OS-9 Newsletter ©

Q&A

Q: I have *Shell+* and want to use the logical argument if that is suppose to be available with *Shell +* for writing script files. Specifically, I want to read a directory and if a file is found I want a confirmation message printed to the screen else a "not found" message displayed. I've read the documentation for *Shell+*, but I cannot get it to work.
-- Terry Goode;Fidonet --

A: Terry, how about the following script:

```
if -f /dd/cmds/grfdrv
echo GrfDrv exists in /dd/CMDS
else
echo GrfDrv file NOT in /dd/CMDS
endif
```

This is the output from Tim Kientzle's HELP utility:

Read over page 8 again of your *Shell+* documentation for a complete listing of all of the condition testing supported by *Shell+*. The command syntax and options are listed below.
Syntax: if <option> <filename>
            <shell commands>
        else
            <shell commands>
        (if or endif)
Options:
        -y read one character from stderr, validate for y/n (y means true)
            -f file true if file exists and is a file
            -r file true if file exists and is readable
            -w file true if file exists and is writ able
            -e file true if file exists in execution directory
            -d file true if file exists and is a directory
                -- Tom Birt;Fidonet --

Q: I used *build* to make a script file called **CHD** to change drive to hard disk...but it does not work. Here is what I entered:

```
Build chd
chd /hd
chx /hd/cmds
```

Where am I going wrong? do I need to add something to tell which window?
-- Thomas Mathes --

Programming Properly in Basic09

One of my pet hates with programmes written for OS9 (and indeed any computer), is programmes which having created a special screen type for running in, proceed to quit WITHOUT RESTORING the screen that I was running.

The fault lies squarely with the (lazy) programmer, because it is mostly very easy to save the current screen settings, and restore them after the programme has finished. I guess the place to start is to save the current settings from what the techos call the PD.OPTS section. The information from the PD.OPTS can be easily saved in a 32 byte buffer (make sure it is in the main routine of your programme). Here is a Basic09 sample:-

```
TYPE registers=cc,a,b,dp:byte; x,y,u:integer
DIM regs:registers
DIM getstt,setstt:byte
DIM opt_buff(32):byte

getstt=$8D \ (* Get Status system call
setstt=$8E \ (* Set Status system call

regs.a=0 \ (* use STDIN all others are automatic
regs.b=0 \ (* GS_OPT call
regs.x=ADDR(opt_buff) \ (* address of 32 byte buffer
RUN syscall(getstt,regs) \ (* do the job
```

That piece of code is broken up into the DIMension statements, including the TYPE complex variable for the CPU

registers, setting the various variables to their values (see comments), and the actual code for the operation. You only need to change the PD.OPTS of ONE of the standard paths, all the others merely mirror STDIN, so what you change in one, is also changes in all the others. The code to reset the PD.OPTS is very simple:-

```
regs.a=0 \ (* STDIN again
regs.b=0 \ (* SS_OPT call
regs.x=ADDR(opt_buff) \ (* address of 32 byte buffer
    of previous
RUN syscall(setstt,regs) \ (* Set Status call this
    time.
```

We still need to deal with changes to the screen, like screen type, palette colours, and foreground, background, and border colours.

Now, there are (at least) two ways to do this. I have used both successfully, so I will show you what I have done. The first method assumes plenty of memory, because it involves creating a new screen OVER THE TOP OF the existing screen. This is a bit more involved for future pieces of code as you will see. Here is the code:- (I'm assuming previously DIMmed variables for clarity.)

```
(* first open a path to a new descriptor
OPEN #wpath,"/w":UPDATE \ (* UPDATE essential here
RUN gfx2(#wpath,"DWSET",7,0,0,80,24,0,1,2) \ (set
    up type 7 graphics screen

RUN gfx2(#wpath,"SELECT") \ (* make the new screen
    visible
RUN gfx2(#wpath,"FONT",200,1) \ (* select font to use
```

Now, that piece of code assumes 1. that you have the descriptor '/w' in memory (from your boot), and 2. that you have merged sys/stdfonts (although this can be done within the programme).

