS","WILL
APPEAR ALONG THE EDGES."
1710 DATA"PRESS THE FIRE BOTTON
ON THE ","JOYSTICK CONTROLLER TO
LAUNCH","A SEEKER DRONE. THESE
ROBOT ","SHIPS CANNOT BE
INDIVIDUALY ","CONTROLLED, BUT
WILL HOME IN ON","THE CROSS-HAIR
CURSOR, WHICH YOU", "CONTROL
WITH THE JOYSTICK."
1720 DATA@
1730 DATA"THE COMPUTERS OF YOUR
BATTLE","STATION CAN CONTROL
THREE DRONES","AT A TIME. THE
DRONES WILL DE","TONATE AFTER AN
INTERVAL EVEN","IF THEY DO NOT
INTERCEPT A","TARGET."
1740 DATA"YOUR HOME PLANET CAN
SURVIVE","THREE DIRECT HITS BY
THE BORONS.","IF THE PLANET IS
DESTROYED, THE","ORBITING
STARION 1 WILL","ALSO BE DE-
STROYED."
1750 DATA"REMEMBER, THE DRONES
ARE CALLED","SEEKERS BECAUSE
THEY SEEK THE","
CURSOR. THEY WILL
DESTROY A BOR-","ON
IF THEY COLLIDE WITH
IT."
1760 DATAEND
```

NEW Hardware coming from

Cloud Nine

c/o Mark Marlette
3749 County Road 30
Delano, MN 55328
email : mmarlett@isd.net
voice: 612-972-3261

operating system nine

512k - 2048k upgrade board

Just install SIMM memory in 512k increments (2x256K 8 or 9 chip SIMMs). Three chip SIMMs WILL NOT work! This is a timing requirement, as the 8/9 chip SIMMs use the same timing as the CoCo DRAM upgrades.

SCSI Host adapter interface

• Comes with OS9 Drivers, 6x09. 63b09e 1.78MHZ system "megaread" times are ~11 seconds with 512 byte sectors (Nitros 2.00 Level3).
  • 256/512/1024 Sector size selection
  • FULL SCSI ID supported
  • Parity generation, enable/disable. Can use with parity devices such as ZIP drives!
  • Gold plated card edge connector
  • 50 pin SCSI header port
  • Installation/Operation Manual
  • Schematic package
  • OS9 Utilities SCSItools, SCSIdesc, ZIP/JAZ Tools
  • SCSItools - A BASIC09 utility that will do low level SCSI commands.
  • SCSIdesc - A BASIC09 utility program that will create the SCSI descriptor for you based upon the menu drive options inputted.

• ZIPJAZtools - This utility will allow the features of the Iomega ZIP/JAZ drives. Eject disk, software protection are some. This utility isn't written yet, but I have the documentation needed from Iomega. Will do this soon!

These products should be available at the Chicago CoCoFest! Look for me there!!

A 512K SIMM upgrade is ready to ship. The unit will ship with the following items:
  1 - 512K SIMM Memory Board with 8 or 9 chip 120ns or faster SIMMS
  1 - Installation Manual
  1 - Schematic package
  1 - RSDOS Memory Test Program supplied on 5 1/4" disk.
$40 each including shipping, UPS ground, within the US. If you are outside of the US please indicate method of shipment desired and I will check into the added cost, if any.

It's like moving into a house with 10 times as many rooms and 10 times the stuff. At first you'll have trouble remembering where you put everything.

You will find, as you progress, that there are alternative ways to do a given task. The third section will involve a little less attitude and a little more technique. There I will go a little deeper into some points I have made so far. Now section two.

Beyond the System

A sort of philosophy is also involved when you begin to deal with OS9 software. If you used only smoothly functioning, shrink wrapped commercial software, you probably wouldn't be reading this. You've probably been on Delphi to try the OS9 SIG's software or ask for help with something that is not so smooth running. There are beginners who frequent Delphi and the internet CoCo list, as well as reading this magazine. Many are as full of questions as you are, but they have some answers also. At the other end of the spectrum there are some very expert and brilliant programmers who know "everything there is to know" about OS9. To often, you will find yourself simply talking a different language than these "tech types."

It isn't that they don't want to help. They're helping each other every day! If you want to know something about the exact syntax of an obscure part of a brand new update to the latest XYZ language enhancement, you're in luck. But if you want to know how to get the menu on yesterdays spreadsheet it might take longer.

