OS-9 Newsletter ©

Volume III Issue 10   Bellingham OS-9 Users Group   October 31, 1992

Getting the OS-9 Attitude

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.

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, have Level 2 on a CoCo3. There is no comparison between a 32 or 40 column screen and an 80 column screen, but for strange reasons 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 computer; how do you get OS9 to use it? There is an article by RICK ULAND entitled "The First Step" included in this newsletter. 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 System Modules database. 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. As of this writing the highest version is v2.1. The sooner you can arrive at an updated system the better off you'll be.

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) thrown in. OS9 is an 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

<-- IN THIS ISSUE -->

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 hierarchical 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 categorizing!

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 Ø
     ┌───────────┐
 Commands      Data
     │           │
   cmd 1       cmd 2
```
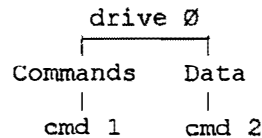
You start with a "/" for the top level /
You add a drive number
/d0
You add another "/" for a level within d0
/d0/
You add the commands identifier
/d0/cmds
You add another level indicator
/d0/cmds/
Finally the name of the word processor
/d0/cmds/wp

In BASIC you enter RUN"WP". In OS9 you enter /d0/cmds/wp.
command. (Nothing magic about the "3". There can be more levels.)
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" 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 /d files without the full command line and all the files you decompress (de archive) would be stored in that same directory.

(Still mor:. 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"** - path name 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 "path name", not "pathlist". Consider a path name a segment (between slashes) of a pathlist. Let's look closer at errors. Forewarned is indeed forearmed.

Usually errors are the bane of the OS9 beginners existence. 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 - path name 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, but 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. 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. Often they're as full of questions as you are, but they have answers. At the other end of the spectrum there are some very expert and brilliant programmers who know "everything there is to know" about OS9. Too 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, Compuserv or some other Bulletin Board. Maybe they wrote 2 pages of documentation 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 have been 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 and/or data storage space. When that space is limited, you cut something. What should get cut?

A very appealing aspect of OS9 is the ability to snap from one program to another with the push of a button. But with space and CPU speed limited, how many programs of what size, power and function can you squeeze into a little CoCo? Working Delphi or some other Bulletin Board is nice. You run a terminal program. Maybe you have an editor to generate messages which are stored. You download a file to disk or ram disk. Maybe you print out documentation in the background. Fun!

But what are you downloading? A spreadsheet that will hold all your small business bookkeeping for a year? Will you run that at the same time you have a memory gobbling graphics program in place? And the graphics user interface you use? And the big game you didn't finish last time you played?

Then, perhaps more importantly, there is the 6809 CPU a terrific little chip that could blow the doors off everything when it first came out (and later). DECB users who've been around a while remember when everyone was scrambling to get things running at double speed - (still less than 2 MHz). If one program can need that, what about several, running at once? Some programs take little CPU time. Some take a lot more, like when there is much disk I/O. Special disk controllers can help a lot, but that is only one of the angles.

How successful multi-tasking will be for you will depend on what you'll be running. Don't plan on having three

action games running at the same time.

With applications growing to accommodate users demands for computing power, more memory and speed is needed. If you're a purist of the 68xx genre, the new 68xxx computers may be interesting. They have many times more memory and speed than CoCo. You'll find programmers right on Delphi, Compuserv, Internet and Fidonet who are working day and night to develop applications for these computers.

Take a look at things. If you're a budding programmer, learning OS9 can be very educational. If you're a hobbyist you can do a lot of experimenting with OS9 software and utilities. You can even run some of those very good applications programs. Or power a rock concert via *Ultimuse* and a MIDI!

A little Body For the Attitude

So far, I've given you just a peek at how life can be in the OS9 world. Some things I've described may be very unlike what you ultimately find to be your specific experience. But I have been more interested in the general idea of things. Now I'll touch on a few points that will probably effect everyday computing more. Still, this isn't an OS9 "how to", just a whiff of its perfume.

Why is it so important to have so many levels of directories? Firstly, as I mentioned, there is a lot of stuff to organize.

With people writing constantly there are new "tools" coming out almost every week! (I have 16 clocks on file and that isn't all of them.) By "tools" I mean utilities to do jobs, etc. Earlier I used "del" in an illustration. That is a very simple tool. "List" is a fairly simple tool. "Free" is like free in DECB. "Mfree" is like "MEM". (There ARE differences between such similar DECB and OS9 commands but the general idea is the same). Improved versions of various commands are written, often somewhat larger. Most veteran OS9 users have an entirely new shell - SHELL +.

