

AUSTRALIAN OS9 NEWSLETTER

Volume 8

August 1994

Number 6

EDITOR: Gordon Bentzen (07) 344-3881
SUB-EDITOR: Bob Devries (07) 278-7209
TREASURER: Jean-Pierre Jacquet (07) 372-4675
Fax messages (07) 372-8325
LIBRARIAN: Bob Devries
CONSULTANT: Don Berrie (079) 75-3537
SUPPORT: Brisbane OS9 Users Group

CONTENTS

OS9	OS9	OS9	OS9	OS9	OS9	OS9	OS9	OS9
OS9								OS9
OS9								OS9
OS9	Editorial					Page 2		OS9
OS9	Programming in Basic(09).....					Page 3		OS9
OS9	Tuning a Burke & Burke H.D.....					Page 5		OS9
OS9	OS-9 P.D. Library					Page 8		OS9
OS9	INFO ! by Rod Holden					Page 8		OS9
OS9	UltiMuse III users					Page 10		OS9
OS9	more INFO!					Page 11		OS9
OS9	OS-9 Usergroup Member list					Page 12		OS9
OS9								OS9
OS9	OS9	OS9	OS9	OS9	OS9	OS9	OS9	OS9

EDITORIAL MATERIAL
Gordon Bentzen
8 Odin Street
SUNNYBANK QLD 4109

LIBRARY REQUESTS
Bob Devries
21 Virgo Street
INALA QLD 4077

AUSTRALIAN OS9 NEWSLETTER
Newsletter of the National OS9 User Group
Volume 8 Number 6

EDITOR : Gordon Bentzen
SUBEDITOR : Bob Devries

TREASURER : Jean-Pierre Jacquet
LIBRARIAN : Rod Holden

SUPPORT : Brisbane OS9 Level 2 Users Group.

Greetings to all,

This is it, the very last, last newsletter from the Australian OS-9 Usergroup. I cannot help but reflect on the last six years and wonder how we managed, with a few exceptions, to produce a newsletter each month.

It all started for us back in mid-1988 when three members of the local Brisbane, Queensland, OS-9 Usergroup decided to take on the task of reviving the "National OS9 Usergroup" which wound up some six months earlier.

So Don Berrie, Bob Devries and myself began the planning and production of the first "new" trial newsletter. This was mailed out in July 1988 to known OS-9 users. The response was encouraging so we pressed on (pun intended) and took subscriptions beginning September 1988.

Now I must admit that if it wasn't for the OS-9 knowledge and enthusiasm of Don Berrie and Bob Devries this whole project would have been very short lived. None of us, however, had a vision of this continuing for anything longer than a year or two.

We have been very proud to support OS-9, mostly the Level 2 version for the CoCo3, and are grateful for the support, contributions and suggestions from many of our members over the past six years.

A good deal of debate has taken place over those years about the future and "place" for OS-9 and this has been evident most recently on the Internet listserver as well as a number of other forums. I can also recall our concern at the news that Tandy was to drop the CoCo. Was this to be the end of the OS-9 Usergroup? - Well, NO.

I do not intend to take up that debate in this editorial but simply state that I, for one, have learnt a lot through my association with OS-9 and OS-9 users. It certainly holds true, in this case, that "the more one learns, the more one realises how much more there is to learn".

In this edition we have a number of very interesting articles, including one by Peter Edwards on the Burke

& Burke hard drive system for the CoCo. Bob Devries makes comment about the future of our P.D. library, and by the way, the P.D. stuff from Hawaii has now arrived.

There are some thirty odd 1.44MB disks from the OCN, OS-9 Community Network. Bob and a couple of helpers are at present sorting through these disks to see what we have got. Some of the files we already have, so some cross reference will be necessary.

Also included with this edition is our membership list, complete with mailing addresses, so that those interested may have a means of maintaining contact with other OS-9 users.

Bob Devries tells me that he has Email messages from the U.S. OS-9 Usergroup requesting a copy of our mailing list. Here it is! We do plan to stay in touch and look forward to hearing about any new developments.

It is with mixed feelings that we prepare this final newsletter now that the decision has been made to stop. Once a month comes around very fast as I am sure you are well aware. A monthly newsletter is put together, printed, addressed and mailed, then, somebody says "it's time to start thinking about our newsletter again". But didn't we just do one?

So, it will be a relief not to have to try for that monthly deadline all the time, but I am sure that we will miss this also.

We are not going to pack up OS-9 and store it in a cupboard somewhere, we will continue to use it, so all is not lost.

Finally, we are grateful for the support of many members who have continued to subscribe to the National OS9 Usergroup even though they have moved on to other operating systems and of course thanks to those members who have been active in this usergroup.

We wish you all the very best in your future endeavours and trust that you continue to enjoy your computing.

Farewell, Gordon Bentzen.

**Programming in Basic(09)
by Bob Devries**

I was recently asked to convert an RSDOS basic programme to work under OS-9's Basic09. It was only fairly short, so presented no problems in itself.

