

ATTR Syntax: Attr filename [permissions] Usage : Examine or change the security permissions of a file Opts: -perm = turn off specified permission perm= turn on specified permission -a = inhibit

AUSTRALIAN OS9 NEWSLETTER

Basic09 Syntax: Basic09 filename CHD S... directory to specified path Cmp Syntax: Cmp filename1 filename2 Usage : File comparison utility COBBLER Syntax: Cobbler devname Usage : Creates OS-9 bootstrap file from current boot CONFIG Syntax: Config Usage: Create system boots and system disks COPY

EDITOR :
Gordon Bentzen
8 Odin Street
SUNNYBANK Qld 4109

(07) 345-5141

DATA Syntax: Data [t] specify : Check for work = save cluster print }<devn filename delete directo [e] [x] names executi Display s converted characters to standard output DSAVE Syntax : Dsave [-opts] [dev] [pathname] Usage : Generates procedure file to copy all files in a directory system Opts : -b make a system disk by using OS9boot if present -b=<path> = make system disk using path as source i = indent for directory levels l = do not process b command ECHO Syn output ED text edito

APRIL 1990

error messages for given error numbers EX Syntax: ex <modname> Usage: Chain to the given module FORMAT Syntax : Format <devname> Usage : Initializes an OS-9 diskette Opts ; R - Ready L - Logical format only "disk name" 1/2 number of sides 'No of

AUSTRALIAN OS9 NEWSLETTER
Newsletter of the National OS9 User Group
Volume 4 Number 3

EDITOR : Gordon Bentzen

HELPERS : Bob Devries and Don Berrie

SUPPORT : Brisbane OS9 Level 2 User Group.

The CoCo4 - We have yet another update regarding the rumoured CoCo4 thanks to Peter Edwards, Victoria, so if you found the information in last month's newsletter interesting, then read on as we have more of the same.

The following text is comprised of messages and postings directly from COMPUSERVE and we reproduce what we think are the interesting parts.

From : Kevin Darling (Wed Feb 14th)

While I'm heavily involved in both KMAs, I can now say who the two companies are so that they get credit. One is being done for Kenneth-Leigh Enterprises (aka Paul Ward) and the other for Frank Hogg Labs (aka Frank Hogg). Details when possible....hang on. Also when possible I'll mention the names of the main designers of each (they're too busy to come up for air right now).

There is already interest in OSK. Unfortunately I don't think I can say yet what PC products are being ported, but there are some nice ones. The big question becomes: will we be able to afford them? The answer to that will depend on how many of these OSK machines are sold... the more market, the cheaper the software can be. best - Kev.

From : L Harris

> I think that the 68k side of the KMA will be the way to go; remember when the CoCo 3 was coming out how > concerned people were about absolute compatibility with the CoCo 1&2, and now nobody seems to use much CoCo 1&2 > stuff anyway. Will there be 6809 emulators still in development when it is released, or how much compatibility > should we expect? [L. Harris]

From : Kevin Darling (Fri Feb 16th)

A lot of people are remembering that CoCo lesson, and we think people will switch fully to the 68K side pretty quickly. Fortunately a lot of CoCo/OS9 software was done in C (no thanks to me :-)) and will port right away. Which is the reason I finally am taking up C, as we'd like to see software at least portable back to the 6809 users, of which there'll still be a good number for a long while.

Both hardware/software emulation should be available. I find the software side to be useful with binary-only tools not yet written under 68K. VDG games are the main reason for the hardware side (oops, I mean games *and* MIDI stuff (grin to Mike K)).

From : RJ Miller

> What are the specs on the new machine gonna be? (speed, ram, disk type and size, etc). How long has it been in > development? At what stage of development is it now? Is software (C compiler, more specifically) being > developed > simultaneously, will it all be ported, will it be left to the users...? [RJ Miller]

From : Kevin Darling

You mean "two new machines".

