OS-9 Newsletter

<<-- IN THIS ISSUE -->>

<<<---SPECIAL BULLETIN --->>>

Signetics 68070 15MHz cpu chips are not available at the present time causing an additonal two months delay in the shipping of MM/1 Kits to their customers. Expected shipping date mid May.

BENEFITS TO SUBSCRIPTION MEMBES:
As a subscribing member of the Bellingham OS9 Users Group, you enjoy the following benefits:
1. 8-10 page monthly Newsletter
2. FREE Public Domain Library (5-Megs)
3. FREE Technical assistance (734-5806)
4. FREE Classified Ads
5. FREE OnLine OS-9 Database (676-5787)
6. FREE "HOW-TO" Video Library

more Shell+
Scriptfiles

by Steve Clark

Shell+ for OS-9 Level II on the Color Computer has introduced some powerful new capabilities for the Level II user. With the numerous extensions made by Ron Lammardo, along with Kevin Darling and Kent Myers, it is indeed possible for the user to "go wild" with shell scripts. I have converted some of my favorite shell programs to take advantage of the features available with shell+. I offer these as suggestions only, you will need to modify them to your liking. I did discover a few facts about the shell+ which I will mention as I go along.

This article contains a few shell+ script files intended to demonstrate some of the things you can do with shell+. In some cases, I am sure there are alternate ways to do what these scripts do. Most of these use other programs available on-line on various information services. I will try to mention these when they are encountered.

RAMDISK - Setup a Ramdisk

This shell+ script sets up the ramdisk, giving you a choice of what size. The rammer and /r0 ramdisk provided by Kevin Darling is used, and you must have these in your bootlist. You could modify the script for other ramdisk setups. It also uses Kevin's dmode utility, to set the size of the ramdisk (cyl=##). It uses the prompt and var.0 input, along with if/else/endif from the shell+. If you press ENTER, no ramdisk is created. Remember that once you create the ramdisk, you must reboot to get rid of it or change its size.

```
*ramdisk
prompt Ramdisk  1=40 track, 2=35 track, 3=20 track:
var.0
if %0=1
  echo 40 Track
  dmode /r0 cyl=28
else
if %0=2
  echo 35 Track
else
if %0=3
  echo 20 Track
  dmode /r0 cyl=14
else
  echo ....No Ramdisk Created....
  goto +end
endif
clrif
```

```
iniz /r0
format /r0 r "Ramdisk">>>/nil
echo /r0 created
free /r0
*end
```

PCTL - printer control

This sends display codes to the DMP-120 printer, setting various modes like condensed print, wide print, etc. It puts up a simple one-line menu for you to choose from, then sends the appropriate control codes to >/p. Change it for your printer and you won't have to look up the codes again. This one traps the error and prints a message when an error occurs (usually the printer has not been turned on).

```
*pctl
onerr goto errtrap
prompt Choose (n)ormal, (c)ondensed, (w)ide, (e)xtra wide:
var.0
if %0=n
    display 1b 0f 1b 13>/p
else
if %0=c
    display 1b 0f 1b 14>/p
else
if %0=w
    display 1b 14 1b 0e>/p
else
if %0=e
    display 1b 13 1b 0e>/p
else
    echo Pctl terminated.
endif
clrif
goto +finis
*errtrap
echo An error %* occurred, check printer status.
*finis
```

WCTL - window control

Similar to ramdisk, this does various types of window setup. Some of the options work on the current window, some create new windows. I use other utilities like w80, gw80, gw40, etc. for quick window changes, but keep wctl for seldom-used window commands that I don't want to keep track of (like how to do an over & under 80 column setup) on paper. Again, modify it for your uses. This script simulates a multi-way if/then by actually nesting the if/then's but only using one endif and doing a final clrif to clear them out. This uses Fred Sawtelle's wmode utility. Also, I keep echo, display, and prompt in memory at all times; this speeds up execution.

```
*wctl
echo Pick one of the following:
echo 1 Graphics 640x192 (black on white 4 color)
echo 2 Graphics 320x192 (16 color)
echo 3 Text Window
```

```
echo 4 Over and Under W4 W5
echo 5 Graphics 640x192 (white on black 4 color)
prompt Choose:
var.0
if %0=1
    display 1b 24 1b 20 7 0 0 50 18 2 0 0 1b 3a c8 01
    display 1b 21 </1 >/1
else
if %0=2
    display 1b 24 1b 20 8 0 0 28 18 0 1 0 1b 3a c8 02
    display 1b 21 </1 >/1
else
if %0=3
    display 1b 24
    display 1b 20 2 0 0 50 18 0 2 3 1b 31 2 0 >/1
else
if %0=4
    wmode /w4 col=50
fgc=0 bgc=2 bdc=3
    wmode /w5 col=50 row=a wnd=5 val=1 sty=ff cpx=0 cpy=e
fgc=0 bgc=2 bdc=3
    iniz /w4
    shell i=/w4&
    echo Go to w4 and start a shell for w5
else
if %0=5
    display 1b 24 1b 20 7 0 0 50 18 0 2 2 1b 3a c8 01
    display 1b 21 </1 >/1
endif
clrif
```