One niggly problem with this approach is that ALL print statements and gfx2 commands MUST quote the #wpath, else the printing will go to the (invisible) window underneath. To reselect the original window, here's what you do:-

```
RUN gfx2(0,"SELECT") \ (* select the old screen
RUN gfx2(#wpath,"DWEND") \ (* kill off the old screen
```

OK, so much for the first method. Here's another way:-

```
DIM pals(16):BYTE \ (* Palette reg buffer
DIM sels(3)BYTE \ (* Fore, background and border regs
    buffer
DIM sctyp:BYTE \ (* screen type code 1..8
DIM screen_x:INTEGER
DIM screen_y:INTEGER
TYPE registers=cc,a,b,dp:BYTE; x,y,u:INTEGER
DIM regs:registers
DIM getstt,setstt:BYTE

getstt=$8D
setstt=$8E
regs.a=0 \ (* stdin path
regs.b=$91 \ (* SS.Palet
regs.x=ADDR(pals) \ (* address of palette buffer
RUN syscall(getstt,regs) \ (* do Get Status call
```

You now have the palette registers of the old screen.

```
regs.a=0 \ (* stdin path
regs.b=$96 \ (* SS.FBRgs
regs.x=ADDR(sels) \ (* buffer for colour registers
RUN syscall(getstt,regs) \ (* do Get Status call
```

Now you have the colour registers.

```
regs.a=0 \ (* stdin...
regs.b=$93 \ (* SS.ScTyp
RUN syscall(getstt,regs) \ (* do Get Status call
sctyp=regs.a \ (* save screen type returned in A reg
```

Now you have the screen type.

```
regs.a=0 \ (* stdin...
regs.b=$26 \ (* SS.ScSiz
RUN syscall(getstt,regs) \ (* do Get Status call
screen_x=regs.x \ (* X size in x reg
screen_y=regs.y \ (* Y size in y reg
```

Now you have the screen size. This info is probably all you need. I know that things like working area can be different, but most of the info is saved, though you could probably take this to all sorts of depths. Now you can kill off the old screen, and restart a new one:-

```
RUN gfx2(0,"DWEND") \ (* Kill old screen
RUN gfx2("DWSET",7,0,0,80,24,0,1,2) \ (* set up your
    screen
RUN gfx2("SELECT") \ (* make it visible
RUN gfx2("FONT",200,1) \ (* select a font
```

Now you can use the screen to your heart's content, and you don't have to worry about a path to the screen. To restore, you must reverse the previous procedure, although, not in as many steps.

```
RUN gfx2(0,"DWEND")
regs.a=0 \ (* stdin...
regs.b=$97 \ (* SS.DFPal
regs.x=ADDR(pals) \ (* palette regs
RUN syscall(setstt,regs) \ (* do Set Status call
RUN gfx2("DWSET",sctyp,0,0,screen_x,screen_y,
    pals(1),pals(2),pals(3))
RUN gfx2("SELECT") \ (* make it visible
```

Now you are back to where you started. If you had a need to change things like the screen pause etc., you should have also used the routine at the start of this article. One point I must stress again, is that you MUST use variables declared in the main part of your programme (as in main() in C programming) because otherwise your variable contents will be lost.

I hope that this will help programmers (and experimenters) to programme more neatly, and perhaps I have been able to impart some of my experiences to someone. If so, great. Regards,

-- Bob Devries;FidoNET --
Australian National OS9 Usergroup
Zone 3 OCN Coordinator

Q&A

A: I thought I'd jump in here to explain what I've seen as a general misconception among a large number of OS9 users. The fact is, you cannot use a script file (as above) to change your current directory settings. What happens is that OS9 actually starts a new shell for running a script, and when it is finished, it restores all the old conditions, as they were before the script was run.

A few users came to grief (myself included) when changing from Level 1, where we could, for example use the line 'tmode pause' in the startup, when the same line was used in level 2, it didn't work! The reason: the startup shell was terminated, and a new one started for the TERM shell.

The other thing wrong with the above idea, is that the name of the script file is the same as a file in the CMDS directory (well it would be in mine...) and OS9 checks in the execution dir first. If you try this for you script file:

```
chd /hd
chx /hd/cmds
dir
dir x
```

you'll find that it does indeed change to the correct directories, as the **'dir'** commands will show, but when the script terminates, the old original directories are back!
-- Bob Devries;Fidonet --
Australian National OS9 Usergroup
Zone 3 OCN Coordinator

Q: How can anyone use a program like *dEd* when you can't even understand what the program is displaying on the screen? It just looks like a *dump* commands display to me.