Well, much of the hardware has been worked on for well over a year... part of an original OS9 UG project which fell through due to members wanting MOTDs more often (sigh) (Editors Note: The MOTD "Message Of The Day" is the newsletter produced by the OS9 U.S. user group.) However, these latest enterprises only started in earnest about four months ago. There are only about a half dozen people involved altogether for both. Consider the number of people and \$\$\$ someone like Tandy has, and how long they take to bring out a new machine. And here we are, attempting to get ours out before the next fest in April! Try gathering a million dollars plus designing machines in that time frame, and I'd say we were all doing pretty well at it.

AUSTRALIAN OS9 NEWSLETTER

Both machines are probably a month or two behind where they'd really like to be. As soon as protos are available, pre-selected developers can begin porting their 6809 stuff over. It may take a month or two to get things really rolling.. not bad. C compiler will be the Microware one, of course, tho it will be sold as cheaply as possible. So start saving now!

Both base systems will be using 8-10Mhz 68000 equivalents. Both will have optional 16Mhz 68030 cards very soon. Both will have a PC keyboard port, plus serial ports. Both should have OSK at least partly in ROM. I know this is all vague, but there is a small bit of competition between the two companies, and I'm there sitting in the middle... more as soon as I'm released. The main biggie to me is that they use different busses. However, I suspect that adapters will show up right quick. - Kev.

(Editor's Note - Some extracts from the OS9 SIG file)

Figure standard 512K, expandable to at least 8meg. Several of each normally expected port types (serial/par/joy) plus one machine has stereo I/O analog ports. Video ranges from 320x400 256-color/pixel to 720x480 (overscan) 16-color/pixel. Of course you can add fancier video cards when available. Oh, and DMA hidensity floppy and SCSI ports of course (capable of about 1 megabyte/sec transfers if using synchronous SCSI, otherwise a meg would take a couple of seconds to transfer with normal SCSI). Both machines have standard PC clone keyboard and serial-mouse ports.

The standard info is that FHL's KMA will have an (optional?) 6809 card and Tandy GIME, but KLE's will not. KLE's software will include an OS9 utility 6809 emulator. The latest word (from Kev himself I believe, on Delphi) is that this emulator will not only interpret the 6809 machine code, but actually *translate* it into equivalent 68000 code as it goes! Thus the speed improves a lot. What a testimonial for Motorola's architectures!

KLE also has another good-news trick up their sleeve for CoCo3 owners, but I can't say anything about it yet.

68030 cards will cost a bundle no matter who you get them from. Plus the bus, memory, and peripherals to go that fast. I don't know much about either machine's upgrade paths in that respect, tho last I heard at least KLE has '030 plans.

> What am I gonna have to pay for the KMA???

Depends on which one, and what you get on the bus. As I've said, perhaps around \$800+ (U.S.\$) for a pretty nice 512K system. An ST or Amiga equivalent, hardware-wise, would be at least \$1200+, I've estimated, not including extra OS9 cost for those machines.

From : Bitnet.princeton (Wed Feb 28th)

Subject : FLH's KMA

I talked to Frank last Monday and he referred to the new computer he is making as "CoCo4". Base (on the motherboard) 6309 3MHz with K-bus. He has already 17 K-bus modules for sale (68000,68020 and 68030 + a lot more). He told me their KMA will be a BUFFERED CoCo bus, like the one we have on the CoCo3, 2 serial and one parallel port and 1 8bit A/D. Your 512K and/or 1Meg boards will plug right into the mother board. IBM (Ups), keyboard interface and RGB output. 100% compatible with OS9 Level II software. He said too that the prototype with Tandy OS9 Level II ran 40% faster than a stock CoCo3 and with the new update of OS9 (when ever it comes) the CoCo4 will run even faster. The price range for a populated CoCo4 board will be from \$200 - \$300. It is actually only a TURBO charged CoCo3 with a LOT of expansion capabilities in it.

END QUOTES

Well what does all this mean? The information presented above represents only part of the files from the Compuserve SIG which to me seems to be the most informative. Some other comments suggested that the ad in March Rainbow by Frank Hogg Labs for the 68030 QT K-System was their KMA - CoCo4. It seems to us however, that this is not the case, at least we hope not. The reason for saying this is that a base 68030 machine would start at about \$US 1100 and quickly add to \$US 5000 plus. That sort of price would certainly exclude most of us from owning one.