Other commands are replaced if the user prefers something different (and if

an alternative is available). But you really can't just commands. There WILL be a time when you need at least some of them if you stick with OS9 long! So you keep copies of them. Probably on a separate disk. But if you happen to want to use both versions you might want them on one disk. You can't do that with DECB if you use the same name, but it is easy with OS9, because you can put them in different directories.

You might want to put them on the same level in differently named directories. You could have, for example, CMDS1 and CMDS2. Then it would be possible to have one version of (perhaps) *ATTR* in one and another version in the other - (/d0/cmds1/attr or /d0/cmds2/attr). How much of this you can do might depend on your own memory.

Or you might want to put them on different levels. You could actually have a CMDS directory listed under the regular CMDS directory (/d0/cmds/cmds/attr), although it would probably not be a very good idea. A "dir" could confuse you if you didn't remember which level you were looking at.

Another reason for various directories is the wide variety of data you can (sometimes must) have on disk. Frequently you will see directories named SYS, DOC, MODULES or ARC. SYS might have data essential to the function of an assembler. It might have "help" or "error" messages. "DOC" would probably have documentation for programs or commands. "MODULES" would likely have descriptions of the parameters of your various I/O devices such as your printer or RS-232 PAK (40 isn't unusual). ARC would probably have archived (compressed) copies of programs, data, or other files.

As mentioned, directories CAN be "stacked" many levels deep. One communications program stores dialing information like this:

/d0/sys/dial/filename.

A MultiVue screen icon might be

named like this:

/d0/cmds/icon/icon.app.

I saw an example in which Basic09 commands were on the same disk with others. Under the regular CMDS directory was BASIC09, and under that was another CMDS. To copy it you could end up with this line:

copy /d0/cmds/basic09/cmds/filename
**   /d1/cmds/basic09/cmds/filename.**

(Ed.Note: This command is entered as one entire line with only a space entered between the two printed lines.)

(NOTE, above, that I have put directory names in capital letters but the command lines in lowercase. If you have many things on the same level, caps make a directory stand out on the screen. When you set up a directory (using MAKDIR) you can just enter the name in caps. After that you need not use caps to call it. In most cases, OS9 accepts lowercase keyboard input for the system. Programs run under OS9 make their own decision on that).

(If you see the reason for multiple directories, we can go on.)

At some point you will probably come across the phrase "Unified I/O System". So far I've referred to pathlists only as long paths through the maze of directories to files on your disks. But paths can access other things. Via pathlists you can list a text file to the screen or you can list it to your printer or to a screen in another window to be seen later! Via a pathlist you can read input from your RS-232 PAK (and therefore whatever it is connected to). You can run a program in a window you don't see. These things can happen because of this "unified" system.

In section one I mentioned that most commands are usually kept on disk. When you need, for example, the command "format" it will probably be on disk. When you enter "format" the computer loads it into memory and executes it (but not by firing squad) then unloads it when it is done with it. But you can also load commands into memory manually, then unload them when you wish. (Computereze for "unload" is "unlink".)

In section one I mentioned that misspelling something in a command line could cause an error such as 216 (path name not found). Something that you can do in OS9 does make life easier when this and other things happen - the command line buffer. What you enter as a command line is saved in memory. You can then repeat it with the appropriate key presses. As it comes, OS9 has an command repeat that is activated by **CTRL-A**. Before you press <ENTER> you can do some overstrike type editing.

With one very popular new version of this you use right arrow or shifted right arrow instead of CTRL-A. You can easily key over to the Offending characters, change, insert or delete them and repeat the   command. This is the "SCF EDITOR" I mentioned at the beginning of this article.

This saves MUCH time over typing in whole lines again, especially when dealing with long full pathlists. If you're not the best typist, it can be a life saver. It also comes in handy if, for example, you have several similar files to copy. By changing only the actual filename part of the pathlist you could move several files without retyping much.

On the other hand, when in doubt about where you are, you can usually enter a full pathlist to "feed the hungry computer". (Sometimes, specific programs might not allow that).

There are other things that can really frustrate the beginner. One thing is the time it takes to do certain tasks according to the manual. For example, at some point you will wish to rewrite your startup file. Beginners haul out the "build" command and start a new startup file from scratch as instructed by the manual. One tiny error and you're back to deleting, renaming, rewriting or what all, from the top down.

Startup is actually a text file called a script. (This may be in the manual somewhere, but I've never found it. It is definitely not in my indexes). Each command is on a separate line. If you can get hold of a line editor such as SLED from Delphi, (or a word processor) you can load such a script as "startup" and add or subtract at will.