A: When using DED with a file such as DED /DD/CMDS/MONTYPE you can determine the actual offset by taking the sector number displayed in the upper left hand corner, adding the offset down the left hand column and then the offset running across the top.

EX. sector=$01
offset down column=$C0
offset across top=$08

actual offset=$01C8

This applies to individual files not necessarily whole disks opened with @ as in DED /D0/@

The actual offset in this case is from the first byte on disk and not a from a byte in the currently displayed file/directory/sector.
-- Dave Gantz;FidoNET --
-- Data Warehouse, Spokane, 509-325-6787 (1:346/9) --

Telecommunications

via local FidoNET BBS's

The below listing is a recent compilation of Fido Bulletin Boards that are currently carrying OS-9/CoCo echoes in the Northwest. Most of these boards have recently added the Pacific Northwest (PNW) OS-9 echo. Everyday, more than a dozen messages are posted in the OS-9 Echo for everyone to read. Private mail service is also available on these Bulletin Boards. There are literally dozens of echoes available besides OS-9 or CoCo. There is even an MM1-Tech Echo which is just as active as the standard OS-9 Echo. Can you imagine reading 10 to 30 messages/day about the latest goings on with the MM/1 or TC70 or KIX30 or System IV or ......

Need help with a CoCo or OS-9 problem? Leave a message and you will have a bundle of messages addressing your problem. Maybe one of them will solve your problem. And it's all FREE. No long distance fees or hourly rates. Select the BBS in your local area and give them a call.

PACIFIC NORTHWEST Fido OS9/CoCo BBS's

BBS Name	Phone No.	Location	Fido Address
Columbia Hts BBS	(206) 425-5804	Longview/Kelso RiBBS	1:105/641.0
Data Warehouse	(509) 325-6787	Spokane RiBBS	1:346/9.0
Doghouse BBS	(509) 325-3169	Spokane RiBBS	1:346/21
The Diamond Mine	(509) 325-5160	Spokane RiBBS	1:346/23
OS-9 Tacoma BBS	(206) 566-8857	Tacoma RiBBS	1:138/164.0
Ultimate Exper'nce	(206) 299-0491	Anacortes RiBBS	90:80/3
Kzin BBS	(604) 589-5545	Vancouver BC RiBBS	1:153/912
Pot O' Gold BBS	(604) 564-8869	Prince George, BC	1:359/251
Top Hat BBS	(206) 244-9661	Seattle [COCO echo]	1:343/40
The Right Place	(206) 581-2775	Tacoma[TaCOCO echo]	1:138/131
Cloud's_Corner	(206) 377-4290	Bremerton[OS9,PNW-OS9]	1:350/30
Graphics Guru	(206) 423-2357	Longview RiBBS	1:105/647
Inland Empire Archive	(509)624-679	Spokane RiBBS	1:346/10

Compiled by Chris Johnson;Sysop: OS-9 Tacoma BBS(1:138/164.0)

OS-9 Calendar

March 27-28
Middle America's Fest
Des Moines, Iowa
See Special Update Article elsewhere in this issue

June 25 - 26
PNW CoCo Fest
Port Orchard, Washington

Frank Hogg's Kix 30

by Allan Morgan

Allan Morgan (Longview/Kelso CoCo Club) recently purchased an FHL Kix30 68K machine and has taken the time to record a journal of his discoveries on his new machine. The Kix\30 is the newest of the 68K machines and has generated a lot of interest.

My Kix\30 motherboard arrived yesterday. Needless to say, I'm anxious to get it set up! I placed the order back on November 9th. Frank Hogg (FHL) had it ready to ship by December 21st. I had him hold off for a week because I was afraid of getting it lost in the last minute Christmas shipping rush so he shipped it Dec. 29th.