The final configuration of the Kenneth-Leigh and the Frank Hogg KMA is not crystal clear at this point, so we can only wait for more details.

Regards, Gordon

AUSTRALIAN OS9 NEWSLETTER

oooo00000oooo CONTENTS oooo00000oooo

For those of you who find it necessary to refer to back issues of the Newsletter, here is a contents listing of all of the Newsletters since the editorial panel transferred to Brisbane. We have decided that we should refer to these as the second volume of the series, hence the Vol 2 No. ... headings. All of the previous newsletters are now considered to be part of Volume 1. In future, we will be including a reference volume and number with each issue.

Vol 2. No. 01 (July 1988)

Page 01. Editorial ... Don Berrie
Page 02. Ramdisk Device Descriptor ... Don Berrie
Page 02. Boot Stepping Rate ... Bob Devries
Page 03. Stylograph Overview ... Gordon Bentzen
Page 06. Membership Application Form

Vol 2. No. 02 (August 1988)

Page 01. Editorial ... Gordon Bentzen
Page 02. Dynacalc Patches ... Gordon Bentzen
Page 04. Iconedit (Basic09) ... Don Berrie
Page 09. User Friendly Windows ... Don Berrie
Page 10. Membership Application Form

Vol 2. No. 03 (September 1988)

Page 01. Editorial ... Bob Devries
Page 02. Letter ... George Francis
Page 03. C-Compiler Patch ... Bob Devries
Page 04. Review "MultiVue" ... George Francis
Page 07. Sculptor Example ... Brad McMaster
Page 09. Iconedit - Further Notes ... Don Berrie
Page 10. National Usergroup Membership List - 31 August 1988
Page 11. Membership Application Form

Vol 2. No. 04 (October 1988)

Page 01. Editorial ... Don Berrie
Page 02. MultiVue & 'C' ... Bob Devries
Page 03. Zap Part 1 ... Don Berrie
Page 04. Koronis Rift Review ... Nickolas Marentes
Page 05. OS9 Error Messages ... Don Berrie

Vol 2. No. 05 (November 1988)

Page 01. Editorial ... Gordon Bentzen
Page 02. Zap Part 2 ... Don Berrie
Page 03. Rogue Review ... Nickolas Marentes
Page 04. Starting with OS9 Part 1 ... Bob Devries
Page 05. MultiVue & 'C' Part 2 ... Bob Devries

Vol 2. No. 06 (December 1988)

Page 01. Editorial ... Gordon Bentzen
Page 02. Making the most of 'C' ... Ross McKay
Page 03. Starting with OS9 Part 2 ... Bob Devries
Page 04. Reading LSN0 ... Bob Devries
Page 05. Running a Terminal ... Don Berrie

AUSTRALIAN OS9 NEWSLETTER

Page 06. Patches to Zap ... Don Berrie

Vol 3, No. 01 (Jan - Feb 1989)

Page 01. Editorial ... Gordon Bentzen
Page 02. Making the most of 'C' ... Ross McKay
Page 03. Starting with OS9 Part 3 ... Bob Devries
Page 04. Running a Terminal ... Don Berrie
Page 05. Convert Temperature 'C' ... Bob Devries

Vol 3, No. 02 (March 1989)

Page 01. Editorial ... Don Berrie
Page 02. Letter ... Ian Clarke
Page 03. Editors Reply ... Gordon Bentzen
Page 04. Making the most of 'C' ... Ross McKay
Page 05. Shellplus Review ... Don Berrie
Page 06. RS232 Interface ... John Usher
Page 07. A Database in 'C' Part 1 ... Bob Devries

Vol 3, No. 03 (April 1989)

Page 01. Editorial ... Bob Devries
Page 02. CoCo Hard Disks ... Don Berrie
Page 03. OS9 System Part 1 ... Gordon Bentzen
Page 04. Questions & Answers ... Don Berrie & Bob Devries
Page 05. A Database in 'C' Part 2 ... Bob Devries
Page 06. Public Domain Disk ... Bob Devries

Vol 3, No. 04 (May 1989)