Some very bright programmers dash off experimental programs for fun. Sometimes they like what they end up with and post it on Delphi or RTSI. Maybe they wrote 2 pages of documentations for a complicated communications program. Maybe they didn't finish it. Maybe they forgot they even wrote it! Grabbing the first thing in sight may or may not be a good idea.

If you're looking to OS9 to be the cure for the common cold, the end all, be all, you need to rethink things. It is a tough thing to learn. But some of those brilliant programmers I mentioned have worked long and hard at getting rid of bugs in the original package and adding yet more power to it.

I guess the irony is that this system sits in such a tiny box wishing it had some place to go. Users require more and more function. That calls for more and bigger software, and that calls for more memory.

Making a new OS9 Boot BootList

(this is part of "operating system nine", contributed by Rick Ulland)

If you look at your new modules directory you will see a file called bootlist. This is just what it seems, a list of the modules to be included in a new boot. The first step in creating a custom boot is to modify this file to reflect what you need for your system. A common change would be editing the disk drive entries- say changing d0_35s.dd to d0_40d.dd for double sided drives. This can be done with edit, or any text editor/word processor.

Sometimes it's handy to have more than 1 bootlist. For example, there are two patches to the CoCo's disk drives, one can read IBM disks and the other has a disk cache, and they won't work together. By keeping two slightly different bootlists, either version can be made up.

Compare the bootlist Tandy made with the module description enclosed. Notice how the list is organized- there is a block of the basic stuff (os9p2, INIT, IOMan) and then the rest of the list follows a definate pattern- the manager is given, then it's driver, followed by the descriptors that driver uses. There is a reason for this-OS9 can get confused if related pieces get separated. When adding new modules to bootlist, follow this pattern and you may never be stricken with the dreaded BLOB (Boot List Order Bug).

As an example, suppose you bought a new hard drive, and now there's a new /h0 and /ddh0 descriptor to install. First, copy the new modules to the MODULES directory, then add their names to bootlist. BUT-just sticking these on the end of the list can cause problems since they will be to far from RBF.mn (the Random Block File manager). Instead, insert them just above or below the floppy disk descriptors.

Tips and Tricks

Sometimes a bootlist just will not work, despite being ordered correctly. If this happens to you, try moving INIT to the very end of the list.

As you add more and more stuff to the bootlist, you're sure to get the dreaded error237 - system RAM full. This doesn't mean you are out of RAM, rather the 64K block the operating system uses is full. The only solution is to remove something from the bootlist.

One thing that doesn't need to be in the boot is cc3go. In fact thats the worst place for it! Everything in the bootlist is locked into memory when OS9 starts. Since cc3go is only used once- why keep it around? The only thing you need to do is place a copy of cc3go in the root (main) directory of the boot disk, then remove it from bootlist. With it as a separate file, OS9 can drop cc3go as soon as it is finished.

Another expendable is VDGInt. It's nice for games, and some other Tandy programs, like DeskMate3, use it. But keeping 2 separate graphics systems in memory at once is wasteful. A second boot disk set up for VDG graphics will allow you to squeeze a little more room.

OS9Gen

OS9Gen is actually a simple little utility to use, after all the parts are grouped together into one MODULES dir and the bootlist is made up.

Put a fresh disk in /d1, the BootMaker disk in /d0,then:

```
chd /d0/MODULES;chx /d0/CMDS
os9gen /d1 < bootlist
```

This will make a new boot that faithfully copies what you specified in bootlist. That will be the only thing on the disk, so you will have to copy the rest with dsave. Insert the System Master in /d0, then

```
chd /d0; chx /d0/CMDS
dsave /d0 /d1 ! shell
```

After dsave gets done, you will have a new System Master in /d1 that is a copy of the original, except it will boot up as you instructed in bootlist.

If you have removed cc3go from bootlist, be sure to copy it from the modules directory to the root of the new system disk.

Cobbler

If you want to make an exact copy of the boot that your machine is running now, Cobbler is the way to go. It's a handy way to make special purpose bootable program disks.

As an example, let's say you use DynaCalc alot, and would like your system to start up from the DynaCalc disk. Rather than using backup to make a working copy, do this: Format a new disk in /d1, and insert the DynaCalc disk in /d0.

