OS-9 Newsletter

Volume III Issue 8 Bellingham OS-9 Users Group August 31, 1992

B&B's PowerBoost *Patches*

Well, it looks like I was right about the F$GCMDir patch in OS9P2 being the main cause of crashes in the Booster for those running RiBBS, StG, and probably other programs that are heavy on modularization. Chris Burke checked it out and has provided the following patch. Bear in mind that it is possible (though not very probable) that your OS9P2 is so different from the original that the patch will have been effected at a different offset than the ones listed below. For this reason, I'll include the entire patch so that you may hunt for this sequence of bytes and come up with your own offset.

–Chris Burke checked it out!–

Booster Patch: OS9P2 - TFM in F$GCMDir

OS9P2 Offset	NonBooster	Booster	Correct
$0C34	$EE	$10	
$0C35	$A1	$86	
$0C36	$EF	$00	
$0C37	$81	$10	$08
$0C38	$EE	$11	
$0C39	$A1	$38	
$0C3A	$EF	$21	
$0C3B	$81	$20	
$0C3C	$EE	$07	
$0C3D	$A1	$12	
$0C3E	$EF	$12	
$0C3F	$81	$12	
$0C40	$EE	$12	
$0C41	$A1	$12	
$0C42	$EF	$12	
$0C43	$81	$12	

I should add that with this problem corrected, my system runs EXTREMELY reliably - even with all other Booster patches installed (including GRFDRV.) So far, six days without a crash.

Chris also mentioned another bug that I had not encountered. It seems that there have been some crashes when running Repack with the KRNL patch (Edition 16) (not sure if this is with or without the Booster patches.) According to Chris, this is a bug in the KRNL patch and not in his Booster software. This makes sense since later versions of the KRNL patch seem to have this corrected already. If you are having such crashes and....

if OS9P1 Offset	byte Equals	Change to
$083E	$85	$81
$0840	$27	$26
$09D7	$85	$81
$09DB	$27	$26

Feel free to spread this information around (per C. Burke).

-- Charles West --

Because I haven't been able to create a boot disk using cobbler, and have been too lazy to track down all the patches I have made to os9 to use os9gen to create a new boot disk, I just call booster from my startup file when ever I reboot. Because I call booster each time I re-boot, I did a search on the booster software looking for the sequence of bytes that replace those in OS9p2. If anyone is interested, you can patch booster directly, so this way if you like me call booster from a startup file that fix will be there in booster saving you the added step of patching os9p2 each time.

All you need to do is change the byte at offset $13CD in booster from $10 to $08. I used the following modpatch file:

l booster
c $13CD $10 $08
v

Then save a copy out to disk after modpatch finishes.

-- Carmen Izzi Jr. --

<<— Inside This Issue —>>

Burke & Burke's 6309 Power Boost PATCHES Pg. 1
Patch to *OS9P2* and *Booster* prevents CRASHES!
6309 *Power Booster* Bench Test Results Pg. 2
Speed comparison's with vs. without Booster Modules
"Hack" a 2nd port to your Deluxe RS-232 PAK Pg. 3
Simple hardware "piggy-back" modification
6309 Assembler BUG Alert Pg. 4
Chris Burke's memo to recipients of *The 6309 Book "XSM"*
File Allocation BITS, Explained and "set" Pg. 4
Set Allocation Table BITS to prevent OS-9 from writing to bad sectors. Rick Adam's ***SET-BIT*** software included.
Corrections: HD Interface (May'92); Tips from 'Terry' (Feb'92) Pg. 7
"Hacking" OS-9 Security on a GIMIX OS-9 Level Two Computer Pg. 7
How to become a *SUPER USER* via the "back door"
OS-9 Newsletter Public Domain Library (Complete Listing) Pg. 9
CoCo/OS-9 Club Activity Report Pg.11
Bellingham - Port Orchard - Seattle - Tacoma - Longview

6309 Bench Testing

by Boisy Pitre

Using my Timer utility which benchmarks programs by the system clock, I tested the speed of several popular commands.

These tests were run on a 6309-based 512K Tandy Color Computer 3 running at 1.9625 MHz; Seagate ST-157N-0 Hard Drive with Ken-Ton SCSI Interface. Although the actual speeds of these commands may be biased against the 1.789MHz clock in stock CoCo 3's, the time ratio between the two comparisons are constant. The version of PowerBoost used for these test was 1.0.

Keep in mind that the issue here isn't the speed of the commands themselves, but rather their speed as it relates to the patched kernel and managers vs. non-patched OS-9.

Note that all commands were either already in memory, or previously loaded to alleviate timelapse in loading from disk. The left column reflects the elapsed time without PowerBooster 1.0 installed, while the right column shows the same command run under the same circumstances.

Table 1

```
WithoutPowerBooster           With PowerBooster
--------------------------------------------------
   Command line:  cobbler /r0               *
Start Time:     15:26:34              15:28:30
Stop Time:      15:26:50              15:28:31
Elapsed:        00:00:16              00:00:01
              Increase: 1600%

            Command line:  mdir
Start Time:     15:25:48              15:28:12
Stop Time:      15:25:52        ......15:28:14
Elapsed:        00:00:04              00:00:02
              Increase:  100%

           Command line:  mdir e
Start Time:     15:26:01              15:28:19
Stop Time:      15:26:09        ......15:28:24
Elapsed:        00:00:08        ......00:00:05
              Increase:  60%

   Command line:  megaread </dd8           **
Start Time:     16:13:37        ......16:30:27
Stop Time:      16:14:50        ......16:31:37
Elapsed:        00:01:13        ......00:01:10
              Increase:  4.3%

   Command line:  rz </t2                   ***
Start Time:     16:18:08        ......16:24:05
Stop Time:      16:23:14        ......16:29:09
Elapsed:        00:05:06        ......00:05:05
              Increase:  0.16%

* The Cobbler command was benchmarked with a 30,535
byte OS9Boot file written to a RAM Disk

** The MegaRead command is a custom assembly
language program which reads 40 blocks of 25,000
bytes each from stdin.  This is NOT the same
MegaRead utility used by others for bench marking
purposes.

***RZ 3.10 was used to download a 69,971 byte text
file at 2400 baud (ViVa Modem 24) to a hard drive.
No errors or retries occurred.
```

COMMENTS:

These are estimates based on my particular system and are accurate within 1 second. Some commands (i.e. *Cobbler* & *MDir*) show a drastic improvement in speed with the PowerBoost patches applied, while other commands (i.e. *RZ* & *MegaRead*) show some to very little improvement.

Indeed, the command which showed the most dramatic increase was the *Cobbler* utility. This is due to Cobbler's heavy dependence on the F$CpMem call (used in three different places in the program, and that's not counting looped calls). I was so amazed at how much faster the command ran that I had to try it several times to be sure I wasn't dreaming!