Page 01. Editorial ... Gordon Bentzen
Page 02. Hex and Ascii Dump ... Ross Pratt
Page 03. OS9 System Part 2 ... Gordon Bentzen
Page 04. FD502 Motoron Mods ... Bob Devries
Page 05. Problems with Dsave ... Peter Barendrecht
Page 06. A Database in 'C' Part 3 ... Bob Devries
Page 07. C Compiler Environment ... Don Berrie

Vol 3, No. 05 (June 1989)

Page 01. Editorial ... Don Berrie
Page 02. The OSK Experience ... Don Berrie
Page 03. OS9 System Part 3 ... Gordon Bentzen
Page 04. Two Disk Programme Conv ... Bob Devries
Page 05. A Database in 'C' Part 4 ... Bob Devries
Page 06. Bootsplit - new mods ... Bob Devries

Vol 3, No.06 (July 1989)

Page 01. Editorial ... Gordon Bentzen
Page 02. Ohm's Law Demonstration ... Phil Frost
Page 03. EDIT Tutorial Part 1 ... Don Berrie
Page 04. PD Library Review ... Bob Devries
Page 05. OS9 System Part 4 ... Gordon Bentzen
Page 06. A Database in 'C' Part 5 ... Bob Devries

Vol 3, No. 07 (August 1989)

AUSTRALIAN OS9 NEWSLETTER

Page 01. Editorial	...	Bob Devries
Page 02. EDIT Tutorial Part 2	...	Don Berrie
Page 03. A Database in 'C' Part 6	...	Bob Devries
Page 04. Label Printer	...	Phil Frost
Page 05. Public Domain Library	...	Don Berrie
Page 06. Upperdir	...	Bob Devries

Vol 3. No. 08 (September 1989)

Page 01. Editorial	...	Gordon Bentzen
Page 02. 4GL's and OS9 Part 1	...	Ole Eskildsen
Page 03. OS9 Profile Review	...	Gordon Bentzen
Page 04. A Database in 'C' Part 7	...	Bob Devries
Page 05. Calendar Programme	...	Ole Eskildsen

Vol 3. No. 09 (October 1989)

Page 01. Editorial	...	Don Berrie
Page 02. EDIT Tutorial Part 3	...	Don Berrie
Page 03. OS9 'C' Compiler Review	...	Bob Devries
Page 04. A Database in 'C' Part 8	...	Bob Devries
Page 05. System Tidyup Notes	...	Don Berrie

Vol 3. No. 10 (November 1989)

Page 01. Editorial	...	Bob Devries
Page 02. 4GL's and OS9 Part 2	...	Ole Eskildsen
Page 03. Window Writer Tidyup	...	Don Berrie
Page 04. Super Controller Review	...	Bob Devries

Vol 3. No. 11 (Dec 1989 - Jan 1990)

Page 01. Editorial	...	Gordon Bentzen
Page 02. Lineater	...	David Eaton
Page 03. Resistor Calculator	...	Bob Devries
Page 04. CoCo's Future	...	Nickolas Marentes
Page 05. Clearing the Screen	...	Bob Devries and Don Berrie
Page 06. Windows Explained	...	Rob Unsworth
Page 07. Savings Calculator	...	Ole Eskildsen
Page 08. /dd Explained	...	Gordon Bentzen
Page 09. OS9 Tips	...	Don Berrie

Vol 4. No. 01 (February 1990)

Page 01. Editorial	...	Gordon Bentzen
Page 02. Basic09 Database	...	Jules Ambrosi
Page 03. Advanced Shellscript	...	Don Berrie
Page 04. Phoning North America?	...	Rob MacKay

Vol 4. No. 02 (March 1990)

Page 01. Editorial	...	Gordon Bentzen
Page 02. Tax, Income & Investment	...	David Eaton
Page 03. MVCanvas Review	...	Bob Devries
Page 04. Module Identification	...	Bob Devries
Page 05. Disk Zap Modifications	...	Don Berrie

oooooooooooo0000000000oooooooooooo

BASIC09 TOYS (OOOO OS9 ONLY)

The following piece of code is somewhat of a novelty, and unfortunately, will only be of real value to those of you who have an SSC Pak (modified hardware and driver if running Level II) in their MultiPak. Still with me? Well read on. It does demonstrate a number of techniques, however, and from that point of view, will be of some interest to all.