ENV – environment setup

This is one of my favorites. The reason I wrote env was to facilitate rapid switching of environments. By "environment" I mean a particular combination of execution directory (cx), working directory (cd), etc. I used Ron's datamod to make this into a "mem" script which I keep in memory at all times. It is 960 bytes in the datamod version, a small price to pay for the convenience.

With my hard disk, I use a lot of subdirectories, for instance the subdirectory /DD/USR/PROG is where I put all my program development stuff (compilers, source code, test files, etc.). Under this I have a directory branch for each type of programming: C, ASM, BASIC09, SHELL programming, and so on. I keep the C compiler, for example, in /DD/USR/PROG/C/CMDS, and keep C source in /DD/USR/PROG/C. Similarly, I keep the assembler in /DD/USR/PROG/ASM/CMDS. This allows me to compile or assemble to the unique execution directory without messing up my working version of a program until the new version is finished. The problem this creates is that a lot of long path names have to be typed. If I am in C and want to refer to a Basic09 program, I have to type something like

    "list /dd/usr/prog/basic/file.b09"

to list the file. If I'm not sure of the file name, I first have to do a dir on that directory. NO MORE! With the shell+ I can instantly switch to it by typing "env" and answering "b". I can get back to where I was by doing an "ex".

This is consistent with the windowing environment, using env

something like the CLEAR key. It makes me more productive by saving a lot of path name typing. Unless your directories are exactly like mine, you can't use this directly, but can modify it to your liking. I experimented with this a lot before finally settling on the method of creating a new shell for the new environment. That avoids the DEAD shell syndrome, and only ties up one extra block of memory, plus I can get immediately back to where I came from. I also set a path=/DD/CMDS or some alternate path so my main CMDS is available when I need it. [NOTE: You can execute a program along a path by typing "edit xxxx" (for edit), but the load command doesn't know the path, so "load edit" doesn't work, but "load /dd/cmds/edit" does work.]

Env uses a computed "goto" which is not an explicit part of shell+ but can be easily created by using the var.0 input as part of a goto. i.e., "goto +lab%0" puts the contents of var.0 at the end of "lab" before starting up. If you have a "+labc" label in the script, that is where it will go if you reply with "c". You need a "+\" at the point where you want to catch non-matching input, and a "goto +somewhere" at the end of each branch to take you out.

I set a special prompt for each environment so I will know when I am in an "environment" vs. being in an original window shell.

```
*env
onerr goto +lab
prompt Choose - Asm Basic C Database Letter Other Shell Terminal
Vef Quit:
var.0
goto +lab%0
*laba
  cd /dd/usr/prog/asm
  cx /dd/usr/prog/asm/cmds
  var.2=/dd/cmds
  var.1=ASM>
  goto +cont
*labb
  cd /dd/usr/prog/basic
  cx /dd/usr/prog/basic/cmds
  var.1=B09>
  var.2=/dd/cmds
  goto +cont
*labc
  cd /dd/usr/prog/c
  cx /dd/usr/prog/c/cmds
  var.1=C>
  var.2=/dd/cmds
  goto +cont
*labd
  cd /dd/usr/data/db
  cx /dd/usr/data/db/cmds
  var.1=DB>
  var.2=/dd/cmds
  goto +cont
*labl
  cd /dd/usr/spc
  cx /dd/cmds
  var.1=LETTER>
  goto +cont
```

l

```
'labo
  cd /dd/usr/prog/other
  cx /dd/cmds
  var.1=OTHER>
  goto +cont
'labs
  cd /dd/usr/prog/script
  cx /dd/cmds
  var.1=SHELL>
  goto +cont
'labt
  cd /dd/usr/comm/download
  cx /dd/cmds
  var.1=TERM>
  goto +cont
'labv
  cd /dd/usr/pix/vef
  cx /dd/cmds
  var.1=VEF>
'cont
ex shell p=%1  i=/1 path=%2;echo -- Type EX to return.
'lab
'labq
^
  echo -- No change.
```