MDir performs a good margin above the unpatched run. This command references the P$CpyMem call twice (again, not counting looped calls) as well as the F$GModDr call. One can definitely tell the difference when running this command, with or without the option.

I threw together my own home-brewed version of *MegaRead* (which probably doesn't resemble the "standard" version at all). Although it performed slightly better after the PowerBoost, Chris Burke's demonstration with his version at the ChicagoFest showed a vast increase. (If that version of *MegaRead* is available for copying, I would appreciate a UNENCODED version emailed to me. <hint>

Although ***RZ*** showed hardly any improvement (you could almost say "no improvement"), I DO notice that my modem transfers, especially my UUCP transmissions, are consider ably faster with Power Booster enabled.

Other commands that I noticed significant speed when using PowerBooster: ***idir, ddir, proc, pmap, paths*** (these commands are part of Kevin Darling's utility package). I have also noticed speed increases in the ***MV Canvas*** graphics program, particularly when moving the canvas vertically using the up/down arrow object in the Tools menu. I would be interested in hearing from others out there with their observations of the Power Boost's performance.

PROBLEMS:

The ONLY problem that my system has with the PowerBoost software (yes there is ONE problem) is random crashes at various times, especially when running disk intensive programs like *MAKE* and *REPACK*. I assume this has something to do with the exotic patches to my system's modules, but the problem only exists when PowerBoost is installed.

Let's not forget that this is version 1.0 of PowerBoost. Not nearly all of the drivers and the system modules that could be patched, were patched. To really squeeze all of the speed out of the 6309, it would take a total rewrite of the kernel, managers, and drivers. Also remember that the programs which were tested don't take advantage of the extra features of the 6309 and were written with the 6809 in mind.

With the implementation of Native Mode in upcoming releases of Burke & Burke's PowerBoost upgrades, another significant increase in speed will be attained. In other words, this is just the "tip of the iceberg." !

"Hack" a 2nd port to your Deluxe RS-232 PAK

by Bob Brose;Delphi

In reference to several users asking about a second serial port for CoCo OS-9. Bob Brose posted this article on CompuServe (CIS) a while ago.

Parts List:

6551 integrated circuit chip
1489 integrated circuit chip
1.8432 MHz Crystal
Optional: 1488 chip (See Instructions)

Hardware Instructions:

1. Remove the existing 6551 from it's socket
2. On the new 6551, bend pins 2, 5, 6, 7, 8, 9, 10, 11, 12, 16 and 17 up so they point directly away from the body of the chip.
3. Place the new 6551 over the old 6551 (line up the pin 1's) and solder the following top pins to the bottom pins 1, 3, 4, 13, 14, 15, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28 without getting any solder on the lower parts of the legs of the lower 6551.
4. Plug the 6551 back into it's socket. (if your 6551 wasn't socketed you can still do the mod but it will be harder to solder the 2 chips together on the board, be careful not to short any connections with solder blobs).
5. Connect a short jumper from the 74LS04 pin 9 to the top 6551 pin 2. This provides the select signal fro the new 6551 (FF6C-FF6F).
6. **3 Wire Port Option:** Connect the following pins on the top 6551 together: 1, 9, 16 and 17. This sets the CTS, DCD and DSR lines to low (true for them) which is correct for a three wire line with no hardware handshaking. If you want a full port, DO NOT connect these pins together

6a. **Full Port Option:** Connect pins 9, 16 and 17 to gates on the soon to be piggybacked 1489.

7. To pins 6 and 7 on the top 6551, solder a 1.8432 MHz Crystal. This is necessary as the two 6551's cannot share the same crystal because of the way they generate a signal from it. Alternately, you can make a crystal generator out of spare gates on the 74LS04 with one crystal and feed the signal into both pin 6's on the 6551's (leaving pin 7's unconnected) but this requires much more work and since crystals are only about $1, it really isn't worth it.
8. Piggyback the new 1489 on top of the current one making sure that the pins 1 line up, bending up all pins on the new 1489 except 7 and 14. Solder the 2 pin 7's together and the 2 pin 14's together making sure not to short out any other pins or traces.
9. Connect a wire from pin 3 of the piggybacked 1489 to pin 12 of the top 6551 (this is receive data).
10. The existing 1488 has 1 free gate which we will use for transmit data. Solder a wire from the top 6551 pin 10 to the 1488 pin 2.
11. Get your desired RS-232 connector (I used a 25 pin female connector like the original). Solder a wire from a convenient ground (I used the pad by the right rear mounting screw) to pin 7 on the 25 pin RS-232 connector.
12. Solder a wire from pin 3 of the 1488 to pin 2 of the RS-232 connector (this is transmit data).
13. Solder a wire from pin 1 of the piggybacked 1489 to pin 3 on the RS-232 connector (this is receive data).

NOTE: *Pins 2 and 3 can be reversed depending on whether you are talking to a modem or terminal.*

Testing:

Test out the RS-232 PAK by plugging it into a multipak (protects you from major soldering errors) and powering up your machine. If your computer doesn't act completely normal, turn it off immediately and recheck all of the connections against the above instructions. If everything is OK, try out a terminal program for the existing RS-232 PAK. If it works, proceed to the software mod section below, otherwise go back and check your work again.

End of Hardware Mods:

If you want to hook up other input status lines, the piggybacked 1489 can be used to hook up the 3 input status lines, CTS, DCD and DSR. If you are going to use this port with a CALL IN modem, you will need to do this. **NOTE:** See the 1489 data sheet for pinouts of the unused gates.

Also, if you need to hook up outgoing status lines like DTR and RTS, you will need to piggyback another 1488 on top of the existing one and connect it up. I'm using my second port for a terminal so none of the handshaking lines were necessary.

Software Modifications:

I use the port only with OS-9 so the changes are minor. You can use the port with RSDOS, but you will need to write your own software to do so. Remember in RSDOS, if you use both ports as interrupt driven ports, your interrupt routine will have to check both ports to see which one caused the interrupt as they are connected together (PC owners WISH they could actively share interrupts!).

For OS-9, I use T3 for my new descriptor, I did this by taking an existing T2 descriptor and changing the least significant byte of the port address at offset 10 (hexadecimal) in the descriptor to 6C from 68. Also you need to change the name of the descriptor, I did this by changing the high bit set "2", which is B2 at offset 38 (hex) to B3 (which is a high bit set "3"). Don't forget to verify the CRC and save out the new descriptor. Create a new boot with the T3 and you are ready to go. The name offset at 38 (hex) above will vary from one descriptor to the next because there are so many versions of the ACIA driver around. I use Bruce Isted's SACIA with great success and recommend it highly.

I routinely call in on T2 and then connect to another computer via T3 and it works completely perfectly.