This little programme is almost certainly guaranteed to tempt you into putting your fist through your favourite computer. It really is quite infuriating to have your CoCo remind you of the time in it's monotonous voice every minute. But that is all it does!

The major point however, is that it demonstrates, quite effectively, how you can change the priority of a programme on the fly. You could set the priority before the process started, but that would take system resources from all of the other active processes for the whole time this programme is running. (Actually this is not strictly true, as the programme requires almost no overhead during the time that it is sleeping.) A much more elegant way is to vary the priority of the programme during execution, dependant on the output requirements of the programme itself.

When the system writes to a device such as the SSC Pak, it needs to do it's writing in real time. In other words, it is dealing with an unbuffered device in the real world, that needs its input all at once. Otherwise you would get (admittedly small) gaps between syllables, hardly what you would want when you are trying to make something intelligible come from the SSC. It's hard enough to understand even when the CoCo is only driving the SSC and nothing else, let alone running it from a multitasking system like OS9. A description of the workings of the programme follows.

First of all, we set aside memory for our variables, and then run Syscall to get the process ID for the programme. This will be needed later on when we want to give the programme extra priority during it's output routine. We then initialize all of the actual speech strings with (approximations of) the output strings. All the programme does then, is to scan the built-in Basic09 DATE# string, and parse the output into the way someone would (well almost) tell you the time. Sort of like the Telecom 1194 service. It then increases the priority of the process, and outputs the time strings to the SSC Pak. Having done that it simply resets the priority, and then puts itself to sleep for about one minute (depending on the number of concurrent processes running). We then loop back and start again.

It does show the versatility of the OS9 system in the way the output device is treated exactly the same as a file, or any other device. It also demonstrates a way to avoid a line like:

```
SHELL "Sleep 2800"
```

by the use of a direct system service request to the F\$Sleep call. Calling the sleep command via a SHELL Command Line would, if used, cause the sleep utility to be loaded each time the main loop is executed. Not much good if your system halts each time the disk is accessed. And of course, it shows you a technique for varying the priority of a process during execution.

To run the program, you have two options. It can be run from Basic09, simply by loading the procedure and typing run. <BREAK> will terminate the Procedure.

As an alternative, and I think the better way, you can pack the module and have it inserted in your Execution Directory. The Pack Command does this automatically. Then you can call the programme from the command line by simply typing its name, followed by an ampersand (&) to make it a concurrent process. To terminate the process, you will either have to have made a note of the process number when the shell forked the process (eg the shell reports this by printing something like &004), or you will have to examine memory to find the process number of RunB (with the "procs" command), and then use the built-in shell command, kill, to kill the relevant process number. In this instance, you will have to have the RunB module in your Execution Directory. In either case, the programme needs to have access to the Syscall function to be able to run the system service request to call the F\$ID, F\$SPrior and F\$Sleep SVC's.

```

PROCEDURE telltime
BASE 0
DIM sec,calcode,procid:BYTE
DIM hours,mins1,mins2:BYTE
DIM mtens(6),ampm:STRING(20)
DIM teens(21):STRING(15)
DIM osa,osb,osc:STRING(80)
TYPE registers=cc,a,b,dp:BYTE; x,y,u:INTEGER
DIM regs:registers
regs.a=0
regs.y=0
calcode=#00
RUN syscall(calcode,regs)
procid=regs.a
teens(0)=" "
teens(1)="wun "
teens(2)="tew "
teens(3)="threee "
teens(4)="phore "
teens(5)="phive "
teens(6)="sicks "
teens(7)="sevon "
teens(8)="eight "
teens(9)="ninen "
teens(10)="ten "
teens(11)="eelevun "
teens(12)="tweliv "
teens(13)="thurtean "
teens(14)="phore tean "
teens(15)="fifteen "
teens(16)="sickstean "
teens(17)="sevuntean "
teens(18)="eightean "
teens(19)="ninen tean "
teens(20)=" "
mtens(0)="o "
mtens(1)=" "
mtens(2)="twentee "
mtens(3)="thurty "
mtens(4)="phore tee "
mtens(5)="phiftee "
OPEN #ssc,"/ssc":WRITE
10 hours=VAL(MID$(DATE$,10,2))
mins1=VAL(MID$(DATE$,13,1))
mins2=VAL(MID$(DATE$,14,1))
IF hours=12 AND mins1=0 AND mins2=0 THEN
mins1=1
mins2=20
ampm="noon"
GOTO 15
ENDIF
IF hours=0 THEN
hours=12
ampm="eeven"
IF mins1=0 AND mins2=0 THEN
mins1=1
mins2=20