SETWILD - turn wild on or off

The final shell+ script in this batch turns the system-wide wild card option on or off. At first I thought I would leave it on all the time, but later decided I would leave it off and keep the utilities I had that do their own wild card processing, like cp, lsH, and rm. However, if I want to experiment I can easily switch back and forth with setwild. I put Ron's modpatch files in /DD/SYS but you can put them anywhere, just change the shell+ script. Here is setwild.

```
prompt on or off:
var.0
if %0=on
  modpatch</dd/sys/wild.on.scr -s
else
  if %0=off
    modpatch</dd/sys/wild.off.scr -s
  else
    echo Answer with 'on' •r 'off'
  endif
endif
```

BATCHFMT - Batch Formatting Floppies

If you need to format a series of floppies, say to use for backing up your hard disk, this shell script automates some of the process. You supply an initial name, and the script formats a floppy with "name00001" then asks if you want to do it again. By replying yes, the script prompts you to insert a new disk, etc. Drive /d0 is hard wired into the script, but you can change it (or add an option to ask which drive). It uses the GOTO and INC options from the new shell+.

```
'batchfmt
display c
load format
prompt Disk Name:
var.0
'loop
inc.1
display c
echo Place disk in drive /d0
prompt Press ENTER when ready to format %0%1:
var.2
format /d0 r "%0%1"
display 7 a
prompt Another (y/n):
if [ -y ]
  clrif
  goto loop
fi
unload format
echo Batchfmt Done.
```

DESKMATE - Run Deskmate 3 Applications

This shell script runs deskmate 3 applications by presenting a menu and starting either the deskmate interface, or one of several deskmate applications. As with other scripts, you will have to use your own subdirectory names (I use /dd/usr/data/ss). It will set up a type 1 window and run the specified deskmate program in that window. See the script for further information.

```
'deskmate - deskmate execute
onerr goto +trap
display c
chd /dd/USR/DATA/SS
chx /dd/USR/DATA/SS/CMDS
path=/dd/CMDS
echo DeskMate
display a
echo 1 DeskMate Desktop
echo 2 Spreadsheet
echo 3 Word Processing
echo 4 Communications
display a
prompt Select:
var.0
goto +label%0
'label1
xmode /w6 type=1;display c>/w6
(desk<>>>/w6;xmode /w6 type=80)&
goto +finis
'label2
xmode /w7 type=1;display c>/w7
(desk dmledger<>>>/w7;xmode /w7 type=80)&
goto +finis
'label3
xmode /w8 type=1;display c>/w8
(desk dmtext<>>>/w8; xmode /w8 type=80)&
```

```
goto +finis
*label4
xmode /w9 type=1:display c>/w9
(desk dinterm termstat<`>>/w9;xmode /w9 type=80)&
*finis
display a
echo Task started in another window.
echo Use CLEAR key to change windows.
*trap
```

PLAYIT – Play Sounds from a Menu

I found Kevin Darling's play program fascinating to use, and have collected several digitized sound files. The one thing I never can remenber is what parameters to pass to each file. One way to handle this (or to have the computer remember for you) is to maintain a script file such as "playit" and put the parameters in. Again, specify your own directory structure.

```
* playit - play sounds
onerr goto +lab
load play
chd /dd/usr/sound
*repeat
echo 1 HAL from 2001      2 Late
echo 3 I'll be back       4 Captain Kirk
echo 5 Disruptor          6 Scotty
echo 7 Clint Eastwood     8 Laugh
echo 9 Monty Python
echo
prompt Which Sound (ENTER to Quit):
var.0
goto +lab%0
*lab1
  echo HAL - Can't Do
  play 18 cantdo.snd</1
  goto repeat
*lab2
  play 11 davidl.pla</1
  goto repeat
*lab3
  play 28 back.mac</1
  goto repeat
*lab4
  play -29 kirk.pla</1
  goto repeat
*lab5
  play -28 dsrpt.pla</1
  goto repeat
*lab6
  play -28 scotty.pla</1
  goto repeat
*lab7
  play -11 clint.pla</1
  goto repeat
*lab8
  play 40 laugh.pla</1
  goto repeat
```

```
*lab9
  play -10 bing.pla</1
  goto repeat
*lab
unload play
```