6309 "ASM" BUG Alert

There is a bug in Version 1.00 of the Burke & Burke XSM assembler. This assembler is an ASM replacement for Level 2 OS9, which supports 6309 mnemonics, long variable names, and other extended features. Version 1.00 of the assembler was shipped with the first ~50 copies of "The 6309 Book".

The bug is that the assembler stops processing an argument list as soon as it encounters an undefined forward reference used as part of a more complex expression (because it doesn't properly find the comma). This causes problems during pass 1 in the processing of FCB, FDB, and FQB directives that use forward references; only those arguments up to the 1st forward reference are processed, causing a Phasing Error in pass 2. Here's an example:

```
TABLE  FDB  (AHEAD-*),(AGAIN-*)
AHEAD  EQU  *
AGAIN  EQU  *
```

This produces a phasing error due to the forward reference AHEAD. In pass 1, the assembler never processes the AGAIN argument and allocates 2 bytes for TABLE, while in pass 2 the assembler (correctly) allocates 4 bytes.

The work-around is to, when using forward references in expressions, put only one argument to each FCB, FDB, or FQB directive. Hence, the following assembles correctly:

```
TABLE  FDB  (AHEAD-*)
       FDB  (AGAIN-*)
AHEAD  EQU  *
AGAIN  EQU  *
```

Burke & Burke is working on a simple patch for Version 1.00, and is currently shipping Version 1.01, which corrects the problem.

-- Chris Burke --

File Allocation BITS

Explained and "set"

Operating OS-9 Level II on a hard drive is the only way to go. No floppy disk to worry about, a CMDS directory as long as your arm, a separate data directory for each application or category. With all that storage space available in an instant makes OS-9 a truly viable operating system. But with a hard drive and expanded storage comes added responsibilities and liabilities. A crashed disk, a bad sector, a virus. It can ruin you're entire year! Precautions must be practiced on a regular basis to avoid major problems that can sneak up on you.

An understanding of how OS-9 stores information on a disk becomes necessary in order to develop a library of utilities to maintain your system. Knowing what can happen and being able to identify clues or indicators on your drives is the first step.

The status of every sector on your drives is recorded on your disk in the allocation table. This is how OS-9 can quickly tell you how much free space you have on your disk, or know if a sector is available to write to or not. Keeping a status file is a quick efficient way to manage a large storage media. OS-9 would really crawl if it had to actually search every track on your disk to find an area that would accommodate a file that you wanted to save. Imagine how long it would take to load a file if OS-9 had to actually start from track Ø and search through each consecutive track and sector to find the file you requested. Sound like a Commodore to me! By keeping an index and status record at the beginning of the disk, OS-9 knows exactly where to go in an instant to find your record or locate a free area to store your files.

Unfortunately this ideal of indexes and allocation tables is not full proof because disk are not created perfect. Almost every hard drive made has at least one bad sector hidden away. Even floppies can develop bad sectors as they age or get bent, kinked or folded. Write a file to a bad sector and you have a bad file. Did I mention you should always back up your hard drive?

PRACTICAL APPLICATIONS:

Recently I downloaded several programs that check the surface quality of your hard drive. In other words they read, write, read and write to every sector on the disk. If the program finds a sector that doesn't pass the test, it reports it as a bad sector. Meanwhile, back in the file allocation table, every sector's status is kept track of by simply assigning one bit of data space to represent each sector on the disk. If the sector is free or empty, that representative bit in the allocation table is set to Ø. If there is data stored in that sector, the representative bit in the allocation table is set high or something other than Ø. If a sector goes bad the allocation table must be told about it by you. There is no automatic sensory system built into OS-9. When you have identified a bad sector, you need to go into the allocation table and set that sector reference to something other than Ø so that OS-9 will know not to write data to that sector.

To do this by hand is incredibly tedious, complicated and scary. Messing with the allocation table can really screw things up. First you need to know the bad sector location, then you have to convert back and forth from hexadecimal to decimal and back to hexadecimal and use a recursive type formula to find the location of the representative bit location in the allocation table, which is actually imbedded into a byte. Then you have to figure how to rewrite the byte so that only the proper bit within the byte is set to other than Ø. If you're off by just one, you may easily destroy a program or at least identify the wrong sector as bad and leave the bad sector still available for storage (or crashing). Fortunately for us, Rick Adam's wrote two short 'C" programs that do all of the above.

EXAMPLE: With Rick Adam's ***setbit*** program, the procedures becomes quite simple.

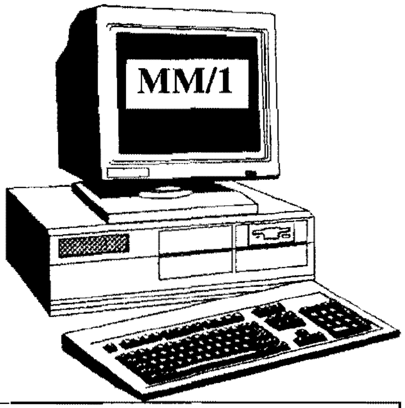
First: Use a disk scanning utility, such as ***dscan*** by Eugene Anderson and Tom Birt, ***scand*** by Steve Pollock, or ***stest*** by Bruce Isted. All these utilities are Public Domain programs. These programs can scan for bad sectors in a matter of a couple of minutes for floppies. If you have a large hard drive you may be in for quite a wait. The scanning programs provide you with the locations of bad sectors on your disk. If you notice that a bad sector exist within the area of a program or file, then you can assume that the program or file is bad. If you have a backup of your program or file, copy it to your disk or hard drive to another directory or under another name. Delete the bad program/file.

Second: As an example, lets say that one of the scanning programs above reports that you have a bad sector located at $99f. With that information you can use *setbit* by entering: **setbit 99f**. *Setbit* will report that either the representative bit in the allocation table has been successfully set or that it was already set, in which case you know that there is a program stored at that location, if so, refer to step one above, deleting the program and then run *setbit* again to reset the bit so the allocation table will not use that bad sector again.