```



```

ampm="midnight"
GOTO 15
ENDIF
GOTO 12
ENDIF
IF hours>11 THEN
IF hours>12 THEN
hours=hours-12
ENDIF
ampm="peeyem"
ELSE
ampm="aayem"
ENDIF
12 IF mins1=0 AND mins2=0 THEN
mins1=1
mins2=20
GOTO 15
ENDIF
IF mins1=1 THEN
mins2=mins2+10
ENDIF
15 regs.a=procid
regs.b=200
callcode=#00
RUN syscall(callcode,regs)
PRINT #ssc,"the tyme is "
PRINT #ssc,teens(hours); teens(mins1); teens(mins2); ampm
regs.b=128
RUN syscall(callcode,regs)
regs.x=2800
regs.b=0
callcode=#0A
RUN syscall(callcode,regs)
GOTO 10
CLOSE #ssc

```

Please excuse my phonetics. You may be able to improve on these spellings. I would be interested to hear from anyone who has written any other Basic09 code which uses the SSC Pak.

The hardware patches to the SSC Pak to allow it to work at the faster clock speed of Level II were published in the U.S. Rainbow. My thanks to Bob Devries for the patches to the SSC driver.

If you have any questions or comments, please don't hesitate to call me.

Cheers, Don Berrie (07) 375-3236.

oooooooooooo0000000000oooooooooooo

CC3Go and the ShellPlus Path command.
By Bob Devries.

Those of you who have the enhanced 'shell' programme called 'shellplus' (Vers 2.1) by Rom Lammardo, will probably be using the 'path' command to make life easier. This is especially true for those of us with hard disk drives, as they are usually set up with a large number of sub-directories. In my own system, for instance, I have the path command:

```
path=/HQ/CMDS /HQ/CMDS/MVUE /HQ/SCRIPTS
```

AUSTRALIAN OS9 NEWSLETTER

This makes it easier for OS9 to find various files when I want to use them.

You will no doubt have found, as I did, that although you can set a path for a shell you start from the command line or from the startup file, you could not do so for the TERM device. The reason for this is that the shell which runs the startup script file is terminated before the shell which runs the TERM device. As well as that, this new shell has its parameters set by the OS9 system in the CC3Go programme.

The assembler file I have included here is a modified version of CC3Go, with the parameters changed so that the TERM device will also have the path command set on startup. You'll find the necessary paths set in lines 49 to 54 of the assembler file. These paths may be changed to any path names which you use regularly, but I think the maximum may be 80 characters.

To use this new CC3Go, type in the assembler source, remembering that only from the label column needs to be typed for the assembler. Then you can assemble it using ASM like this:

```
ASM cc3go.asm l,o=cc3go #20k
```

Now you need to create a new OS9Boot file and remove the old cc3go and replace it with the new one, and then os9gen to a new disk.

In the assembler source file there MUST be a 'fcb \$20' between the path fcc's, just as there would be if you typed it on the command line.