ADDRESS – Build an Address File

This won't replace a database by any means, but it is a quick and dirty way to create a program to obtain prompted input from a user. It adds to an address file called "address.dat". You can use the same idea to create any type of ascii data file. It uses the IF, append, prompt, and GOTO features of the new shell+.

```
*address
var.0="address.dat"
*repeat
prompt Last Name:
var.1
prompt First Name:
var.2
prompt Address:
var.3
prompt City, State:
var.4
prompt Zip:
var.5
prompt Phone (999-999-9999):
var.6
prompt ----- Add to file (y/n):
var.9
if %9=y
  echo %1, %2>+%0
  echo %3>+%0
  echo %4 %5>+%0
  echo %6>+%0
  echo ->+%0
else
  echo Record not added.
fi
prompt Add Another (y/n):
var.9
if %9=y
  clrif
  goto repeat
else
  echo Done. %0 Data input complete.
fi
```

MAN – Online Manuals

One of the nice features of having a hard disk is the ability to keep some things on-line which you would otherwise have to store on floppy or in the case of documentation, keep printed copies. I created this shell script to simulate the MAN command (somewhat) by providing access to documentation files. I have a subdirectory called /dd/usr/man which stores the text files I want to access,

and use this script to choose the one I want to view. Pete Lyall's MORE command allows you to page back and forth within a file, and is perfect for implementing this application. When you execute man, something like the following appears:

```
Directory of /dd/USR/MAN  00:11:48
CLib        Datamod      MacPaint     Mail
MaxiPic     Mkdir        NBS          Pilot
Shell       TelStar      Tiny         UltiMuse
Xcom9       Xlisp

Which manual entry:
```

You then type the name at the prompt (or press ENTER to ignore) and the documentation is available. Note that the exact names will depend on what you have in the /dd/usr/man directory. This example shows what I have in mine at the present time. The man script looks like the following:

```
*man - manual processor
onerr goto repeat
cd /dd/usr/man
echo
dir /dd/USR/MAN
echo
*repeat
prompt Which manual entry:
var.0
if -r %0
  echo
  echo Manual for %0:  Use SPACE/B to page through, Q to quit
  /dd/cmds/more %0
else
  if %0 >= a
    clrif
    echo -- man: No manual entry for %0
    goto repeat
  fi
fi
```

TYME - Digital Clock

We all seem fascinated with turning our expensive computers into cheap clocks, myself included. I have used the new shell, along with DATE, DISPLAY, and SLEEP to create an on-screen digital clock. It doesn't do anything but sit there and display the date and time in the middle of an 80 column window. Use it when you are going to leave your computer idle for a few minutes. To stop it, use control-E.

```
*tyme
onerr goto +trap
tmode -pause
display c 05 20
*repeat
display 2 3a 2b
date t
```

```
sleep 30
goto repeat
*trap
display c 05 21
date t
echo Tyme end.
```

Use these as supplied, or use them to generate ideas for your own shell+ scripts.

Steve Clark - Compuserve 73135,1204
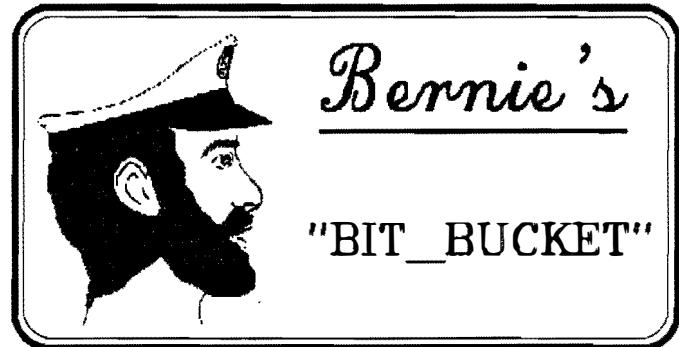
by Bernie Besherse

Motorola Inc. of Illinois has unveiled a global communications system that will allow people to communicate by telephone anywhere on earth, - on land, sea, or in the air - via portable cellular radio-telephones operating as part of a satellite-based system.

Callers will not need to know the location of the person being called; they will simply dial that person's number to be connected instantly, says the company. Motorola calls the new system IRIDIUM and has established a satellite communications business unit to develop it.

The heart of IRIDIUM is a "constellation" of 77 satellites in low orbit, working together as a digital switched communications network in space. The system will be able to handle both voice and data.