-- Rodger Alexander --

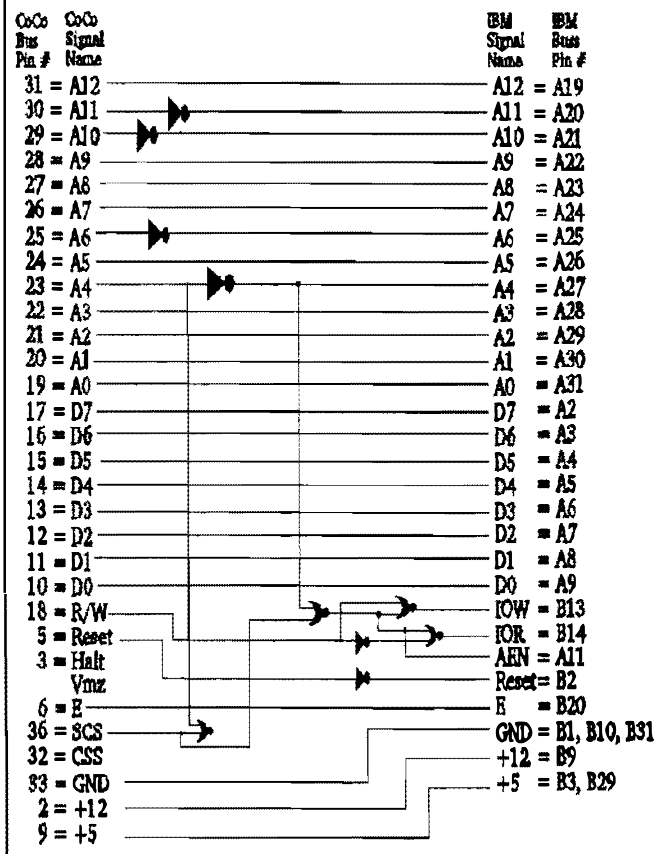
SETBIT by Ricky Adams:

```
#include <direct.h>
#include <stdio.h>
/* Copyright 1992 by Rick Adams */
main(argc, argv)
int argc; char *argv[];
{
        long isector, jsector, i;
        pflinit();
        if (argc == 2)
        {
                sscanf(argv[1], "%X", &isector);
                setbit(isector);
        }
        else if (argc == 3)
        {
                sscanf(argv[1], "%X", &isector);
                sscanf(argv[2], "%X", &jsector);
                for (i = isector; i <= jsector; i++)
                        setbit(i);
        }
        else
        {       printf("Usage : setbit <sector_number>\n");
                exit(0);
        }
}
setbit(isector)
long isector;
{
        static struct ddsect lsn0;
        static char data[256];
        static char masks[] = {0x80, 0x40, 0x20, 0x10, 0x08, 0x04, 0x02, 0x01};
        long sector, nsectors;
        short offset;
        char byte, bit, bitvalue;
        FILE *file;
        /* open disk */
        if ((file = fopen("/dd@", "r+")) == NULL)
        {       printf("Can't open disk\n");
                exit(0);
        }
        /* read lsn0 sector */
        fread(&lsn0, sizeof(lsn0), 1, file);
        /* get number of sectors on disk */
        l3tol(&nsectors, lsn0.dd_tot, 1);
        /* too large? */
        if (isector > nsectors)
        {       printf("Sector number too large\n");
                exit(0);
        }

        sector = (isector / lsn0.dd_bit) / 2048 + 1;
        offset = (isector / 8) & 0xff;
        bit = isector & 0x7;
        printf("File allocation map at sector 0x%lX, offset 0x%X, bit %d\n",
            sector, offset, bit);

        /* read sector */
        fseek(file, sector * 256, 0);
        fread(data, sizeof(data), 1, file);
        /* get value of proper bit within sector */
        byte = data[offset];
        bitvalue = byte & masks[bit];
        /* is it on already? */
        if (bitvalue != 0)
        {       /* display bit value */
                printf("Bit for sector 0x%lX is already ON\n",
                    isector);
        }
        else
        {       /* set bit */
                data[offset] |= masks[bit];
                fseek(file, sector * 256, 0);
                fwrite(data, sizeof(data), 1, file);
                /* display bit value */
                printf("Bit for sector 0x%lX is now ON\n",
                    isector);
        }
        /* close disk */
        fclose(file);
}
```



Corrections . . . . .

On page 3 of the of the June '92 issue, the schematic diagram shown is correct for Pat Pleuard's own custom made I/O buffer board, but will not work on a standard CoCo-3. The inverter on address line A5 must be removed and inverters put on address lines A11 and A10. The changes have to be made to make it work on your CoCo/Multipak:



What happened to the 'C' Tutorial ?

Due to the postage rates, I can only send 6 double sided pages in the mail for 29 cents. This limits the number of articles I can include each month in the Newsletter. The 'C' Tutorial consist of 14 lengthy chapters, including source code. Each installment consumes almost 1/2 an issue. However, for those who would like to have the 'C' Tutorial, I would be glad to share it with you. The Tutorial is FREE, but the cost of the disk and postage is $1 (25 cents for the disk and 75 cents for the postage). Mail your request to the OS-9 Newsletter, 3404 Illinois Lane, Bellingham, Wa. 98226

Apologies to Terry . . . .

In the February '92 issue, I headlined Terry Laraway's article as: **Tips from Larry.** Of course this should have been **Tips from Terry**. Terry never said a word (until I mentioned it to him). Please take a look at Terry's "FREE" add in this issue for ***CoCo Etcetera***. Anything you want for your CoCo....Terry has or can get for you.

"Hacking" OS-9 Security

This month, I was blessed (or cursed) to receive two GIMIX OS-9 Level Two "mini-frame" computers. The computers were used at our local high school to introduce students to computer keyboarding and mainframe system skills. Each OS-9 computer supported 15 terminals, a printer and a modem. The modems have long since "disappeared". When the IBM micro computers were introduced, 24 of the terminals were surplused (thrown out!). The computer instructor retired this summer and the only other teacher in the district who knew how to operate an OS-9 system computer was.... ME!

The system has been in disuse for the past three years except for an occasional Bellingham OS-9 Users Group meeting held at the High School. Since these computers were set up for multi-user operation, they were very password protected, and you guessed it.......The computer teacher had forgotten the password. Can you imagine my frustration to be given these two $10,000 OS-9 computer systems and not be able to gain access!

If you have OS-9 Level One or the Development System Package for OS-9 Level Two, you can duplicate my experience by typing **"Login"** at the OS-9 prompt. OS-9 will then prompt you for "User name?:". Enter: **User1.** OS-9 will print the Message of the Day (a text file in your SYS directory named **MOTD).** Now proceed with OS-9 as you always have......What? You say you keep getting error #214 (No Permission)????

(Cont'd next page)

The first error I received was when I tried to delete a student's text file. I had no permission. Why? Then I did a **DIR e** of the CMDS directory and looked at the attributes of the DELete utility:

```
Owner Last modified attributes sector bytecount name
----- ------------- ---------- ------ --------- -----------
    0 91/11/19 0957  ---e-rewr   B115       117 del
```

Public execution was set so how come it would not delete the file? Then I proceeded to a **DIR e** of the text file:

```
Owner Last modified attributes sector bytecount name
----- ------------- ---------- ------ --------- -----------
   f9 92/08/30 2205  -----r-wr    13C        48 sample.txt
```

The **ATTRibute** settings were correct for public read, but still I had no permission to delete the file so I used the **ATTRibute** command to make the file "Public Write" by entering: **attr sample.txt pw**. Now the **DIR e** listing looked like this:

```
Owner Last modified attributes sector bytecount name
----- ------------- ---------- ------ --------- -----------
   f9 91/11/19 0957  ----wr-wr    13C        48 sample.txt
```

But I still could not delete the file. Why? **(Because you are not the owner of the file or the SUPER USER!)**

Due to our habits gained from using OS-9 on a single user computer (CoCo) we never bother to look at the owner column. Looking back at the listings above you will notice that all of the files, including **DEL** and **ATTR** are owned by user #0, also known as **THE SUPER USER**. As long as the files are public attributed, I may read them, or execute them, but I may not modify the file, change it's attributes or delete/destroy the file. It doesn't belong to me. It's private property. It's SECURITY PROTECTED. And it is one of the major features of OS-9.