Oops!, they shipped it with a 25mhz 68030. Darn, I ordered a 16Mhz system <grin>. The crystal is for 16Mhz but I think the upgrade to 25Mhz will be a snap! I will be getting a mini-tower case for it tomorrow. I will probably set it up with a 1.44Meg floppy only. I have one of those on the CoCo too, so I can do any transfers via the 3 1/2in. drives. I had previously picked up a Quantum LPS240 SCSI hard drive. The hard drive performance should be outstanding! The video-audio-keyboard card will be several more weeks. I will have to use the CoCo as a terminal for the time being. . I also found a deal on 1 Meg 60ns simms a while back so there are 8 of them installed. One interesting note-- The DMA chip was socketed on the back side (solder side) of the board. Why? I'll have to ask about that. I'll keep you all posted as this goes along.

I got my Kix\30 up and running last Saturday night (1-9-93). Everything went quite well except for some struggles with the address jumpers on the new floppy drive and a mis-wired serial cable I made. Of course, it's Steve Hammond's fault. He held the connector upside down while I soldered it!

Since the video-audio-keyboard is not here yet, I have been using my CoCo as the system terminal (on the /term port). We made up a null modem cable for that and it runs at 9600 with very few problems.

After booting the Kix off of Frank Hogg's floppy, we formatted the hard drive and then *Dsave*'d OS9 to the hard disk. I pressed the reset button and.....it worked first try. The hard drive is real fast compared to the CoCo, and even quite fast compared to the IDE drive on my 486/33 at work.

I have been downloading all of the OSK stuff I can find on Delphi and am working now on getting the TOPS software. Since the CoCo is now connected to the Kix on a null-modem cable, my modem is connected to /t1 on the Kix. I have a simple terminal program, *Sterm*, running quite well now on the Kix\30.

That's a lot of software at work: The modem is on /t1 of the Kix, *Sterm* is running in the Kix, the output of *Sterm* is to /term, but /term is actually /t2 on the CoCo which is running *Osterm* in (sort of) Ansi mode. It all works quite well, though. File downloads use RZ in the OSK machine, so

the data ends up on the right hard drive.

I have been starting to use the OSK compiler. At first, I just typed "cc <sourcefile.c>" but later, I discovered the -K=2 switch. This tells the compiler that you have a 68020 or better. The resulting output is quite a bit faster. I found a switch in *dmode* to set up my floppy for high-density mode today. That's nice! The transfer speed is nearly twice as fast and now each disk holds 1.47Megs.

1/16/93: I downloaded two programs for testing and tuning: *Dhrystone.c* and *ddtest*. *Dhrystone.c* is the source code for that well-known meaningless performance measurer. *Ddtest* was written by Mark Griffith (on Delphi, usually). I used *ddtest* to help me configure the floppy drive descriptors. In 3.5 inch high-density mode, with an interleave of 2, *ddtest* reports 17.98Kbytes/sec for write and 14.41Kbytes/sec for reads. Oh, verify was off. *Ddtest* reports 1942.86Kbytes/sec reading from the hard drive but only 193.46Kbytes/sec writing to it.

The *Dhrystone* benchmark was educational for me. I am not a wizard yet when it comes to C, but it compiled first try just by typing "cc dhrystone.c" When run, it reported 2272 Dhrystones/second. There is a "switch" for the C compiler to cause it to compile for the 68020. I re-compiled *dhrystone.c* and ran it again. This time, it reported 3125 dhrystones/sec. My machine is a 16Mhz 68030. For comparison, I transferred *Dhrystone* to my son's 386-SX/16. The same source code ran through Turbo-C without a hitch. Sadly, he obtained 3850 Dhrystones/sec. I suspect that a 386DX would produce about the same results for this test as there are no affordable 32-bit compilers under MS-DOS anyway. It appears that the 68030, under OSK at least, is about clock-for-clock the same performance as a 80386 under *Desqview-X*, as both machines reported 3125. The 3850 reading was under single-tasking MS-DOS. Oh, yes...the CoCo runs about 221 Dhrystones/sec.

So far I have accumulated about 12 Megs of OSK stuff. I found another disk transfer testing utility like the *ddtest* program mentioned above. This one is called *disktest* which appears to work a lot like Megaread. It reports 4.6 Megabytes/sec read and 325 Kbytes/sec write. Now what do I do, average the results? <grin>. That's for a 7 Megabyte size file, which is certainly larger than any cache.