Just loading one and looking at it will take much of the mystery out of it.

Always press <ENTER> after each command as you would on the keyboard. Just as in regular word processing, this will start a new line for the next command.

If you wish to keep a copy of the original startup file you can rename it the way the manual suggests - before starting on a new one. Then when you are finished with the new one you save it to disk as   "startup" just like the original.   If your line editor has an overwrite feature (such as the one in SLED) you can just write the file into the space where the original was. Once you get comfortable with the system you will probably do this.

The "edit" utility that comes with OS9 can do this job. If you find it easy to use, by all means do so. But with a regular text editor you can see it all at once and use the arrow keys to edit.

Since a script is just a series of commands in a text file, you can write different scripts to do different things and call them just as you would any other command. This is a good way to set up a series of commands you use repeatedly.   If a particular program calls for doing several things to prepare for it, you can probably do them in a script.   You could put your preparatory commands in it, save it in the "cmds" directory as "pstart", for program start, and do the job by just entering "pstart" (assuming your chx was set to "cmds").

Scripts are really very easy to write. You'll see.   But all is not rose petals here either. Sometimes, even when the script lines appear to be perfect, a line will just not work.   Swapping lines sometimes makes a difference.   Script lines do not give error messages. Shell+ will automatically get a script from the execution directory (set by chx).   You can NOT load a script into memory like other commands unless you upgrade to Shell+ v2.0 or higher.

I have tried to say (perhaps warn), in a simple way, how complex OS9 can be. (Even the explanation has become

somewhat complex.) I have tried to offer some preparation for dealing with that by covering an area or two that has been especially problematic for the new OS9 user - the manuals can be very frustrating sometimes.   Perhaps the best preparation is to tell you that this is just the "tip of the iceberg."

The Firing Line

Since directories and the **CHD** and **CHX** commands are at the center of a lot of beginners problems here is still more to help make things clear. When you type in a line like:

list /d1/modules/bootlist

the system looks for those names on the disk.   It looks first for "list" in the execution directory then looks for a descriptor named "d0", a directory named "modules" and a file named "bootlist". It must have the numbers that these names represent, and it scours one or more disks looking for them in logical ways. But when you use **CHD** and **CHX** those numbers are stored for quick use. If the "CMDS" directory is, for example, on line 1 to drive 1 and on sector 872, the **CHX** command stores that information. It is as if a big arrow were pointing to that sector. After that, any time the system "sees" a command it jumps to those numbers - that sector. That is why it cannot look for the "CMDS" directory by name.

Here are some typical command line problems:
Say you have put a system disk in drive ● and issued a "CHX /d0/cmds" command. Then you enter:

copy /d0/sys/stdptrs /d1/sys/stdptrs
Things will probably be okay.   But if you were in a hurry it may be:

cpy /d0/sys/stdptrs /d1/sys/strptrs
The system cannot find "cpy" and you get a 216 error.

If you put the disk in but forgot the **CHX** command, a perfect command line will probably give you a 215 error - bad path name - because the system jumped to sector so-and-so and there was no "CMDS" directory there!   See?? Right there is where a lot of frustration comes in. (Sometimes this may work, because if the command is in memory [where the

system first looks] it will work anyway.)

copy /d0/sys/stdptrs /d1/sys.stdptrs
isn't a hard mistake to make if you're not a good typist. This will cause the system to store a file in the directory last indicated by CHD and name it "sys.stdptrs."

copy /d0/sys/strptrs /d0/sys/stdptrs
will probably give you a 218 error - file already exists.

copy /d0/sys/stdptrs /d1/sys/stdprts
will not give you an immediate error. But when something looks for "stdptrs" and there is nothing but "strprts" it is 216 time again.

Believe it or not, if you have set **CHX** to **/d1/cmds**, for instance, and you decide to use a command (let's say "list") which is in your /d0/cmds directory you can enter something like:
/d0/cmds/list /d0/txt/message.

Lastly let me touch on redirection again. While making up a new bootfile (OS9Boot) I had some problems. I decided to use the "ident" utility to look at what was in the file. With 30 or 40 or more modules in OS9Boot you just can't get the idea on the screen. So I redirected the output of the ident. A normal ident on a single module might look like this:
ident /d0/modules/cc3disk.dr
By redirecting output like this:
ident /d0/OS9Boot >/p
I had all of the modules information sets printed out in hard copy instead of the screen.