So how do I HACK this GEMIX OS-9 SYSTEM? If I could just get access to the password file I could find out what the super user's password was and gain entry that way. OR if I could change my user number from peasant number 92 to the super users number, which is zero, I would have the system beat.

Well, believe it or not, I must give thanks to TANDY at this point. Going through their OS-9 Manual in the Basic09 and Technical Reference section, I discovered a syscall referenced as **F$SUser (Set User ID)**. With the information I found I came up with the following program:

```
PROCEDURE superuser_poke
TYPE registers=cc,a,b,dp:BYTE; x,y,u:INTEGER
DIM regs:registers
DIM callcode:BYTE
regs.y=1
callcode=$1C
RUN syscall(callcode,regs)
PRINT regs.b
PRINT regs.y
PRINT callcode
PRINT regs.cc
PRINT regs.b
```

IT WORKED! Well, not the first time. Two obstructions still remained. For one thing, the GIMIX computer uses the standard OS9 format which is different from the CoCo's. Fortunately I have D.P. Johnson's **SDISK3** and the accompanying utility **disktype**, so I was able to format a "standard" OS-9 disk so that I could transfer my Basic09 file. The other obstruction was that OS-9 Level Two on the GIMIX does not have the Basic09 SYSCALL utility, and therefore could not execute my Basic09 program. I solved the problem by loading the Tandy version of Basic09 into the GIMIX and then ran my program. Once the program was executed, I edited the password file to give me user number zero (super user status) and I was in!

Later, I found out by reading some of the text files left on the system by previous student SYSOPS (SYstem OPeratorS) that they had accessed super user status by simply creating their own password file in the RAM DISK and then logging on to "their" password file. How clever! Oh well, my way was more creative!

--Rodger Alexander --

Bellingham OS-9 Users Group

Public Domain Library

Applications Public Domain Disk

AR	Archive/UnArchive file Utility
JERRYBENCH	10 x 10 matrix multiply benchmark
SIEVE	Self-timeing benchmark
SAVAGE	Bench mark
PIL●T	Pilot language processor
FILER	Miscellaneous Scratch Pad and DataBase
ADRLABEL	Sunple Address Label Maker program in Basic09
SPEEDISK	Floppy disk compression/repak utility
INVENT●RY	Home inventory program
EXAM	Basic09 Multiple Choice Quiz Program

Graphics Public Domain Disk

GROUP1-20	Clip Art graphic files f●r Tandy's Home Publisher
GR●UPTXT	Documentation file for Group1-20 clip-art files
VEFPRT	VEF graphic format print-out program
VIEW	VEF, MGE, CM3, 64D format viewer utility with window
MANDEL	Mandelbrot Graphic generator (BasicO9)
COCD3IBM	4 VEF files re: mounting CoCo-3 into IBM type case
VIEWGIF2	GIF format viewer with color "flickering"
VIEWVEF2	Displays "flicker" VEF files as well as normal VEF
AIF_MGR	Basic09 utility to create AIF files for Multivue
SUPERIKE	Icon editor. Mouse control & drag palette colors
PSHELL	Mouse controlled Graphics User Interface for those who don't like Multivue
PUBFONTS	Fonts for Tandy's Home Publisher

Graphics Public Domain Disk 2

CLICK1	Graphics clock time display
ICON EDIT	Creates icon graphics f●r Multivue
PAK	File management Archive utility. Pak/unPak into/from I
PRTPIX	Prints DESKMATE pictures to Gemini NX-IO printer
ICONS	Library of Multivue AIF and Icon files.
PIXLOADER	Laods and displays VEF format graphic files
OTHEL	Packed basic●9 othello game '
PIX	VEF pix viewer, Atari ST>VEF converter and docs file
FLTBRG	VEF pix from Flight. Simulator II
MGEPIX	View MGE format pics like VEF files
GIF0S9	View GIF format. pictures
PGCHARTS	VEF pix of charts
VEFNUDES	3 VEF pix of nudes
WATERFALL	VEF pix of a waterfa l l
SHUTTLE	VEF pix of NASA Space Shuttle
BITBUCKET	VEF pix of bits running around circuit board
BALLS	VEF pix of Universal Balls
MAXPIX	VEF pix of CoCo Max screen
●LIVER	MGE pix of Bloom County's Oliver viewing compuler
GOOFY	MGE pix of Walt Disney's Goofy

Miscellaneous Public Domain Disk:

WIND	Basic09 program thal pronipts for window parameters
FUNDOW	BasicO9 windows demonstration program
CLOCK	Basic09 script file t.o init.ialize a clock/dat.e
FUNOV	BasicO9 window overlay dem●nstrati●n/'utilit55 pr●gram
WIN■●WS	7 window scnpt files for use with ahove programs.
INTFIX	IRQ fix required when using the RS-232 PAK. Hack is done to the CoCo not the Multipak. VEF picture included

Patches and Memory Modules Public Domain Disk:

CC3DISK,VI 1	Allows disk I/0 under' different, formats (PC, At.ari)
CC3GO.SSYS	Required when using Speech Systems Real lime clock
CLOCK,SSYS	Required t● access Speech Systems Real Time cl●ck
FSTGRF	Kevin Darling's High speed patch for Level-II GRFDRV
UTRAACIA	Creates larger huffer area for faster RS-232 access
TSEDIT	Pat.ch to enable TSEdit, t.o displ ay 80 column Level-II
ACIA.PAT	AnotherpatchliyPeterLyalltoenlargehufferarea
■YNAPAT	Patches ■ynacalc to permit IO0 column display
BOOT80COL	Permit blue/white 8O column text screen on boat-up.
GRFDRV2	PatchesKevenDarlingspatchedGRFDRV for25 1ines,
GSHELL24A	Patches Gshell+ version 24a f●r faster operation
GSHELL	This is the GSHELL+ upgrade patches package.
GSHELLPA	Enhancement patches for GSNELL+ vers. 1.24 or 1.24a
GRFPATCH	Patches GRFDRV for 25, 27 or 28 line screens
SSPAK	OS9LevelTwoDriver&DescriptorforSpeechSoundPak Includes instructions modify the PAK for Level I I.
GFX■PAT	Patch to correct misalignmenl of the "Fill" name in the enhanced gfx2 (GFX■.AR) file on Utilit.y Disk 3.
KRNL	Patch t● Kernal t● give it OsK (OS9-68OO0) compatble filenames. "A" "Z" "a" "z" "O" "9" " " "-" "$"
DEDPLUS	PatchforDEDtoauto-recongizetheBitAllocationmapsect.
WINDFIY	Patch to prevenl Mullivue menu-bar "roll●ver"
MV2PAT	4 IPatch files: WindInt; CC31O; CC310.●rg; Gport.ipc
PARTGEN	Patches Burek&Burke BBFHDisk and provides new /H0 & /H1 descriptors to allow partitionung of the hard drive.