```
cobbler /d1
chd /d0
dsave /d0 /d1 ! shell
```

For a 'custom' bootable to work, there are a few things it must have. First, it must have a CMDS dir with shell and grfdrv in it, along with any commands called by the program or the startup file, like echo or setime. (If the disk contains BASIC09 programs, you'll need runb, gfx2, syscall, and inkey). It should also have startup, to set the system up for the program. Once again, if cc3go has been removed from the boot, it must appear in the root dir. In the example above, all of these things (except cc3go) were supplied by the Tandy source disk, but that won't be true with programs from another source.

Classic BASIC Game Series
Master Mind
BY TOM O'BRIEN

This game was written by Tom in 1991. It is specifically for a CoCo 3 and will not run on a CoCo 1 or 2 as is. It could be modified to run on an earlier model, but would require extensive knowledge of graphics programming on both machines. Note that spaces in the text will have to be adjusted after the program is typed in to fit the screen correctly (it is difficult to typeset a program listing and get the spacing right!).

Name: MASTER MIND GAME
Type: PROGRAM
Date: 14-SEP-1991 12:09 by TOMO
Size: 8384 Count: 122

This is a Color Computer 3 version of the game "MASTER MIND" that I wrote myself. The object of the game is to guess the color code the computer has picked.

```
10 '****************************
20 '*    WRITTEN BY:       *
30 '*    TOM O'BRIEN       *
40 '****************************
50 PALETTE RGB:PCLS1:GOSUB 1080
60 L=10
70 PALETTE 0,34:PALETTE 1,0:
PALETTE 3,54:PALETTE 11,6
80 HSCREEN 2
90 HCLS 4
100 HLINE (0,0)-(160,140),PSET,B
110 HPRINT(24,16),"COLOR CHOICES"
120 HLINE (183,140)-(304,185),PSET,B
130 HLINE(187,145)-(300,170),PSET,B
140 HLINE(203,145)-(203,170),PSET
150 HPAINT (188,146),2,1
160 HLINE(219,145)-(219,170),PSET
170 HPAINT(204,146),3,1
180 HLINE(235,145)-(235,170),PSET
190 HPAINT(220,146),5,1
200 HLINE(251,145)-(251,170),PSET
210 HPAINT(237,146),6,1
220 HLINE(267,145)-(267,170),PSET
230 HPAINT(266,146),7,1
240 HLINE (283,145)-(283,170),PSET
250 HPAINT(282,146),9,1
260 HPAINT(290,146),11,1
270 HPRINT(24,22),"B Y L P O G D"
280 HPAINT (1,1),0,1
290 HLINE (80,00)-(80,140),PSET
300 FOR X=20 TO 140 STEP 20
310 HLINE (0,X)-(160,X),PSET
320 NEXT X
330 FOR Y=10 TO 130 STEP 20
340 FOR X=15 TO 60 STEP 15
350 HCIRCLE (X,Y),5,1
360 NEXT X