Anyway here is the code for CC3Go as output from my ASM programme.

```

00001          nam    CC3Go
00002          ttl    modified to provide paths for TERM
00003          *      use    /dd/defs/os9defs
00004          ifpl
00006          endc
00007 0001      vers   equ    $01
00008 0000 87C001D6  mod    endmod,name,prgrm+object,vers,start,endmem
00009 D 0000      buffer rmb    $0200
00010 D 0200      stack  rmb    $0100
00011 D 0300      endmem  equ    .
00012
00013 0000 43433347 name    fcs    "CC3Go"
00014 0012 05      edit    fcb    5
00015 0013 204F532D banner  fcc    " OS-9 LEVEL TWO VR. 02.00.01"
00016 002F 0D      fcb    $0D
00017 0030 0A      fcb    $0A
00018 0031 20202020 fcc    "    COPYRIGHT 1986 BY"
00019 0047 0D      fcb    $0D
00020 0048 0A      fcb    $0A
00021 0049 2020204D fcc    "    MICROWARE SYSTEMS CORP."
00022 0063 0D      fcb    $0D
00023 0064 0A      fcb    $0A
00024 0065 2020204C fcc    "    LICENSED TO TANDY CORP."
00025 007F 0D      fcb    $0D
00026 0080 0A      fcb    $0A
00027 0081 20202020 fcc    "    ALL RIGHTS RESERVED."
00028 0099 0D      fcb    $0D
00029 009A 0A      fcb    $0A
00030 009B 0A      fcb    $0A
00031 009C          banend  equ    *
00032 009C 2F4830  hddir   fcc    "/H0"
00033 009F 0D      fcb    $0D
00034 00A0 2F48302F hxdir   fcc    "/H0/"

```

AUSTRALIAN OS9 NEWSLETTER

```

00035 00A4 436D6473 cmddir fcc 'Cmde'
00036 00A8 0D fcb $0D
00037 00A9 20202020 fcc "....."
00038 00AE 5368656C shell fcc "Shell"
00039 00B3 0D fcb $0D
00040 00B4 20202020 fcc "....."
00041 00B9 4175746F autoex fcc 'AutoEx'
00042 00BF 0D fcb $0D
00043 00C0 20202020 fcc "....."
00044 00C5 53544152 statat fcc 'STARTUP -P'
00045 00CF 0D fcb $0D
00046 00D0 20202020 fcc "....."
00047 00D5 693D2F31 shedat fcc 'i=/1'
00048 00D9 20 fcb $20
00049 00DA 70617468 pthdat fcc 'path=/HQ/CMDS' path command + first path
00050 00E7 20 fcb $20 space character to separate path names
00051 00E8 2F48302F fcc '/HQ/CMDS/MVUE'
00052 00F5 20 fcb $20
00053 00F6 2F48302F fcc '/HQ/SCRIPTS' you may add extra paths after this one
00054 0101 0D noparm fcb $0D don't forget spaces between them
00055 0102 20202020 fcc "....."
00056 0107 pthend equ *
00057 0107 57021000 deftim fcb $57,$02,$10,$00,$00,$00
00058 010D 308D00C1 start leax rti,pcr point x to interrupt vector
00059 0111 103F09 os9 f#icpt set interrupt vector
00060 0114 103F0C os9 f#id get proces ID
00061 0117 0680 ldb ##80
00062 0119 103F0D os9 f#sprior set priority to 128
00063 011C 308DFF53 leax banner,pcr point to copyright banner
00064 0120 108E0089 ldy #banend-banner get message length
00065 0124 8601 lda ##01
00066 0126 103F8A os9 i#write and write to stdout
00067 0129 308DFFDA leax deftim,pcr get default date and time
00068 012D 103F16 os9 f#stime and start the clock with it
00069 0130 308DFF70 leax cmddir,pcr point x to CMDS dir name
00070 0134 8604 lda #EXEC. set execute mode
00071 0136 103F86 os9 i#chkdir change directory to it (if possible)
00072 0139 308DFF5F leax hddir,pcr point to hard drive root dir
00073 013D 8603 lda #UPDAT. use update mode
00074 013F 103F86 os9 i#chkdir and attempt to chd to it
00075 0142 2509 bcs nohard
00076 0144 308DFF58 leax hxdir,pcr point to hard drive CMDS dir
00077 0148 8604 lda #EXEC. use execute mode
00078 014A 103F86 os9 i#chkdir and do a chx to it
00079 014D 3460 nohard pshs u,y save regs
00080 014F 103F0C os9 f#id get process ID (again)
00081 0152 2579 bcs quit quit if error
00082 0154 30C4 leax buffer,u point x to 512 byte buffer
00083 0156 103F18 os9 f#gprdesc copy process descriptor into it
00084 0159 2572 bcs quit quit if error
00085 015B 31C4 leay buffer,u point y to buffer
00086 015D 8E0000 ldx ##0000 set for block #0
00087 0160 0601 ldb ##01 only 1 block
00088 0162 103F4F os9 f#mapblk map block into workspace
00089 0165 2566 bcs quit quit if error
00090 0167 8655 lda #$55 set warm-start flag byte
00091 0169 A70B71 sta D.CRStrt,u store it at warm-start flag
00092 016C E0C84A ldd D.SysPrc,u get system process descriptor address