I have quit using the CoCo as a terminal. I had an old "boat-anchor" VT100 terminal in the shop and it performs quite a bit better. OSK Rogue runs perfectly on it! The terminal is a lot faster and smoother than the CoCo.

1/25/93: I had yet to receive the OSK manuals. I had been learning my way around by typing each command with -? and giving each thing a test. I was quite proud when I figured out the C compiler that way. I now have the official Microware manuals which are quite nice.

-- Allen Morgan --

SYMBOLISM IN C by Randy Kirschenmann

The C programming language is seen to be a very versatile medium for creating a wide range of program types. From the most simple of applications, to complete operating systems, this language provides the sophistication and efficiency needed to accomplish both the low level accessability as well as the high level functionality which makes C so attractive to programmers. This versatility is accomplished by a very small set of keywords and symbols. The list of keywords can be learned in a couple of hours of study, but to fully understand the intricacies of C's symbolism will take much longer.

There are several aspects of the C language which may make it difficult. The difficulties in learning the C language's use of symbols come from the double meanings that many of them have. For example, the & can be used in three different ways. When preceeding a variable data name, as in &data, it represents the main storage address of the data item. When used as a binary operator, placed between two data items of type int or char, as in data1 & data2, it becomes the bitwise AND operator. Finally, coded as && it represents the logical AND operator and is placed between two expresions, as in (expresion1) && (expression2)... Nearly all of the special characters used in C have double meanings. For novices of the language, reading C code must be done with care to get the true meaning of the seemingly ambiguous notation. Where the special characters are placed in relationship to other elements of the statement must be taken into account.

Another aspect of the C language which must be mastered in order to fully understand the programs one reads, is the rules of precedence and associativity which govern the evaluation of complex expressions. A seemingly simple expression may take on a completely different meaning than what is expected if the order in which the elements of the expression are evaluated is something other that what you expect. I imagine we've all seen the example of the statement: 2 + 3 * 4; yeilding a result of 14 instead of 20. This is true in C as it is in other programming languages that place a higher precedence to the multiplicative operator (*) than to the additive operator (+). However, this isn't the end of it in C. All of the special characters have their place in a hierarchy of precedence which must be studied with care. Not only their precedence, but also the direction of evaluation can be different. While most of the special characters are evaluated in a normal left to right

progression within the C statement, a few of them are evaluated from right to left. This allows one to code: a = b = c = 0; to assign the three data variables a, b and c the value of 0 all at once, because the assignment operator (=) is evaluated from right to left. And don't mix assignment with equality (==). I've done this myself many times. It can be very frustrating to search through a program which isn't functioning the way you expect looking for this mistake.

Another source of difficulty in reading C code is found in the style of most C programmers. The C language allows a programmer to combine several statements into one complex expression which the compiler quite comfortably evaluates for him. This allows the efficiency of code that we all prefer. It does, unfortunately, make for difficult reading at times. For example the block of code:

```
for(;;) {
        c = getchar();
        if(c == EOF)
                break;
        if(c == '\n')
                break;
        printf("%c", c);
}
```

will display a line of text to the terminal. It is quite straight forward and easy to read, but most C programmers will code this as:

```
while(((c = getchar()) != EOF) && (c !='\n')))
        printf("%c", c);
```

For as simple an example as this, the second form is as easily deciphered as is the first, but you can imagine how criptic more complex expressions can become.

In general I find C to be a very natural programming language, but be forewarned that it has its problems. For the novices (and the experts, too) care should be taken when reading C code to avoid the traps that await us. Its easy to misread a complex expression and often I find myself going over a block of code several times before I suddenly say "aha... that's what I've missed". So when learning C as a new language take your time with it and get the full meaning of its symbolism.

-- Randy Kirschenmann --
Mt. Rainier Coco Club

$400 worth of software for $34

Dave Meyers, President of CoCoPRO is under financial stress and as a result is making an unbelievable offer to the CoCo Community:

If you purchase at least $30 of software from the list below before February 28, 1993, you will be given license to obtain and freely use all of the products listed below.