370 NEXT Y
380 P(1,1)=RND(7)
390 P(2,1)=RND(7)
400 P(3,1)=RND(7)
410 P(4,1)=RND(7)
420 IFP(1,1)=1THENP(1,1)=2ELSEIFP
(1,1)=2THENP(1,1)=3ELSEIFP(1,1)=3
THENP(1,1)=5ELSEIFP(1,1)=4THENP(1,1)=6
ELSEIFP(1,1)=5THENP(1,1)=7ELSEIFP(1,1)=6
THENP(1,1)=9ELSEIFP(1,1)=7THENP(1,1)=11
430 IFP(2,1)=1THENP(2,1)=2ELSEIFP
(2,1)=2THENP(2,1)=3ELSEIFP(2,1)=3THEN
P(2,1)=5ELSEIFP(2,1)=4THENP(2,1)=6E
LSEIFP(2,1)=5THENP(2,1)=7ELSEIFP(2,1)=6
THENP(2,1)=9ELSEIFP(2,1)=7THEN
P(2,1)=11
440 IFP(3,1)=1THENP(3,1)=2ELSEIFP
(3,1)=2THENP(3,1)=3ELSEIFP(3,1)=3THEN
P(3,1)=5ELSEIFP(3,1)=4THENP(3,1)=6
ELSEIFP(3,1)=5THENP(3,1)=7ELSEIF
P(3,1)=6THENP(3,1)=9ELSEIFP(3,1)=7
THENP(3,1)=11
450 IFP(4,1)=1THENP(4,1)=2ELSEIFP
(4,1)=2THENP(4,1)=3ELSEIFP(4,1)=3
THENP(4,1)=5ELSEIFP(4,1)=4THENP(4,1)=6
ELSEIFP(4,1)=5THENP(4,1)=7ELSEIF
P(4,1)=6THENP(4,1)=9ELSEIFP(4,1)=7
THENP(4,1)=11
460 T=1
470 X=15:PG=0:IF T>7 THEN
HPRINT(21,10),"YOU LOSE":GOSUB
920:GOTO880
480 IF T=1THENY=11ELSEIFT=2THEN
Y=31ELSEIFT=3THENY=51ELSEIFT=4
THENY=71ELSEIFT=5THENY=91ELSE
IFT=6THENY=111ELSEY=131
490 HPRINT (21,1),"ENTER YOUR
CHOICE:"
500 CH$=INKEY$:IF CH$=""THEN500
510 IFCH$="B"THENHPAINT(X+1,Y),
2,1:X=X+15:PG=PG+1:CH(PG,1)=2:
IFPG>3THEN 600ELSE500
520 IFCH$="Y"THENHPAINT(X+1,Y),
3,1:X=X+15:PG=PG+1:CH(PG,1)=3:
IFPG>3THEN600ELSEGOTO500
530 IF CH$="L"THENHPAINT(X,Y),5,1:
X=X+15:PG=PG+1:CH(PG,1)=5:IFPG>3
THEN600ELSEGOTO500
540 IFCH$="P"THENHPAINT(X,Y),6,1:
X=X+15:PG=PG+1:CH(PG,1)=6:IFPG>3
THEN600ELSEGOTO500
550 IF CH$="O"THENHPAINT(X,Y),7,1:
X=X+15:PG=PG+1:CH(PG,1)=7:IFPG>3
THEN600ELSEGOTO500
560 IFCH$="G"THENHPAINT(X,Y),9,1:
X=X+15:PG=PG+1:CH(PG,1)=9:IFPG>3
THEN600ELSEGOTO500
570 IFCH$="D"THENHPAINT(X,Y),11,1:
X=X+15:PG=PG+1:CH(PG,1)=11:IFPG>3
THEN600ELSEGOTO500
580 IFCH$=CHR$(8)THENIFX<16THEN
SOUND25,10:GOTO500ELSEX=X-15:
PG=PG-1:HPAINT(X,Y),0,1:GOTO500
590 SOUND25,10:GOTO500
600 P(1,2)=0:P(2,2)=0:P(3,2)=0:P(4,2)=
0:CH(1,2)=0:CH(2,2)=0:CH(3,2)=0:CH(4,2)=0
610 IFCH(1,1)=P(1,1)THENP(1,2)=1:CH
(1,2)=1
620 IFCH(2,1)=P(2,1)THENP(2,2)=1:CH
(2,2)=1
630 IFCH(3,1)=P(3,1)THENP(3,2)=1:CH
(3,2)=1
640 IFCH(4,1)=P(4,1)THENP(4,2)=1:CH
(4,2)=1
650 IFP(1,2)<>0THENGOTO690
660 IFP(1,1)=CH(2,1)ANDCH(2,2)=0
THENP(1,2)=2:CH(2,2)=1:GOTO690
670 IFP(1,1)=CH(3,1)ANDCH(3,2)=0
THENP(1,2)=2:CH(3,2)=1:GOTO690
680 IFP(1,1)=CH(4,1)ANDCH(4,2)=0
THENP(1,2)=2:CH(4,2)=1
690 IFP(2,2)<>0THENGOTO730
700 IFP(2,1)=CH(1,1)ANDCH(1,2)=0
THENP(2,2)=2:CH(1,2)=1:GOTO730
710 IFP(2,1)=CH(3,1)ANDCH(3,2)=0