```

AUSTRALIAN OS9 NEWSLETTER

```

00093 016F 33C8      leau  d,u
00094 0171 33C820    leau  P%D10,u    point to default IO area
00095 0174 31A820    leay  $20,y
00096 0177 C60F      ldb   #$0F      move 15 bytes to local storage
00097 0179 A6A5      loop  lda   b,y
00098 017B A7C5      sta   b,u
00099 017D 5A        decb
00100 017E 2AF9      bpl   loop
00101 0180 308DFF2A    leax  shell,pcr  point x to 'shell' name
00102 0184 338DFF3D    leau  statat,pcr point u to parameter data
00103 0188 C00100     ldd   #$0100    a=typ/lan,b=size of data area
00104 018B 108E0010   ldy   #$0010    y=size of parameter area
00105 018F 103F03     os9   f$fork    fork the shell to run startup file
00106 0192 2539      bcs   quit      quit if error
00107 0194 103F04     os9   f$wait    wait for startup to finish
00108 0197 308DFF1E   leax  autoex,pcr point to 'AutoEx' name
00109 019B 338DFF62   leau  noparm,pcr point to parameter area
00110 019F C00100     ldd   #$0100    a=typ/lan,b=size of data area
00111 01A2 108E0001   ldy   #$0001    y=size of parameter area
00112 01A6 103F03     os9   f$fork    fork 'AutoEx' if possible
00113 01A9 2503      bcs   nofork
00114 01AB 103F04     os9   f$wait    wait for 'AutoEx' to finish
00115 01AE 3560      nofork puls  u,y      restore regs
00116 01B0 308DFF21   leax  shedat,pcr point x to shell parameter area
00117 01B4 31C4      leay  buffer,u   point y to buffer area
00118 01B6 C632      ldb   #pthend-shedat b=length of parameter area
00119 01B8 A680      loop2 lda   ,x+      move b bytes from x to y
00120 01BA A7A0      sta   ,y+
00121 01BC 5A        decb
00122 01BD 26F9      bne   loop2
00123 01BF 308DFEED   leax  shell,pcr  point x to 'shell' name
00124 01C3 C00100     ldd   #$0100    a=typ/lan,b=size of data area
00125 01C6 108E0032   ldy   #pthend-shedat y=size of parameter area
00126 01CA 103F05     os9   f$chain   chain 'shell' and never return
00127 W 01CD 7FFFAB    quit  clr   $FFAB  here on some errors set block #0 in task 1
00128 01D0 0E63      jmp   D.Crash   jump to CC warmstart routine
00129 01D2 3B        rti   rti      interrupt routine (does nothing)
00130 01D3 8F1396     emod
00131 01D6          endmod equ   *
00132          end

```

```

00000 error(s)
00001 warning(s)
#01D6 00470 program bytes generated
#0300 00768 data bytes allocated
#1F24 07972 bytes used for symbols

```

TRADING POST

WANTED: Rohan Brown is looking for an original copy of Tandy DeskMate 3 for the CoCo3, if you have a copy for sale, please contact Rohan 75 Pembroke Road, Mooroolbark Vic. 3138 - home phone 03 725 6583.

FOR SALE: AVTEK Mini Modem II \$100.00 complete with cable and plugs. Contact Rob Mackay 27 Mawarra Street, Kingston Qld 4114 - home phone 07 209 3104