The catch? Well there is a condition. Dave sent out flyers to all of their present and past customers and if 33% of the mailings return a favorable response, then CoCoPRO will make all of the below listed software available for download via password access to *Jim's CoCo Corner BBS* at (313) 292-4713, or via E-Mail to your Delphi account. If 40% respond, then the packages will be made freely available on major BBS' for download by **ALL ! ! ! ! !** For those who do not own modems or those who would prefer not to incur the downloading time and expense, copies of the program packages via mail for a duplicating charge of $1/disk, plus $3/order for shipping & handling will also be made available.

You've got to admit, this is one great way to demonstrate your support for those who support you in the CoCo Community and come out a *winner* at the same time.

I have viewed and operated most of the software listed below and can vouch for the professional quality of all of the software. My favorites are *Data Windows* with *Data Merger* (SUPER COOL DATABASE), *Simply Better* Word Processor for RSDOS, *LV2 Tools* Utilities for OS-9, *V-Term* Terminal Program, *Zapper* Disk Editor program, *MultiMenu* which is a user customized replacement for MultiVue, and finally (my special favorite) *Presto Partner* appointment calendar and telephone dialing rolodex database.

Purchase $30 of software from this list before February 28, 1993 and get the rest for FREE!

Data Windows (OS-9) $39.95
Data Merger (OS-9) $12.95
Newspaper09 (OS-9) $34.95
NewsFONTS09 (OS-9) $9.95
Simply Better (CC3) $24.95
V-Term (CC3,modem) $24.95
CoCoPRO Solitaire (OS-9) $9.95
Zapper (OS-9) $14.95
LV 2 Tools (OS-9) $19.95
Presto Partner (OS-9) $19.95
Tools 2 (OS-9) $24.95
MultiMenuOS-9,Mvue)$14.95
OS9 Lev 2 BBS (OS-9) $19.95
Disk Mgr.Tree (OS-9) $18.95

CoCoPRO also has a large inventory of CoCo/OS9 hardware and software products. Give Dave Meyers a call. he'll probably have what you want. AND take advantage of this special deal and we all will be winners.

CoCoPRO, PO Box 763, Ypsilanti, MI 48197   (313) 482-8128

WOW

OS-9 NEWSLETTER 1992 INDEX

OS-9 NEWSLETTER INDEX

Microware Gossip

It seems that Microware is getting into the gaming business for CD-I. Maybe they'll put a shame to some of those companies like the one that sounds like Microware but is a bit softer.

Also the February issue of BYTE is dedicated to CD-ROMs, and mainly CD-I, VIS and MMCD. Both VIS and MMCD use some stupid derivative such as MS-DOS and windows. While CD-I uses the Pirate of OS's, O S - 9

-- DON VAILLANCOURT;FidoNET --

Club Activities Report

Bellingham OS9 Users Group - Longview/Kelso CoCo Club
Mt. Rainier CoCo Club - Port O'CoCo Club - Seattle 68xxx Mug

Bellingham OS-9 Users Group

Unfortunately, our January meeting was a total bust. Our agenda looked good but destiny chose another path. It was like a comedy of errors, but not very funny.

First item on the agenda was the installation of *Dynacalc* in the GIMIX computer. The hard drive crashed last month and only the installation of *Dynacalc* and the *Sculptor* Database remains to return the system to its pre-crashed state; seem simple enough, but first you have to be able to log on to the GIMIX and I forgot the super user's password. No problem! We booted the computer with the system backup floppy and were able to look into the password file to view the super user's password and then rebooted the hard drive, gained access and began the installation of *Dynacalc*.

Unfortunately, the documentation for the *dynacalc* installation into an OS-9 system using terminals is uninformative, vague and lacking any hint of what we are supposed to do. A configuration program was run but some of the questions asked were beyond our understanding. such as: What is the character representing the first column location, row location? The second column location, row location? DUH?
You mean there is an ASCII character that represents a location on a screen? Am I missing the pixel or what? (Pun intended!)