THENP(2,2)=2:CH(3,2)=1:GOTO730
720 IFP(2,1)=CH(4,1)ANDCH(4,2)=0
THENP(2,2)=2:CH(4,2)=1
730 IFP(3,2)<>0THEN770
740 IFP(3,1)=CH(1,1)ANDCH(1,2)=0
THENP(3,2)=2:CH(1,2)=1:GOTO770
750 IFP(3,1)=CH(2,1)ANDCH(2,2)=0
THENP(3,2)=2:CH(2,2)=1:GOTO770
760 IFP(3,1)=CH(4,1)ANDCH(4,2)=0
THENP(3,2)=2:CH(4,2)=1
770 IFP(4,2)<>0THEN810
780 IFP(4,1)=CH(1,1)ANDCH(1,2)=0
THENP(4,2)=2:CH(1,2)=1:GOTO810
790 IFP(4,1)=CH(2,1)ANDCH(2,2)=0
THENP(4,2)=2:CH(2,2)=1:GOTO810
800 IFP(4,1)=CH(3,1)ANDCH(3,2)=0
THENP(4,2)=2:CH(3,2)=1
810 GOSUB 1020:W=97
820 IF P(1,2)=1THENHCIRCLE(W,L),
5,1:HPAINT(W+1,L+1),4,1:W=W+15ELSE
IFP(1,2)=2THENHCIRCLE(W,L),5,1:
HPAINT(W+1,L+1),1,1:W=W+15
830 IFP(2,2)=1THENHCIRCLE(W,L),
5,1:HPAINT(W+1,L+1),4,1:W=W+15ELSE
IFP(2,2)=2THENHCIRCLE(W,L),5,1:HPAINT
(W+1,L+1),1,1:W=W+15
840 IFP(3,2)=1THENHCIRCLE(W,L);
```

```
5,1:HPAINT(W+1,L+1),4,1:W=W+15
ELSEIFP(3,2)=2THENHCIRCLE(W,L),5,1:
HPAINT(W+1,L+1),1,1:W=W+15
850 IFP(4,2)=1THENHCIRCLE(W,L),5,
1:HPAINT(W+1,L+1),4,1:W=W+15ELSE
IFP(4,2)=2THENHCIRCLE(W,L),5,1:HPAINT
(W+1,L+1),1,1:W=W+15
860 IFP(1,2)=1ANDP(2,2)=1ANDP(3,2)=
1ANDP(4,2)=1THENHPRINT(21,10),"YOU
WIN!!!!":GOTO 880
870 L=L+20:T=T+1:GOTO470
880 HPRINT(21,11),"DO YOU WANT
TO PLAY"
890 HPRINT(21,12),"AGAIN?"
900 A$=INKEY$:IFA$=""THEN900
910 IFA$="Y"THEN60ELSEEND
920 HPRINT(1,19),"SOLUTION:"
930 HCIRCLE(15,175),5,1
940 HCIRCLE(30,175),5,1
950 HCIRCLE(45,175),5,1
960 HCIRCLE(60,175),5,1
970 HPAINT(16,176),P(1,1),1
980 HPAINT(31,176),P(2,1),1
990 HPAINT(46,176),P(3,1),1
1000 HPAINT(61,176),P(4,1),1
1010 RETURN
1020 FL=0
1030 IFP(1,2)>P(2,2)THENSA=P(1,2):
P(1,2)=P(2,2):P(2,2)=SA:FL=1
1040 IFP(2,2)>P(3,2)THENSA=P(2,2):
P(2,2)=P(3,2):P(3,2)=SA:FL=1
1050 IFP(3,2)>P(4,2)THENSA=P(3,2):
P(3,2)=P(4,2):P(4,2)=SA:FL=1
1060 IFFL=1THEN1020
1070 RETURN
1080 PMODE 3,1
1090 DRAW"BM1,1;S3C2D80R11U55F1
2E12D55R11U80G22H22"
1100
DRAW"BM44,1;C3D80R14U40R13D40R14
U80L39;BM63,13;D12L12U12R12"
1110 DRAW"BM80,1;C4D43R22D24L22
D13R35U51L22U15R22U14L35"
1120 DRAW"BM115,1;C2D14R20D66R1
3U66R20U14L53"
1130 DRAW"BM160,1;C3D80R48U13L3
0U22R20U13L20U23R30U13L48"
1140 DRAW"BM204,1;C4D80R19U35R8
D3R2D3R2D3R2D3R2D3R2D3R2D3R2
D3R2D3R2D3R2D3R2D3R2D3R13U3L2
U3L2U3L2U3L2U3L2U3L2U3L2U3L2U3
L2U3L2U3L2U3L2U3L2R10U52L47"
1150 DRAW"BM217,13;D14R14U14L14"
1160 DRAW"S4;BM1,90;D90R17U65F1
0E10D65R17U90L17G10H10L17"
1170 DRAW"BM60,90;C2D15R21D60L
21D15R58U15L21U60R21U15L58"
1180 DRAW"BM125,90;C3D90R15U60R