MOTOROLA says its IRIDIUM system provides several key improvements over the geo-synchronous satellites currently used for international communications. The low altitude of IRIDIUM satellites allows easy radio links with celular radio-telephones on earth, using small antennae rather than satellite dishes. It also supports re-use of radio frequencies, in a similar fashion to land-basedcellular systems.

In addition, the system solves the problem of low-orbit satellites "disappearing over the horizon" by combining a large number of satellites in a space-based, inter-satellite switching system.

Although IRIDIUM uses cellular communications principles, it is designed to complement, not compete with, land-based cellular systems. Land-based cellular will remain the most efficient way to serve high-density areas. whereasIRIDIUM will bring communications to remote or sparsely populated areas that lack communications.

Subscriber units for IRIDIUM are similar to MOTOROLA's original cellular radio-telephones and will offer additional features such as latitude, longitude, altitude, and Greenwich Mean Time. In addition to the lightweight portables, IRIDIUM subscriber units will be available as mobiles or small, fixed units.

MOTOROLA envisages the IRIDIUM system being operated by one or more international consortia whose members have the necessary liscenses to operate in each country. MOTOROLA will serve as the supplier of the system itself. This will include the satellites, the communications links, and all necessary support.

Implementation of the entire system is planned to begin in 1994, and full service will begin as early as 1996.

This article was taken from an amature radio magazine. It is a news release from MOTOROLA and not believed to be copyrighted.

OS9 Tips

#1 A big trend lately is to put the CoCo into a PC type case. Neat, compact and looks spiffy! But what to do about some of the option switches on the front of the PC case. Like the "Turbo" switch. Craig DuBois suggest that you mount your Hi Rez Joystick Interface inside the PC case and use the turbo switch to switch your joystick port from "low" to "hi rez". Works great and eliminates the need of having your joystick interface hanging out the back end of your "spiffy" new case.

#2 What to do about the keyboard keylock on your PC case? At first I routed the 5 volts B+ that went to the multipak so that with the lock on there would be no access to the floppy and hard drives. It worked, but I was concerned about the B+ coming on and off line with the system running and possible volatage spikes. Brian White suggested that a much better solution would be to supply the +5 volts to pin 4 of the unused joystick port through the lock switch. That would truely disable the keyboard as the lock switch was intended to do.
WARNING!!!!!! The +5vdc must be brought

to pin 4 of the joystick through a 10K resistor so as not to create an excessive load to the B+ (and smoke your power supply).

+5vdc ——— key-lock switch ———— 10k resistor ———— to joystick (pin4)

#3 Want to speed up your OS9 boot? Simple, change the interleave on the tracks that contain the boot file. According to Brian White, here's how:
1. Format a blank disk normally
2. Format the disk again as follows:
   format /d0 "diskname" r '7' :1:
3. Now do a logical format of the disk:
   format /d0 "diskname" r L
4. OS9Gen or Cobbler the bootfile to the disk
Normally, OS9 can not read a disk fast enough to take advantage of faster interleaves, but it in this case it works.

Hard Drive Concepts

Hi I'm Kerry Kowalski, I run a BBS out in Wisconsin (another "W" state) The DATA STASH, Phone # 414-684-4115, 300,1200,& 2400 baud. Just like most RiBBS. Most know me as "Rocky" here...

When I received my first 40 meg set up from Howard Medical I immediately incorperated it into the BBS setup I had (1meg COCO 3, 3 3.5" 720K drives, MPI, & modem ect) as /H0. I'd noticed problems right away. The system wouldn't reboot using the B&B XT-ROM... Also I noticed the system would slow down when it had to pull stuff from /DD & /D0 which many OS9 programs default to. Well to me this was a BIG problem. When people call my BBS, they expect it to be there when they call, if it's not they DON'T call back! Well the system wouldn't reboot because the disk in /d0 had crashed due to the voltage spike that occurs when power is turned on or off with the disk in the drive. Time to make another BOOT disk! What a pain!

It didn't take me long to figure a way around BOTH problems... Soon after I got my 80 meg so I set up the boot for the 80 meg as /DD & /D0 on the BBS. So the system is set up like this: I renamed the

80 meg ST-4096 "HD" descriptor (using EZGEN) to "DD" & "D0", then I renamed the floppy descriptors: "D2" to "D3", "D1" to "D2", "D0" to "D1". Finally, I incorporated all of the renamed descriptors into the os9boot file using EZGEN.