CACHE	IPatch to CC3Disk to permit cache capability to 1000 sect.
CC3DIS	Ipatch to CC3Disk to permit high density (1.2/1.4Meg) floppies when using the older 12vdc disk controller.
REPAKFIX	Corrects bug in OS9p1 modules that have been patched for 1/2Meg ram systems or modified to read 68K kernel names
DESK3.SCR	Modpatch Script file to Patch Deskmate's DESK module to use hi0resolution joystick
DMHELP	Text file explaining improvements of DeskMate3 over Desk Mate2. Includes patches to add linefeeds to Xmodem and to make DESK module to read the /DD drive instead of /D0. Patch to change DESK to default to any window instead of term. Also standard 6msec step rate/40 tracks and dbl side
MULTID	Source code to modify each Deskmate application and supply ICONS for use with Multivue
SERMIDI	Semi device driver module to provide midi output from the "bit-banger" port
SIMON PATCH	One byte patch to enable Level ●ne *Simon* utility to operate correclty under Level-2

Programming Public Domain Disk 1:

CC2	Lev-2 C Compiler double pass execution module
AUB	RMA(RelocatableMacroAssembler)Libraryfiles
CTUTORIAL	14 Chapter text with example source codes for learning "C".
DEVPAKCUPDATE	Inslruclional text on how to apply updales to lhe "C" Library from from the 0S9 Development. System disk.
CC2RAM	C Compiler execuiton module for compiling 'C' files in RAM

Programming Public Domain Disk 2:

CUBT.I	Kreider "C" Library
CLIB.I	Kreider "C" Library
CGFX.I	"C" Graphics Library
STDLIB2.C	Kreider's slandard "C" Labrary files
PASCALDEFS	Pascal Library files for Level Two
AUZIE_SCULPTOR	Sculptor source codes for creating/demonstrating Sculptor datahuase sh'slem
DEVSYS	User friendly interface for C, Pascat, ASM & RMA Assemhlers,
CSTART	Direct replacement for Microware's cslart.r LIBrary File.
MINILINT	l.It.ilil.y h.o prechieck C source fils for consh.ruue/synh.ax errors
SIGMON	OS-9 Level-II Debugger (belter lham Microware's DEBUG).
BLIST	Prints basic09 programs with pagination and breaks long lines.

Sound-Music Public Domain Disk 1:

PLAY	"Plays" digitized Mac audio files
ULTIMUSE	Let'.s you write and edit sheet music on a graphics screen, and play on any MIDI-equipped synethesizer.
LYRA	OS9 Demonstration of Lyra (Similar to UltiMusE)
SERMIDI	New /T1 descriptor that outputs bytes at the MIDI rate of 31,250 baud

Sound-Music Pullic Domain Disk 2:

"PLAY" FILES:

jtk.mac	aye.mac	hack.mac	beame.mac	bimbos.mac
damnit.mac	davel.mac	door.mac	genqtrs.pla	mrscott,mac
power.mac	jvulcan.mac	landing,pla	Liftoff.pla	disrupt.pla

Telecom PulllcDomainDiskl:

ACIAMapin	Filter to change control strings from a terminal into corresponding ansi strings.
ACIAMap0ut	Appendage from the ACIA Driver, gets control of Init, Read, and Write and passes them thru ACIA to the physical device.
DNLOAD	Copies input from specified device to Stdout.
UPLOAD	Capies Stdin to specified device.
BBS	BasicO9 Bulletin Board Modules
BIGT	Smart Terminal wuth xmodem, ascii protacols
TUBE.C	Copies charact.ers to/froni device.
DTERM	Dumb terminal wuth xm*dem protaca l.
KERMIT	Download/Upload prot.acol hetween t.erminal syst.ems
USER	BBS Sysop utility to view user files,
TELSTAR2.3	Excellent smarl terminal prograam (40kl for level-11.
XCOM9	Another excellent full function terminal program.
AUTOBAUD_TSMON	Detects incoming baud rate.
SUPERC●M	Level-II full function terminal program with xmodem

Telecom Public Domain Disk 2:

WIZPRO	BasicO9 - OS-9 Level 2 Terminal program

Telecom Public Domain Disk 3:

OSTERM208	Telecommunications package. Version 2.08 Requires RS-232PAK
NEWXCOM9	Telecommunications package. Bells and Whistles added to orig.
SCRIBE23	Offline message reader and auto posting
BITBANG	A device driver for the internal serial port permitting 1200 baud
MAX●WIK10	Off line message reader

Utility Public Domain Disk 1:

ARCHIVE	Creates destination directory and copies files to new dir.