You don't have to be typing command lines to get errors. Some utilities REQUIRE certain CHD settings, and if you just forgot to do one the utility could be getting the wrong information or none at all.

To really bring your system up to date, Rick Uland is distributing some very cheap sets of all the things you need to make a new system out of what you started out with. Contact him on the OS9 Sig on Delphi.

This is really a good idea, because you can spend days, weeks or even months acquiring all the patches

and getting them flawlessly into place. AND you will find that some of the very good public domain software available does not work well unless the original bugs are eliminated from the operating system itself.

I'm going to mention something Rick suggested, so you can "chew" on it. It comes with a sort of catch 22 built in so be careful how you size it up.

Although the OS9Boot file must be in one contiguous file, not stored in various parts of the disk, it does not need to start at the beginning of the disk. So you could put your commands directory on the disk first - on all your system disks - and the pointer set up by the **CHX** command would be the same for every disk. Therefore you would not have to reset it except for times when you used /d1.

The trick here is that by the time you might feel you have the expertise to do this you would probably have several system disks set up the other way. Those disks would require "CHXing" as usual and the mix might not be worth the effort.

-- Jim LaLone;Delphi: TERMITE --

If you need help beginning OS9 this tape is for you!

Using the original Radio Shack OS9 Level II Operating System disk set, you will be guided in creating an OS9 Boot disk which will boot up with an 80 column, type 2, terminal window, and will configure your 502 style drives as 40 track, doubled sided.

You will be instructed in the use of the following commands which are furnished with the original Radio Shack OS9 Level II operating system disk set:

format   backup   load   build
modpatch   chd/chx   edit   os9gen
makdir   cobbler   dsave   copy

The tape assumes that you have the Original Radio Shack OS9 Level II operating system disk set, a CoCo III, CM-8 monitor, two 502 style 40track drives, and several unformated disks. Tape is VHS format and speed is SP. Tape plays approximately one hour.

Orders may be sent to:
David Wordell
833 Woodhaven Lane
Grand Prairie, TX 75051
Tel (214) 264-3610

Include your name and address and $14.95 per tape. Money orders orders only please. Orders will be shipped immediately upon receipt.

Q & A's

Q: Can anyone tell me the best way to install a smartwatch?
-- Daniel Hauck --

A: *Just unplug your rom chip in your disk controller and insert the SWATCH. Reinsert the rom chip into the SWATCH chip! That's all there is too it! There is a file called SWATCH.AR that should be vailable on many boards. That will take care of all your needs if you can find it. If not I would be glad to send you a copy!*
-- Alan Dages;FidoNET OS9 Echo --

Easy way to make a Boot Disk

The main problem with OS9 as issued by Tandy is their stupid insistence on adhering to the CoCo 'standard' of 35 track, single sided drives and VDG (very dumb green) 32 column video. Fixing it is a little bit of a pain; you'll need 3 blank disks and an hour of work. At least it only has to be done once! After enduring this procedure, your CoCo should boot to 2 80 column windows in RGB mode, with 2 (or 3) 40 track double sided drives @ 6ms step rate, with your choice of serial port and printer baud rates.

Make a copy of the basic09 disk using backup (you can even do this from DECB [Disk Extended Color Basic] if you want), then put the new disk in drive 1, and the system master in drive 0. Type:

chx /d0/cmds;chd /d1/modules
edit bootlist

When you see "type:", you always press "ENTER" at the end of a line.

Bootlist is just that, a list of the modules to be used when making a custom version of OS9Boot. Since the version Tandy included is pretty lame, we are going to change it with *edit* to use 40 track drives and the windowing system. Since you entered **"edit bootlist"** you should now be seeing **E:**, the edit prompt. Type:

c*/35s.dd/40d.dd/
c/vdg.dt/win.dt/
q

Q saves the changes and returns to OS9: prompt. Now to load some stuff into RAM to free up the drive the system disk is in, type:

load format;load os9gen

Remove the system disk from drive 0 and insert a blank one in it's place. Type:

format /d0

When the formatting is done, type:

os9gen /d0 </d1/modules/bootlist
makes OS9Boot using bootlist

Now, remove the BASIC09 backup disk from drive 1 and replace it with the System Disk! Type:

chd /d1; chx /d1/cmds; dsave /d1 /d0 ! shell [copies rest of disk]

We have to use *dsave* here because backup would clobber the new boot just made. This is a good time to take a coffee break. *Dsave* will take something like 5 minutes to copy the entire system disk file by file.

That's version 1 - a 40 track capable boot stuck on a 35 track disk.