Now the reason I stayed with using /DD, /D0, /D1, /D2, /D3 is this....Have you ever tryed to DMODE a drive by any other name??? It doesn't work! Try it for yourself! I did find though that the new DMODE that comes with the Disto's Super Controller-II works with drives by other names but the original DMODE comes back in error...

Now there are a couple of ways to BOOT your system with the HD as /DD & /D0...
#1 Try to do like it says in the XT-ROM by clearing track 128 & 129 for your BOOT & ALTBOOT (which I found to be a pain!)
#2 Copy your HD to disk using DIRCOPY, formating it and installing your boot (read the XT-ROM book) then copy your files back. To me this was easier because I crashed my HD using #1, and Chris Burke seems to be hard to get a hold of when you need him!
#3 Copy yer startup file and yer CMDS directory from your disk to the HD. Then make an OS9Boot disk as described above, with your HD as /DD & /D0. All you need then on this BOOT DISK is the OS9Boot all by itself! No startup, CMDS, SYS, or any thing else is needed on it, as long as you have it all on your HD everything should go! Put the flopy boot in your physical disk drive 0 and type DOS and watch the magic!

OOPS I almost forgot... (HECK no one's perfect!) Add to your startup file Burke's D0OFF command and make shure it's in your CMDS directory too! In a 128K of 512K system, put the D0OFF command at the beginning of your startup file.

Any questions call the DATA STASH BBS #414-684-4115 8N1 300,1200,2400 24 HRS Wisconsins ONLY RiBBS BBS! Svsop Rocky

Club Activities

SEATTLE 68XXX MUG will be having a special April Fools meeting that should be lots of fun. Scott Honaker will bring several assorted computers, from an ancient Model-II to a high speed PS-2. All of them will be operating OS9 Level-II! At last, OS9 for MS-Dos machines? **APRIL FOOLS!** All of the machines will be connected to one little CoCo-3. The other computers will simply be operating as OS9 terminals. After all of the "fooling" is done, we will be given a presentation on how to hook up other computers to the coco and the necessary software, etc., including the ●S9 commands TSMON and LOGIN.

A bonus feature of the meeting will hopefully be the demonstration ●f a new MM/1 OSK computer, thanks to Brian Wright. Brian is a temporary employee of Microsoft, but actually hails from the University of Winnepeg, Ontario, Canada. Brian has been selected as a program developer by Keneth Leigh Enterprise for the MM/1. Brian programs in 12 different languages and is exciting to listen to as he describes all of the new features of OSK compared to OS9.

There will also be a CoCo-3 installed in a "mini-lower" case for viewing (If I get it done in time.....Ooops, just blew my second 20Meg hard drive in as many days, Damn!)

BELLINGHAM OS9 USERS GROUP will not be having a meeting in April. Individual projects going on, but not completed for presentation are installing CoCo-3's into mini tower cases. Craig DuBois is completely re-designing the older Multipak, etching his own board so that it will more neatly fit into the PC case. He promises a complete write up with diagrams for the newsletter when he is finish.
-=Rodger Alexander=-

LONGVIEW/KELSO COCO CLUB will have it's next meeting on March 30, Saturday at 2:00 p.m. We will be installing the 256k upgrade and the MPI irq hack that was in the January OS9 Newsletter if my upgrades arrive before the meeting. We will also be transferring some rom-paks to disk for some users and general help on the coco.
-=Mark Johnson=-

MT.RAINEER COCO CLUB March 12th meeting of the Mt Rainer Coco club featured the following:

Ken Ecker discussed Max 10 and demonstrated its use as a word processor. He also showed the various font's and editing features.

Alan Johnson talked about Basic09, debug fuctions, print, state, tron, and step. A basic demo of programing in "C" was also led by Alan.

A general discussion of problems individuals were having with their coco 2 & 3 in communications. RSDos and OS9.

An agenda for the April 9th meeting is not available.

-=John Schliep=-

PORT O'COCO is growing and thriving in Port Orchard. March's program started off with the sale of an extensive system consisting of a CoCo II and CoCo 3, printer, drives, and lots of software. The interest way high. Everything moved into a new home. For those tracking prices the CoCo 3 which had had only light use went for only $60.