4D4R2D4R3D4R2D4R2D4R2D4R2D4R2
D4R2D4R2D4R2D4R2D4R2D4R2D4R2D
4R15U90L15D45L3U4L2U4L2U4L2U4L2
U4L2U4L2U4L2U4L2U4L2U4L2U5L23"
1190 DRAW"BM195,90;C4D90R40E10U
70H10L40;BM210,105;D60R12E5U50H
5L12"
1200 PAINT(2,2),2,2:PAINT(46,3),3,3:
```

```
PAINT(82,2),4,4:PAINT(117,2),2,2:PAINT(162,
2),3,3:PAINT(206,2),4,4
1210 PAINT(2,92),4,4:PAINT(62,92),2,2:
PAINT(127,92),3,3:PAINT(197,92),4,4
1220 FOR I=1TO 30
1230 SCREEN 1,0
1240 SCREEN 1,1
1250 FORJ=1TO50:NEXTJ
1260 SCREEN 1,0
1270 FORJ=1TO 50:NEXTJ
1280 NEXT I
1290 SCREEN 1,0
1300 WIDTH 40
1310 CLS4
1320 LOCATE 5,12
1330 ATTR 2,3
1340 PRINT "DO YOU NEED
INSTRUCTIONS?(Y/N)";
1350 AN$=INKEY$:IFAN$=""THEN1350
1360 IF AN$="Y"THENGOSUB1380
ELSEIFAN$="N"THENWIDTH32:RETURN
ELSEGOTO1350
1370 RETURN
1380 CLS5
1390 LOCATE 3,0
1400 ATTR 0,4
1410 PRINT"MASTER MIND IS A GAME
OF BREAKING A  COLOR CODE. YOU
HAVE SEVEN CHANCES TO  BREAK
THE CODE THAT THE COMPUTER
HAS  SELECTED."
1420 LOCATE 10,3
1430 PRINT"A COLOR CODE BAR
WILL APPEAR  IN THE LOWER RIGHT
HAND CORNER OF THE
SCREEN. YOU SELECT YOUR COL-
ORS BY  PRESSING THE KEY UNDER-
NEATH ITS COLOR."
1440 PRINT"IF YOU MAKE A MISTAKE,
YOU CAN USE THE  BACK ARROW
KEY AND MAKE ANOTHER SELEC-
TION."
1450 LOCATE 3,10
1460 PRINT"THE BOARD IS DIVIDED
INTO TWO  SECTIONS. THE FOUR
CIRCLES ON THE LEFT  HAND SIDE
OF THE BOARD ARE USED BY THE
PLAYER AND THE RIGHT HAND SIDE I
S USED  BY THE COMPUTER."
1470 PRINT"  THE PLAYER CHOOSES
COLORS FROM THE  COLOR BAR
TRYING TO MATCH WHAT THE
COMPUTER HAS PICKED. THE
COMPUTER CAN  PICK ANY NUMBER
OF COLORS FROM THE BAR. FOR
EXAMPLE, THE COMPUTER MIGHT
SELECT  BLUE BLUE PURPLE OR-
ANGE."
1480 LOCATE 0,22
1490 PRINT"PRESS ANY KEY TO
CONTINUE";
1500 ST$=INKEY$:IFST$=""THEN 1500
1510 CLS5
1520 PRINT"  AFTER THE PLAYER
HAS MADE THEIR FOURTH CHOICE,
THE COMPUTER WILL TELL  THEM
```

```
HOW MANY COLORS ARE RIGHT
AND HOW  MANY ARE IN THE COR-
RECT SPOT."
1530 PRINT"  A BLACK SPOT INDI-
CATES THAT THE PLAYER HAS
PICKED A CORRECT COLOR  BUT IT
IS IN THE WRONG SPOT. A WHITE
SPOT INDICATES THAT THE PLAYER
HAS PICKED THECORRECT COLOR
AND IT IS IN THE CORRECT SPOT."
1540 PRINT"  WHEN A PLAYER GETS
FOUR WHITE SPOTS YOU WIN THE
GAME. IF AFTER SEVEN TRIES  THE
PLAYER HAS NOT FIGURED OUT THE
CODE,THE COMPUTER WILL DISPLAY
THE SOLUTION AT THE BOTTOM OF
THE SCREEN."
1550 PRINT:PRINT"PRESS ANY KEY
TO START THE GAME";
1560 ST$=INKEY$:IF
ST$="" THEN 1560
1570 WIDTH32
1580 RETURN
```