Press reset once to reboot the system. It should come up in a 40 column term screen.

Now there are a few more things to change before making version 2 - the true 40 track boot disk. First to change term screen to 80 columns and set the 6millesecond stepping rate for the disk drives. To do this, we'll use *modpatch*.

Modpatch uses a file that lists the changes to be made. To create this file, we'll use edit again. Type the following - (Note each line between edit and q starts with a space).

edit patch
l term
c 002c 28 50
c 0030 01 02
v
l d0
c 14 00 03
v
l d1
c 14 00 03
v
q
Now type:
modpatch patch

There are a few last changes - permanently setting printer and modem baud rates. In the following lines, replace # with 3 for 1200, 4 for 2400, 5 for 4800, 6 for 9600, or 7 for 19.2K baud.

xmode /t2 baud=# [makes changes
xmode /p baud=# in memory]

Now everything is set up in RAM. All that's left is to save the final version to disk. **Format one last blank disk in drive 1.** This time it should verify all they way to 4F and probably show $5A0 sectors, signifying a 40 track double sided disk.

cobbler /d1
transfers OS9Boot from memory
chd /d0
dsave /d0 /d1 ! shell [rest of disk]

When *dsave* gets done, reboot with the new disk. This disk has more than twice the room the old one did; so the Basic09 stuff will fit on it. Copy it over as follows.

Assuming new Master in /d0, Basic09 in /d1, type:

chd /d1/cmds
copy Basic09 /d0/cmds/Basic09
copy Runb /d0/cmds/Runb
copy gfx2 /d0/cmds/gfx2
copy inkey /d0/cmds/inkey
copy syscall /d0/cmds/syscall

You'll note I didn't include gfx- it's only for CoCo1/2 compatible Basic programs.

One last thing you'll like is a second window started automatically. (And I guess RGB mode would be nice [assuming you are using an RGB monitor]). Once again, time to edit! (Assumes Master in /d0) Type:

chd /d0
edit startup
+*
montype r
iniz /w7
shell i=/w7&
q

(Finally) Done!

P.S. No attempt was made to explain what every line given above does, since a reasonable explanation would swell this file to unmanageable size. The idea was to just get the job done, and figure it out later.

Easy way to make Boot Disk. . . . . . .*Cont'd*

This last section is a selection of common problems

OS9 has the equivalent of two DRIVE(x) commands **chd** and **chx**. The first points to current data directory and the second to current execution dir (usually a CMDS dir).

They are a little more complex than DRIVE(x), due to the directory tree system used by OS9. Not only do you have to repoint to use a different drive, but also when using a different disk in the same drive, since its directories may be in a different place on the disk. A 216 or 214 error often means you've forgotten to do a **chx** or **chd**.

To avoid the need to **chx** or **chd** all the time, you can use a full pathlist to get a file. Say you're pointed to a disk in drive 1, and need to run *something* off of the system disk in drive 0. **/d0/CMDS/something** will find it.

Another reason for a 214 error is that the file execution attributes are not set. Use **attr filename e   pe** to make it runable.

OS9 doesn't have an **llist** command, instead you have to redirect to the printer. **list file >/p** will do that. Most commands may be redirected. To get a printed directory **dir >/p**. And so on. If you have   a long file to print, you can do it in the background by adding an ampersand:

list file >/p&

One other neat redirection: You can list a file or print a directory to another window simply by redirecting the output: **mdir >/w7**. You won't see much happening on your terminal screen but when you clear key to window 7 you will see a pretty formatted **mdir**.

Something apparently missing is CLS. It's not really missing, just hidden. Use **display c**.

Many of the 3rd-party utilities contain a built in help file. To see it, type:

utilityname -? or utilityname -h

Printing these help files often stumps folks, since they are output on the error path which is usually the current screen. Use **utilityname -? >>/p** if the normal method doesn't work.

Starting another window isn't that difficult. To form a full screen 80 column window with a shell in it, type:

iniz /w#
display 1b 20 02 0 0 50 18 1 0 0 >/w#
shell i=/w#&

Where # is the window you want to start (notice we've already used 7), and the last 3 numbers in the display command(1   0   0)   are foreground, background, border color. These are hex numbers, so 50 18 is 80x24. There is a utility (wcreate) that uses decimal numbers, but display is faster, since it's always in ram.