MODUTIL	: Collection of enhanced standard OS9 utilities: *mbackup, mdate, mdump, mformat, mmakdir, mmfree, pmode, rep, split, unuse, verm, mxmode*
RSDOS	: File transfer utility from Radio Shack Dos format to OS9
SDIR	: Super Directory to enhance/replace standard DIR utility
AMPUTATE	: Forces the term call of the cache device driverreturning memory to the system and delete all file on the device
PALETTE	: Change screen/window palette (color) on the fly
AR	OS9 Archiving utility (version 1.3))
DLS	: Directory utility
PCDOS	: File transfer utility from PC format to OS9
WCONFIG	: Window configuration utility
BCOLOR	: Change background color: Bcolor <color>
FCOLOR	: Change foreground color: Fcolor <color>
BORDER	: Change border color: Border <color>
EATLF	: Deletes Line Feeds from downloaded files
PRINTHELP	: Prints SYS/helpmsg to screen/printer
QTIP	: Disk zap utility (self prompting) Version 3.1
RESET	: Keyboard "warm start"
CALL	. Calls commands repeatedly. Command list with lines on stdin.
IPATCH	: Creates a new file from an Original file and a Patchfile
SORTDIR	; Sorts directory entries in ASCII order
WMODE	: Returns status of current window
CLEARD	: Deletes all files from directory
LABEL	: Renames the Disk Name/Label (self prompting)
STRIP	: Strip or Add Character (line feed, carriage returns)
ZAP	Disk zap utility. Must be in 80 column mode.
LSH	: Unix type Directory Utility
SYSINFO	: Completestatus report of current window/terminal including current palette colors
CONVERT	: Converts decimal to hexidecimal equivalent numbers (visa versa) (only goes as high as 65000 decimal)
MAKPATCH	: Created patch file for use with IPATCH
TREE	: Directory utility, List hierchical listing of all directories, files, subdirectories, files
PAK	File Archive utility — NOT THERE
UTIL3	: Kevin Darling's merged utilities files:
PROC	: Enhanced "PROCS" utility showing I/O of each process
PMAP	: Memory map of each process location
PATHS	: I/O path of each process
DDIR	: Device Directory
IDIR	: IRQ Directory
DIRM	: Dir -e equivalent of MDIR
MMAP	: Memory map of used and unused 8K blocks
SMAP	Memory of pages in RAM
DMEM	: Memory dump by location instead of file name
COPY	: Updaled PD version of standard Copy utility
DIR	. Updated PD version of standard Dir utility
PURGE	: Deletes file(s) from a directory via user prompt
DASM	: Disassemliler for Level-I ASM and Level-II RMA files
DISKOPT	: Graphics DCHECK Utility (Self Prompting)
CUTS	: Coder/Decoder for transferring binary or ascii files on systems that do not support error checking protocols.
DAM	. Gives a graphic display of disk sector allocation map
NEWCRC	: Replaces the VERIFY command, boesn't require new filename.
DPOKE	: "Poke" hex code to the offset address of a file
DUPEFILE	: Duplicates any file on the same disk.

Utility Public Domain Disk 2:

MORSE	: Reads data from std-in and converts to Morse Code audio tones.
SDIR	: Unix "LS" type 'Super Directory' utility
SHELL21	: OS9User Group's expanded "shell+" module for OS-9
SCRIPT	: Script files for use with Shell+
ULDIR	: Converts files and directory names to proper case (Upper/Lower)
CRC	: Turnsoff the CRCcheck routine in OS9pl
HDKIT	: Peter Lyall's Hard Drive Backup/Restore Utility
BOOTSPLIT	: Separates merged modules into individual files
CC2	: Executive routine for microware C.Compiler on CoCo
D	: Single column (non-alphalietized) directory ulility
DEMODE	: Device MODE utility to change disk drive parameters
DMODE	: Device MODE utility to change disk drive parameters (not as complicated or complete as DEMODE)
DIRCOPY	: Copies files from ane directory to another
PRINT	: Formted I/O listing to printer device
PRINTERR	; Level-II version of micir'oware's PRINTERR.
SEPARATE	: Separates merged modules into individual files
DIR	: Enhanced Directory Utility with Sort and Search options.
DISKCAT	: Directory cataloging utility with graphics overlays.
DISASM	: Disassembler of machine code files (ASM and RMA)
DED	: Disk ZAP utility, Must be in 80 column mode.
ULIR	: Converts Directory names and files to proper Upper/Lower Case: Directories=UPPERCASE, Files=Lowercase
COMPARE	: CMP utility that compares 2 files (one in memory, other in ram)
RSFORMAT	: "DSKINI" standard 35 track RS-Dos format
RSRENAME	: Renames a file on a standard 35 track RS-Dos format
REBACK	: Basic09 enhancements for Peter Lyall's HDKit Utility
MAIL	: Multiuser logûn MAIL (scan/read/post) program//utility — NO DOCS [illegible]
DIRUTIL	: Graphics enhanced Directory utility with overlay windows
SBACK	: Re-write of REBACK for use with a mouse and Multivue.

Utility Public Domain Disk 3:

DOALL	: Utility that supplies a directory listing in pipe format, designed to take advantage of shell+ wildcard features
EASYEDIT	: Change device descriptors in your OS 9 Boot file.
GFX2	: Updated GFX2 for OS9 Level-II by Kevin Darling and Kent Meyers
INDEX	: Contains three programs that create, maintain and read an index of disk files
KUTIL	: To extract/modify/install the OS-9 Kernal
SDUMP	: Screen dump in background task.
SMOUSE	: Permits standard PC serial mouse usage from an RS-232 port
SUMMARY	: Utility to summarize spreadsheet data,
SWISE	: Utility to produce side wise printouts from dynacalc
SPEEDISK	: Disk repack/compression utility by Brian White. version 2.1a
VDG	: Generates VDG type screen for Level-I applications.
SCOPY	: Single drive copy util. for copying to dissimilar formats
PARTGEN	: Hard Drive Partitioning utility for Burke & Burke systems.
PS	: This is a "Point & Shoot" Menuing utility/application
KFORMAT	: Disk Format utility that is self prompting.
SMENU	: GShell graphics menuing environment/application.
MENU	: Text windows menu utility/application

Utility Public Domain Disk 4:

DATES	: Calculates the day of the year given the year, month, date.
DUALDOS	: PutsanRS-DOS sector on an OS-9 formateddisk
MDUMP	: Dumps memoy to printer or screen
CCUNZIP2	: MS-DOS de-archiving utility (PKUNZIP) for use on OS-9 (FAST!)
BRUIN	: A UNIX type backup/restore utility
SCOPY	: Single disk drive copy utility between non-simular formats
BOB VAN DER POEL UTILITIES:	
INTLEAVE	: Checks disk drive to find optimuim interleave.
DSETIME	: Causes system clock to be reset by DISTO hardware clock.
DIR	: A more complete version of Super LS by Conejo Computer
FFIX	: Listing utility that eliminates TABS and CR/LF
PHONE	: Phone dialer and "auto login" to BBS's
BLIST	: Display a BASIC09 program that has been listed to a file with pagination and properly formats the break up of too long lines.
HDIR	: A hard drive TREE Directory.
VEFSAVE	: Save an existing graphics screen to disk
VEFPRINT	: Print a VEF file to an Epson printer with a 640 dot mode.
VEFPRINT.23	; VEFPRINT for printers that print 23/216 inch line spacing
VEFPRINT.25	: VEFPRINT for printers that print 25/216 inchline spacirig
MAKEFILE	: Assembles VEFSAVE and VEFPRINT listed above.
CCHDISK	: Replacement for DISTO's hard drive adapter driver.