Second on our agenda was to complete the printout routine in the PDS Database that was started by the Seattle 68xxxMUG. Did anyone remember to bring their OS-9 Manual.... Of course not. Fortunately we have the hard bound copy of *Basic09 Tour Guide* by Dale Puckett. Surely any help we needed about writing and opening files to the printer would be included there....**NOT!**

Third on our agenda was the completion of the 6309 installation to **Mike Pleas's** CoCo. Mike is also our Basic09 expert. Unfortunately Mike had a relapse of the flu and pulled out of the meeting at the last minute, so his project is tabled until next meeting.

As a diversion, **Rodger Alexander** demonstrated a new graphics software imaging program that displays 3 dimensional objects on a standard VGA screen without the aid of 3-D glasses. However, to view the objects you must look "through" the monitor by focusing your eyes beyond the front of the monitor screen. Two dots are displayed at the top of the screen for the viewer to concentrate on and adjust the eyes until three dots are seen. then the 3 dimensional objects become visible deep within the depths of the monitor. WOW! Unfortunately, I was the only one who could see it and have

been labeled crazy by those who could not.

February's meeting will feature almost the same agenda as this meeting, but hopefully we will have a better success rate.

← -- Rodger Alexander -- →
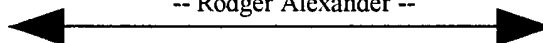
Port O CoCo Club

With each new year comes new surprises. The first meeting of the Port O' CoCo Club for 1993 had two. One positive, one negative.

Phyllis Armstrong has been doing our calling before each meeting to our list of about 40 people interested in the CoCo. There is a usual cross section of "Thanks, I almost forgot," "I'll be out of town Monday, but I'll try to make it next month," and "Oh, that's a topic I especially like, I'll be there." There are even one or two "I've sold my system. Please remove me from your list." But with each one of those we seem to find one or two new people who drift into the group. January's phoning seemed no different than any other month. But much to our surprise we had an exceptionally strong crowd. 18 people by the time the last one arrived. And even at that there were two who called later and said they had a bad sector in their cranial memory and forgot to attend.

So we started off with a crowd. Before we got to the fun things we had a couple business matters to discuss. First, our $10 to the State needed to be mailed for our corporation registration for 1993. **Bud Helck** is taking care of that and a set of by laws for us to have. He hopes to have those completed before spring.

We also discussed the Computer Fair at the Kitsap Mall in Silverdale which took place the weekend before. We handled the organization of the event for the second (and last) year. Not only did we handle the clubs, but this time we also invited retailers and computer services to have table space. For the two latter we charged a nominal fee. For this fee each company received free advertising in a mailing into 20,000 homes before the event, a supply of flyers to pass out to customers the week before, and a business card ad along with a list of local bulletin boards. Everyone seemed pleased with the arrangement and we wound up with about $40 in the club's account. **Terry Laraway, Tom Brooks, Gene Elliott, Buzz Jones** and **Donald Zimmerman** manned the booth for the two day event. Thanks for your commitment to the cause.

Upcoming is the Computer Swap Meet on the 30th. We will not be directly present anymore at these events because the organizers are now charging non-profits for space. The

$50 per event is beyond our means. But we will be indirectly involved because the Computer Bank Charity has voted to pay the fee and be at 7 of the events during 1993. So if anyone asks any CoCo questions you can be sure they will be handled!

The biggest surprise was a recent letter from Ed McMann himself. He wrote us to tell us that we, the Port O' CoCo Club, may already be the winner of $10,000,000.00. The drawing will be January 29th during the NBC Nightly News. Well, you can be damned sure that our form has been mailed in. And plans are already in the works for what to do with the $10 million. One suggestion is to have a fully outfitted tower for each person who comes to the February meeting. Another suggestion was to hold the February meeting in Fort Worth. While another suggested holding a Fest on a cruise ship, the USS CoCo, of course! You can be sure we will let you know when the money is being delivered!

Another topic of brief discussion was the establishment of some cause or another for the club to take up and underwrite. One suggestion was a small grant to a college student, say $250 a year. The driving concept behind the suggestion is that any group should not only serve itself, but also serve its fellows in some way. Since computers naturally work hand in hand with education maybe a scholarship of some kind would be a good idea to reach out into the community and serve those striving to better themselves and the fabric of goodwill in our area. Concerns were expressed that we had only $700 in the account and that our first project is the upcoming PNW CoCo Fest III. We should focus on that first. Also suggestions were expressed for various projects in different parts of the county. We may want to look at making a contribution to some project that is bigger than we ourselves can handle. The topic reached no conclusion, but will be brought up again in the fall.