Best of luck:
--Rick Uland;Delphi --
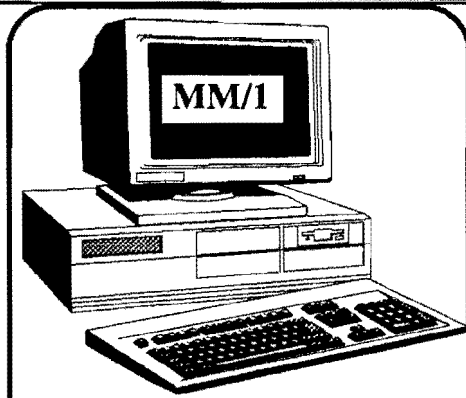
Professional OSK for under $200 for Ataris

An alternate operating system for the AtariST or MegaST for the serious home computer user or hobbiest has just been released at an affordable price. It is so good that some people may choose to buy a ST for the purpose of running this system. Many of us have dreamed of being able to afford a UNIX machine someday and this may be the opportunity to have some of the benefits of UNIX while operating on a much smaller platform.

On my desk is a press release dated September 7th, 1992 from CUMANA, Ltd. Pines Trading Estate Broad Street, Guildford, Surrey, GU3 3BH, England Tel:(0483)503121, Telex:859380, Fax No. 503326.

This press information reveals slashed pricing for the personal user of OS-9/68000 upgrade version. The new package not only includes the professional edition of OS-9 but also contains a word processor with mailmerge and spell-checking, a spreadsheet, a C compiler, a basic compiler, a screen editor and an assembler with debugger and linker. A complete set of manuals weighing approximately 8Kg are also a part of this package. The home user may never use parts of the manuals, but one cannot tell what references he/she may desire in the future so the entire set is shipped.

Allow me to paraphrase a little from my older version manual: The author calling OS-9 a sophisticated operating system says that it is specially designed to make its powerful features easy to use, even by persons with limited technical knowledge (of which I am one).   OS-9 is a multitasking and multiuser operating system which means that many programs can be running at the same time and if desired can be used by more than one operator at the same time.   OS-9 is compatible with UNIX at the C language source code level. However, OS-9 is written in assembly language and being smaller than UNIX, it is faster than UNIX and offers the user more performance. Now from the press release: OS-9 is said to be the only major commercial operating system that has built-in, easy-to-use functions to support modular programming techniques.   All routines are independent of Atari's

OSK Professional...Cont'd

firmware and no Atari ROM routines are used.

From the CUMANA brochure: A single-sided boot disk can be supplied but one double-sided drive will be necessary to run the package. Both the monochrome and the color monitors are supported. All ports including the MIDI ports are supported with this version of OS-9.

The Atari hard drive is supported. Totally Atari compatible drives probably are OS-9 compatible but cannot be individually supported. Caching is incorporated in the floppy disk driver and multi-sector "read and write" is implemented in the hard disk driver as an alternate to caching to ensure data integrity.

And now the GOOD stuff: The commercial price of around $900 US will no longer apply to the individual home computer user. CUMANA offers their private purchaser package at about $179.00 US plus shipping.

The shipping is rather expensive for a five day delivery at $81.00 US; however, if 10 Atari users group together (such as a club) then the individual shipping costs will be reduced. To send 10 packages, the shipment cost is about $219 air freight.

Anyone wishing to become an agent for CUMANA in the USA should contact the Managing Director, Mr. John Simnett. The marketing manager is Mr. Steve Dickenson and Richenda Wood handles public relations.

My only interest in submitting this information is one of an OS-9 hobbiest. I have no financial connections with CUMANA or any OS-9 or Atari vendor. My information should not be considered as binding on anyone. Each interested party should make their own contacts for full verification. I regret having to write this last paragraph, but I have been mis-understood in the past as to my motivations. Hoping to hear from many new Atari OS-9ers in the near future.