This auction atmosphere was followed by a discussion of our participation at the Kent Swap meet. We agreed that we have had growing interest with each appearance at the event over the last year. The last swap had an estimated 5,000 people wandering around. Now we need to raise our profile by having a colorful banner and some king of a handout about the various CoCo clubs in Puget Sound. Prices for a banner have been gathered since and a banner can be made for only $5 per square foot. But the challenge is what to call ourselves. The curtain is coming down on the Tandy Color Computer. And yet the family of machines contiue to lives. Thus what do we call ourselves? The next Swap Meet is June 1st.

We continue to let others know that we exist by a continual notice on two cable systems, another year of notice in Puget Sound ComputerUser, notice in the county greater's service, and notice in this newsletter. All are spreading the word.

In reaching beyond our own community and our own machine, the next topic was about a complete IBM machine that was given to a Port Orchard woman who is unable to hold down a typical job because of continueal and severe back pain. Now she is able to work out of her home when she feels up to it. The machine was given to her by the Computer Bank Charity. Port O' CoCo is oneof the support clubs and delivered the machine. We have also given some assistance in getting her system up and going.

We also discussed the idea of a CoCo Fest/Computer Swap Met in Port Orchard during the summer. We had a 5 ring circus coming to town June 22nd (Friday). We might want to shirt-tail on that event with such an event the next day. We will discuss further a the next meeting.

Finally Mark King wound up his 7 part series on BASIC with part two of his tutorial on string variables. Mark has agreed to begin a new tutorial next month on C. All interested people should call Mark at 697-5576.

Next meetings April 15th, May 20th, and June 17th. The April meeting will kickk off our series on "C" and there is talk of a demo of transferring files from a CoCo to a Mac. All are welcome! Bring a cup (for the club's beverage, a $1. and a bag of your favorite snack to share with your love of the CoCo.)
-=Donald Zimmerman=-
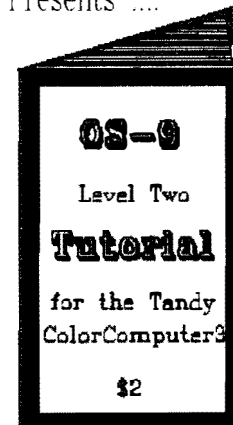
DataMod
for Shell+

by Rodger Alexander

DataMod is a companion utility for Shell+ written by Ron Lammardo. The purpose of this utility is to convert shellscripts to loadable data modules or data modules to shellscripts. This is accomplished by DataMod appending (or stripping) a module header and CRC to the scripfile so that OS9 will "think" that your scriptfile is an executable memory module, and will load your scriptfile into memory for instantaneous execution.

Keep in mind that when OS9 looks for a file, it looks to see if the file is in memory first, if the file is not there then OS9 searches the execution (CMDS) directory for the file, if the file is not there then OS9 will finally search the default data directory to find the file.

Since script files, like startup, are in the data directory, a great deal of delay in the execution of your startup file is the result of OS9 going through the above mentioned search procedure. OS9 is not smart enough to realize you are requesting the execution of a scriptfile, and that scriptfiles can only be in the data directory. (Shell+ will execute scriptfiles in the CMDS directory.)

The advantages that DataMod provides is only realized when properly utilized. Keep in mind that Level Two always loads memory modules in 8K blocks no matter how small the actual module might be. Your startup file may only be 75 bytes long, yet if you convert it into a memory module using DataMod and then load it into memory to gain instantaneous execution speed you just used up 8K of valuable memory.

The idea is to merge scriptfiles with associated command files or applications. One such example would be to merge the RAMDISK scriptfile included in Steve Clarks article in this Newsletter with Kevin Darling's Rammer utility. By merging the two files under the name RamDisk both files will load into memory together with the scriptfile in control of the Rammer utility as if it were a new "enhanced" Rammer.

ENTER: datamod ramdisk </dd/ramdisk
>/dd/cmds/ramdisk
(This converts scriptfile to a memory module
ENTER: attr ramdisk pe e
ENTER: merge ramdisk rammer >ram
ENTER: delete ramdisk
ENTER: attr ram pe e
ENTER: rename ram RamDisk

Another example would be to merge Steve Clark's WCTL file with Fred Sawtelle's wmode utility. The merged file must be the same name as the scriptfile so that when the merged files are loaded, the scripfile will be the file executed and it in turn will control the execution of the utility.