The downside of the meeting was our unsuccessful attempt to get a phone line into the meeting room. You'd think we were caught up in an episode of "McGiver" trying to brake into a secret installation! We made some phone calls to people who were suppose to be in the know about multiline phones. We checked out the system the morning of the meeting. We had three men and a boy stringing wires and trying all combinations trying to get a simple dial tone. Nothing. So we were unable to demo Cloud's Corner, the BBS that's been so supportive to the OS-9 echo in our area. The next plan of attack is outlined at the end of this article.

The highlight of the meeting was a brief discussion and exciting demo of the MM/1 by the **Gonzales** brothers of Bremerton. They are the company reps for this area. They actually had company flyers to hand out. They also opened the case and showed off the guts of the system. As seen on a Sony Trinitron Professional Monitor, the graphics were stunning! Refer to their ad in this newsletter to request the MM/1 flyer and answer any of your questions. The MM/1 and the brothers will be at many of our future meetings. So we should have a continuing report of what is happening with this

next step in the CoCo family and some mind blowing demos as more software comes along.

During all of this excitement there was a fair amount of wheelin' and dealin'. **Bob Vail** brought a color plotter in that he wanted to sell and had it sold before he found a chair to sit in. More of the programs the club bought a few months ago were sold also. So it was a great evening all the way around.

February's meeting is the 15th. We will be starting at 7 p.m. with a brief business meeting and an update on our own system. We hope to have a hard drive in place by then. Then the meeting will move to a nearby home for a pretested and secure normal single line phone demo with Cloud's Corner. A caravan of cars will make the trip so directions are not needed. Just get to the Stock Market Foods location before 8:15 and you will be part of the parade to the ACTUAL demo of BBSs! As they say, the fifth times a charm!

◀ -- Donald Zimmerman -- ▶

Seattle 68xxxMug

The January meeting was of special interest to Amateur Radio Operators who have computers. **Scott Honaker** demonstrated telecommunications via HAM Radio or Packet.

Scott used a laptop PC portable computer for his demonstration because we did not have access to a monitor for the club's CoCo. Scott also brought along a very small, hand held, 2 watt, 2 meter transceiver plugged into a TCU or Terminal Communications Unit that plugged into the computer via the RS-232 port. Since the software is in ROM in the TCU it is possible to operate this setup on any computer that has a standard RS-232 port. OS-9 works very well for this. Of course you need to have an RS-232 Pak plugged in to the CoCo, the Bit-Banger port on the back of the CoCo will not work.

Operating the TCU was very similar to running most standard terminal programs. You can send and receive messages via packet radio repeaters to almost anywhere on the continent or even the world for FREE! There are even Packet BBS's that you can log on to and download files just like with normal Bulletin Boards that you call over the phone.

The word PACKET refers to how messages are sent over the air waves. Each message is broken down into small units of 128 or 256 bytes (I'm not sure of the actual size). These small packets of information also include error checking protocol so that if the message is not correctly received, the same packet is sent again until the receiving end confirms an error free reception.

If you wanted to send a message to your uncle in San Francisco, you could call up a local packet BBS and transmit your message with an address header indicating the final destination. The BBS stores your message and then relays it south automatically to another BBS that will do the same until your message reaches its destination. WILD!!

-- Rodger Alexander --

Washington State BBS List

COLUMBIA HTS. BBS
-- Lonview/Kelso --
RiBBS (FidoNET)
(206) 425-5804

DATA WAREHOUSE BBS
-- Spokane --
RiBBS (FidoNET)
(509) 325-6787

BARBEQUED RIBBS
-- Bellingham --
PC-Board (PC-Net) - CoCo Conference #5
(206) 676-5787

OS-9 TACOMA BBS
-- Tacoma --
RiBBS (FidoNET)
(206) 566-8857

ULTIMATE EXPERIENCE BBS
-- Anacortes --
RiBBS (MaxNET)
(206) 299-0491

OS-9 Newsletter
3404 Illinois Lane
Bellingham, WA 98226-4238