Utility Public Domain Disk 5:

CACHE	: IPatch that sets the cache size and prints the cache status for each drive.
DEARC	: Dearchives files that have been archived by MS-Dos/Unix ARC utility
DIFFY	. Creates multiple windows (with shells) on the same screen (OVERLAYS)
GSORT	: Replacement for original Multi-Vue GSORT
MAKDIR	: Replacement for original Microware MAKDIR utility
OS9ARC	OS9 version 1.0 of MS-Dos/Unix ARC utility
REBOOT	: Archive contains "cold" and "warm" reset
TCU	. LF & CR stripper and TAB to SPACE converter
VIRUS	: Maintains a file of specified files and their current CRC
WC	: Counts characters, words and lines in a specified input file
WHEREIS	: File finder utilities: *FF* and *WHEREIS*
ALIAS	: Creates a file-name that will chain to a specified execution module

Utility Public Domain Disk 6:

DISKCOPY	: Copies files from one directory to another or from one media to another regardless of the format.
DSCAN2	: Scans Hard or Floppy drives to locate bad sectors
SCAND	: Scans Hard or Floppy drives to locate bad sectors (Faster than DSCAN)
POP_V44	: Creates windows on the fly
FSEDIT	: Disk Editor
COMPRESS	: UNIX/VMS data compression utility
AR14	: Most recent update to the standard AR archive utility
CMPFIX	: Changes a CMP command output to an IPatch format file
SETBIT	: Used to set bits in allocation table representing bad sectors on disk
HEXDEC	: Hexidecimal/Decimal Conversion program that goes beyond 65000
MD	: Color Graphics Display of *MDIR*
MERGEMODS	: Provides user with a selection of files for merging that will fit into an 8K block

Utility Public Domain Disk 7:

CRON9	: Background task manager that wakes up every minute to check data file to see if it should execute a specified process/application
CRONTOOL	: Utilities for CRON9
SCULPT_SCRN.PAINT:	Self prompting utility to create custom input screen for Sculptor database
SS_MENU	: Shell+ script file that allows you to run various OS-9 Level-2 screen savers
EASYEDIT	: OS9Boot editor. Re-writes changes to your boot disk
FREE	: Faster replacement for Microware's *Free* command. Select sectors, bytes, kbytes, mbytes
WIDTH	: Select 40 or 80 column screen
JTFM	: File manager. Graphic display of directories and available commands. Mark file(s) for processing All standard command processes availalbe on screen
MELT	: Cute screen dissolve routine then returns screen to original state
MORE	: Text viewing utility
VU	: Better text viewing utility
SDUMP	: Dumps screen to printer by holding down ALT-CTRL and SHIFT
TAR	: UNIX file compression utility — NOT THERE
STREAM	: Fast Hard Drive Backup Utility. 20 Meg Drive in 2 hours
CDR	: File manager. Fast text screen display. Permits Copy, Delete, Rename and Listing of files

Word Processor Public Domain Disk:

UEMACS	: "Micro E-Macs" Unix Text Editor
ED16	: Simple full screen text editor (version 16) for Level -II
MROFF	: Text formatter using Word Star DOT "." formatting commands
SLED	: Text processor with on screen help windows
ED20	: Graphics screen editor for Multivue environment
ED31	: Graphics screen editor. (Laest version)
XLED	: Complete all in one text editor. 30K buffer
WP	: Complete word processor version of ED20 above

Subscribers to the ***OS-9 Newsletter*** have free access to our Public Domain Library. Either send a disk with 75 cents postage along with the specific disk that you want OR send only $1 for each disk that you want ($1.25 for 3-1/2in 720K disk). Please specify desired disk format for 5-1/4in. disk: 35 or 40 track, single or double sided. Mail to **OS-9 Newsletter, 3404 Illinois Lane, Bellingham, WA. 98226**

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

Our August meeting was consumed by the setting up, installation and "hacking" of the two GIMIX OS-9 Level Two systems assigned to Rodger Alexander's classroom at Fairhaven Middle School in the Bellingham School District..

Craig DuBois is one of the main technical engineers for General Telephone assigned to the Whatcom County area. They recently installed new equipment and Craig was able to retrieve an electronics cabinet that houses both GIMIX computers in a vertical position.

We literally took the computers apart, examined all of the chips and plug in boards. We discovered that the 5 serial boards each had their own 6809 processor. CRC checks are done by a separate hardware plug in board. 2 Megs of RAM is accoplished by 4 plug in boards full of 64K ram chips. Very, Very WARM.

Originally, each computer had 15 terminals plugged into them. Now there are only 8 terminals left. We checked the wiring harnesses and checked out each terminal.

One of the computers went dead and by swapping circuit boards we were able to discover an intermittent short between boards due to lack of support bracketts in the design. Eventually we got both machines back up and then turned our attention to "hacking" into the password file so that we could obtain "SUPER USER" status. (See page 11.)

The computers are to be used primarily for testing student's knowledge skill in different subjects. Our next objective is to find or write appropriate software.

Port Orchard CoCo Club

No specific presentation was made at the August meeting. Instead a report on the club obtaining a Non-Profit Status for liability purposes was given by Donald Zimmerman. Turns out to be much simpler than first thought, and only a yearly registration fee will be necessary. No lawyer!

The remainder of the meeting was devoted to talking about future meetings and it was decided that a full year calendar be created with the dates and topics printed on the calendar and a copy of the calendar be distributed to each member.

Input on what events to include on the calendar is being requested from everyone at the next meeting. Completing the calendar will be next month's major objective.

In the area of old business. All of the '92 CoCoFest T-shirts have been sold and only a very few mugs are left.

Seattle 68xxx MUG

The August 4th meeting featured the successful completing of the 6309 installation started last month (JULY) by Rodger. Burke and Burke's *"Power Booster"* software patched most of the OS-9 system modules to take advantage of the 16 and 32 bit register in the 6309 chip. The most obvious performance improvement was the MDIR listing to the screen. Much faster than with the original 6809.

Roy Chapman brought his floppy disk drives that he could not get to work properly since we configured the drives for him several months ago. We discovered that his old vertical 35 track drive was malfunctioning, preventing him from writing files in both RS-DOS and OS-9, although he could format disk. Roy's other disk drive was a model FD-502 and was working fine. Roy had also just recently purchased (from the Bellingham OS-9 Users Group) a 3-1/2in 720K drive. So it was decided to make the FD-502 drive equal to /D0 and the 720K drive equal to /D1. Unfortunately we did not have a 5-1/4 inch mounting frame for the 3-1/2 inch drive. We also cannot locate a 5-1/4 face plate for the Epson Drive. These 720K drives were designed to be used in lap tops and finding an appropriate face plate may be a problem. If anyone reading this article has a face plate that they think would fit, please get in contact with Rogder Alexander at (206) 734-5806 or Scott Honaker (206) 453-6002.

No report available for Mt. Rainier CoCo Club or the Lonview Kelso CoCo Club.

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

The *OS9 Newsletter* is published by the Bellingham OS-9 Users Group. Rodger Alexander, Editor. Subscription rates: $10/12 issues, $6/6 issues. Make your check payable to *OS9 Newsletter*; 3404 Illinois Lane; Bellingham, WA 98226

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