PhxKen on Delphi
Ken/Az on Fido, OS-9 ECHO
Farrell Kenimer

CoCo Items FOR SALE

NEW:

Tandy CM-8 RGB Monitor	$100
Color Computer 3	$100
5-1/4in. Double Disk (Model 501 + 502)	$150
5-1/4in Bare Disk Drive (FD-502)	$ 20
Color Computer Dust Cover	$ 5
DMP-105 Printer Dust Cover	$ 5
512K RAM Upgrade	$ 75
Tandy Modem Pak	$ 10-
Disk Controller Chip to run Double Sided Drive	$ 20

USED:

Tandy 1000 with CM-5 Monitor	$200
TRS-80 Monitor	$ 20
Tandy CM-8 RGB Monitor	$ 65
Blue Streak Printer Interface	$ 30
Double Disk (Disto 4 Side) 5-1/4in. Drives	$175
Color Computer 2	$ 30
TRS-80 Micro Color Computer (TC-10)	$ 25
Color Computer 3 (for parts)	$ 10-
DMP-105 Printer	$ 50
Color Computer Joystick	$ 5
Color Computer Mouse	$ 5
Hight Resolution Interface	$ 5-
Tandy Multi-Pak	$ 90
CC-81 Tape Recorder (for CoCo)	$ 25
CC-81 to CoCo Cable	$ 5

NEW SOFTWARE:

Flight Simulator II	$ 10
Carmen Sandiego	$ 20
OS-9 Level II	$ 20
VIP Writer III & Disk-Zip	$ 50

USED SOFTWARE = $10:

Carmen Sandiego   OS-9 Level II

USED SOFTWARE = $5:

Kings Quest III	Rogue	Koronis Rift
Rescue on Fractalus	Interbank Incident	Shuttle Launch Control
Sub Battle	Pantomgraph	Flight Simulator II
CoCo III Ramdisk	Spell & FixDyPrint	CoCo Artist
DeskMate		

ROM PAKS = $5:

Typeing Tutor	HandyMan	BackGammon
Super Pitfall	Springster	Thexder
Color File	Rad Warrior	Silpheed
Color Scriptsit	Personal Finance	Gin Champion
Malcom Mordar	Castle of Tharoggad	Predator
Shanghai		

TO ORDER

Send Personal Check, Money Order, Certified Check or COD to: **Gary Glass, 616 Hudson Ave., Alliance, NE 69301** or call between 7pm - 9pm MST (308) 762-1421. Included $5 Shipping, and please no orders under $20

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

The October meeting was held at Fairhaven Middle School. The first part of the meeting dealt with playing around with the GIMIX Computer and a working demonstration of a multiple choice exam program being used at the school to test student's knowledge of school rules and consequences.

A multiusing environment problem was presented by **Rodger Alexander**: *A user is unable to run a packed basic09 program until the Super User loads the program, and RunB and Inkey into memory. WHY? (The user is in the correct execution and data directories, and all of the files are attributed for public read/write/execution and sharable)* See answer at end of article.

Wes Payne was unable to attend so his presentation on his multiuser log in security system for OS-9 was not available.

The Bellingham School Board was using the school Library where the PC Computers and modems are located so a demonstration of FidoNET featuring the OS9, PNW and MM1-Tech echos could not be presented.

Modification of the Hard Drive Boot file (OS9Boot) was accomplished using EZGen by Burke & Burke. No one had actually done an EZGen re-write of the boot file on a hard drive and we were not sure it could be done without disabling the boot capability. The hard drive had no "DD" driver and of course many applications call up /DD instead of /H0. So we save out a copy of the H0 descriptor to the root directory and used EZGen to rename the file to **DD**. Then we EZGened the OS9Boot file and inserted our **DD** descriptor right after the **H0** descriptor. We quit EZGen and held our breath. No problems. Then we re-boot the computer fully expecting it to fail.....but it came up and sure enough our new descriptor was there. Success!

The main activity of the meeting was the distribution of the OCN (OS9 Community Network) Newsletter that contained an exceptional article on assembly language by **Daniel Hauck**. We entered the assembly code listing **CLS.BIN** and then reviewed each line as it was examined in the article. Daniel Hauck also included a generic **Header** listing that takes care of most housekeeping chores. This simplifies writing assembly code leaving the programmer to only insert the necessary execution codes between the START and EXIT labels. Of course it's not really that simple but it did cause light bulbls to light above our heads and comments like "OH YEAH, NOW I GET IT!?", "KIND OF LIKE BASIC09 SYSTEM CALLS???"

The remainder of the meeting was spent in planning future club activities:

1. Produce some more video tape presentation on Installing a CoCo into a PC Case and Installing a 6309.
2. Discover how OS-9 Level III on the GIMIX lets modules load into memory at there real size instead of 8K blocks, and to come up with a patch to OS-9 Level II to achieve the same advantage.
3. Continued investigation of the GIMIX machines and their multiuser security system.

-- Rodger Alexander --
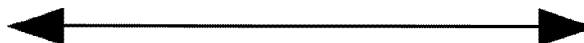
Port O-CoCo Club

The papers for the non-profit incorporation of the club have been sent to the State. The registration papers should be back in our hands in 10-15 days. Everyone wants to thank **Bud Helck** for single handily taking on this task and for working on the by laws, which will be available at the next meeting. The whole process is not costly nor complicated. Any group that believes it will be around for a few years should entertain such a move for enhanced credibility and legal recognition of the club. Port O' CoCo has all the details, most of which have been mentioned in this publication, available to anyone who wants it. Just ask. Birthdays this month, and the free movie tickets that go with it, went to **Donald Zimmerman** and **Russ Griggs**. These $6.50 tickets are one of the few benefits of admitting you are growing old.