ENTER: datamod wctl </dd/wctl >/dd/
cmds/windows
ENTER: attr windows pe e
ENTER: merge windows wmode >window
ENTER: attr window pe e
ENTER: del windows
ENTER: rename window windows

The most obvious example of merging scriptfiles with a companion application is Steve Clark's PLAYIT merged with Kevin Darling's PLAY program. But you may choose to keep PLAYIT as a strict scriptfile in your data directory so that it can be easily modified as you add more "sound" files. Take a look at the rest of Steve Clark's scriptfiles and determine which scriptfiles would benefit by being converted and merged with companion programs or utilities and which scriptfiles would not.

As always, Review your shell+ documentation for more information on using datamod with shell+.

Editor's Note

I have written to The Rainbow Magazine notifying them of all of the CoCo/OS9 Clubs in Washington State with the name of each Club's President and their phone numbers. I also included the names and phone numbers of all of the CoCo/OS9 Bulletin Boards in Washington State.

Keeping in mind that The Rainbow operates on a two or three month lead time, hopefully we should see a posting in the June issue. Currently, they show Washington State void of any CoCo/OS9 activity. Hope this will straighten things out.

Review . . . .
New Public Domain Files

This has been a hecktic month with both hard drives dropping dead on me <Grrrrrrr>, so I have only one Public Domain Software review for this months issue. But what a sweet program: Speedisk (version 0.21a) by Brian White.

Brian is visiting the Pacific Northwest for three months while serving temporary internship with Microsoft in Bellevue. Brian is a Canadian student from the University of Winnepeg.

Speedisk is a "disk optimizer" utility. The function is to be able to do a "DCHECK" type of analysis and re-write the data on the disk, unfragmenting, re-arranging, sorting and compressing file data to optimize the disk storage space.

The obvious advantage of this utility is the unfragmentation of files on a disk, improving file access speed and preventing data read errors due to fragmentation. An added feature of speedisk is its ability to "intelligently" place all directories at the beginning of the disk, all files within each directory stored together on the disk, and most amazing, place the OS9boot file at the end of the disk and updating track 34 with the new location of the boot file.

Burke & Burke also has a reformat and optimizer utility, but complaints have been loud and strong about how s-l-o-w the B&B utility is.....Like 30 hours to reformat a 30 meg hard drive. More than an hour just to reformat a single floppy disk. Speedisk is much, much faster. I reformatted/optimized a 360K double sided floppy in just a little over 5 minutes.

THERE IS A CATCH: Speedisk is written in 'C' for both OSK and OS9. Unfortunately, 'C' does not provide code to permit OS9 to access more than 64K, which means that Speedisk can only reformat disk of 1 Meg or Less. Great for floppies, but not suitable for hard drives where the need is greatest. Of course the OSK version of Speedisk can handle hard drives of any size.

Brian is painfully aware of the OS9 shortcoming and is looking into several possible solutions. Unfortunately Brian's main programming responsibilities will be towards OSK and the MM/1 computer, so be very patient.
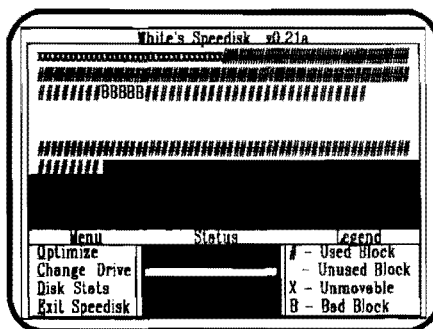
Speedisk is only available from Brian White or Bellingham OS9 Users Group Pub.Domain Library. Brian intends to release seedisk to Delphi and Compuserv in May. Brian also wants to remind all users that this is a beta version and there are possible bugs!

-=Rodger Alexander=-

Washington State BBS Listing

The following BBS list will be of interest to CoCo and OS-9 users:

FAR POINT BBS (Seattle)
(206) 285-8335 RiBBS (Fido NET)

COLUMBIA HTS. BBS
(Longview/Kelso)
(206) 425-5804 RiBBS (Fido NET)

DATA WAREHOUSE BBS
(Spokane)
(509) 325-6787 Level-II BBS

TIME MACHINE BBS (Tri-City)
(509) 586-2559 CoBBS

BARBEQUED RIBBS (Bellingham)
(206) 734-5806 PC-Board
(CoCo/OS9 SIG on Conference 5)

Notice. . . #1
Submit your article for publication. Share your ideas, tricks or any information of OS9 concern. Receive one month's issue FREE!

OS9 Rules

Notice. . . #2
Back issues of the OS9 Newsletter (Jan '90 - Feb '91) available for 55 cents each. Please specify by month and year. Issue numbers may be incorrect.