The rest of the meeting was divided between two topics: Our tower project and ADOS.

The tower project is going VERY WELL. **Gene Elliott** has devoted many hours to creating the club's own tower system. The drives are now in place, the circuit boards are mounted, and he has come up with a dependable way to use the original keyboard in its original case with a ribbon cable to the tower. This approach saves about $100 until we determine that using an IBM keyboard is a real plus for our use in the club. In the weeks to come Gene is going to prepare all the connections for joysticks, input/output and even ROM PAKS so they can still be used while all the guts for the CoCo are in the tower. Such an approach may be a first. If you are interested in the tower shell that made this possible or any other questions contact Gene at (206)692-7649

The ADOS presentation was enthusiastically given by **Terry Laraway**. ADOS is an enhanced, EPROM able Disk

BASIC for the CoCo (by Arthur Flexser distributed by SpectroSystems, 11111 N. Kendall Dr #A-108, Miami, FL 33176, phone (305)-274-3899). ADOS may be used either as a utility to be loaded from disk or may be burned into an EPROM chip used to replace the Disk BASIC ROM in the disk controller. Special attention was given to ADOS so it would be virtually 100% compatible with commercial CoCo software. Just as abbreviations such as "ASAP" accomplish the same thing as the fully spelled out words "as soon as possible," ADOS allows you to issue commands and write BASIC by using a family of single key strokes instead of written out words. Every key on the keyboard has a second purpose with ADOS when used with the ALT key. In addition further information is available. For example, when DIR is used you get not only what's on the disk in question, but also the remaining number of free granules is displayed as well. Also the date and time of the file's creation (or other helpful information) can be added to each file name.

Some of the other options Terry demonstrated were:
1. Repeat & edit off last commands
2. Automatic line numbering
3. Lower case command recognition
4. Error trapping
5. One or two column directory
6. AE error override w/COPY or RENAME option
7. RUNM for ML programs
8. RAM for ROM to RAM transfer
9. SCAN to list ASCII files to printer or screen 10. Customization capability

Contact Terry (see ad in this publication) for further information.

The next Port O' CoCo meeting is Nov. 16th. The topics are:
1. Update on tower project
2. OS-9 discussion (Beginners welcome)
3. MM-1 show and tell

-- Donald Zimmerman --
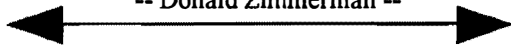
Seattle 68xxxMug

The meeting began with a progress report by **Rodger Alexander** of the club's CoCo installation into a PC Case. Rodger demonstrated how he installed the multipak 90 degrees to the motherboard along the back wall of the case. This was accomplished by cutting the top (even numbered) bus wires on the I/O socket so it could be bent up from the motherboard.

The feature presentation was given by **Scott Honaker**. Scott had written a small Basic09 routine to operate the X-10 **Appliance Controller**. Handouts were provided showing the binary/hex codes required to program certain functions in the controller. Most of these functions could be controlled by the buttons on the controller or it's remote unit, but timed

functions and a larger number of devices can only be controlled via a computer which is connected to the **X-10** through a standard RS-232 port.

It is important to note that the currently available **Appliance Controller** is not the same one that used to be manufactured by Tandy specifically for the Color Computer and plugged into the cassette port. The model Scott demonstrated came with software for a PC type computer. But since the command codes were so simple and the RS-232 PAK for the Color Computer is standard, it was a simple task to create a procedure in Basic09 to write the control codes to the RS-232 Port.

As part of the demonstration, Scott wrote out the hexadecimal codes required to activate a one of the remote control units. Then using the display command, we were able to "poke" the appropriate values to the RS-232 descriptor:

display xx xx xx xx xx xx xx >/t2

A tape recorder plugged into the remote control unit was turned on and off by typing in the appropriate display characters

The remainder of the meeting was the sharing of the Bellingham OS-9 Users Group Public Domain Library. Rodger handed out a complete listing of all of the available files in the Library and offered to make the entire library available to the Seattle 68xxxMUG members at every meeting. Bring your own disk (any CoCo format), or a disk (5-1/4inch) will be provided at a cost of 50 cents each. The software is FREE! So for the next 45 minutes was occupied copying off selected files.

-- 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