Nowadays, when I do any development work, I do so with my SECAD OSK computer, and Basic programming is no exception. I have Microware Basic for my SECAD AS-68K, and I have found that programmes written for Basic09 and MW Basic are interchangeable at source level.

However, as always, there are SOME exceptions. MW's Basic for OS-9/68000 was written in C (as opposed to Basic09 which was written in assembler), and so it uses the C convention of terminating strings with NULL byte (\$00). Basic09 uses a \$FF byte. Neither of them appear to use the length of string system that RSDOS basic uses internally.

The MW Basic can cause problems with this method of string termination, especially if you want to send control code strings to the printer.

I tried to be a NICE programmer, and set up the control codes in a string variable, like this:

```
DIM control:STRING

control = CHR
$(1b)+CHR$(5b)+CHR$(40)+CHR$(00)+CHR$(00)+CHR$(02)....
```

The problem here is, MW Basic gets to the first \$00, and figures that it's found the end-of-string, and stops sending anymore characters, regardless of whether I use PRINT or PUT.

If I use Basic09, a similar thing happens if I use a \$FF code in a printer control string.

Here's a little Basic PROCEDURE which I used to get around the problem. It is equally at home in Basic09 as in MW Basic.

To use this procedure, first open a path to your printer:

```
DIM printer:integer
OPEN #printer,"/p":WRITE
```

Then, to use, for example, double strike mode, run

the procedure like this:

```
RUN ptrctl(printer,"doublestrike","on")
```

Don't get the spelling wrong, or it won't work. Also, the 'doublestrike' and 'on' MUST be in lowercase, else it won't work. If, for example, you type 'ON' (in capitals) the procedure won't recognise it, and do the 'off' code. To make this code case-insensitive, would take a bit more code.

Bob Devries

```
PROCEDURE ptrctl
PARAM path:INTEGER
PARAM action:STRING
PARAM onoff:STRING
DIM number:INTEGER
DIM count:INTEGER
DIM char:BYTE
DIM found:BOOLEAN

(* PTRCTL - send printer control character *)
(* strings to printer *)
(* By Bob Devries. *)
(* InterNet: bob@splat.paxnet.com.au *)
(* PUBLIC DOMAIN - 22nd July, 1994 *)
```

```
found = FALSE
IF action = "expand" THEN
    found = TRUE
    IF onoff = "on" THEN
        RESTORE 100
    ELSE
        RESTORE 110
    ENDIF
ENDIF
IF action = "underline" THEN
    found = TRUE
    IF onoff = "on" THEN
        RESTORE 120
    ELSE
        RESTORE 130
    ENDIF
ENDIF
IF action = "condensed" THEN
    found = TRUE
    IF onoff = "on" THEN
        RESTORE 140
    ELSE
        RESTORE 150
    ENDIF
ENDIF
```

AUSTRALIAN OS9 NEWSLETTER

```
ENDIF
IF action = "compressed" THEN
    found = TRUE
    IF onoff = "on" THEN
        RESTORE 160
    ELSE
        RESTORE 170
    ENDIF
ENDIF
IF action = "doublestrike" THEN
    found = TRUE
    IF onoff = "on" THEN
        RESTORE 180
    ELSE
        RESTORE 190
    ENDIF
ENDIF
IF action = "emphasized" THEN
    found = TRUE
    IF onoff = "on" THEN
        RESTORE 200
    ELSE
        RESTORE 210
    ENDIF
ENDIF
IF action = "doubleheight" THEN
    found = TRUE
    IF onoff = "on" THEN
        RESTORE 220
    ELSE
        RESTORE 230
    ENDIF
ENDIF
READ number
FOR count = 1 TO number
    READ char
    PUT #path,char
```

```
NEXT count
END
(* The first data statement is the *)
(* number of code bytes *)
(* that follow for the selected code *)
(* These codes are for a Tandy DMP202 *)
(* (IBM ProPrinterXL compatible) *)
(* "expand" "on" code *)
100 DATA 3,$1b,$57,$1
(* "expand" "off" code *)
110 DATA 3,$1b,$57,$0
(* "underline" "on" code *)
120 DATA 3,$1b,$2d,$1
(* "underline" "off" code *)
130 DATA 3,$1b,$2d,$0
(* "condensed" "on" code *)
140 DATA 1,$f
(* "condensed" "off" code *)
150 DATA 1,$12
(* "compressed" "on" code *)
160 DATA 2,$1b,$3a
(* "compressed" "off" code *)
170 DATA 1,$12
(* "doublestrike" "on" code *)
180 DATA 2,$1b,$47
(* "doublestrike" "off" code *)
190 DATA 2,$1b,$48
(* "emphasize" "on" code *)
200 DATA 2,$1b,$45
(* "emphasize" "off" code *)
210 DATA 2,$1b,$46
(* "doubleheight" "on" code *)
220 DATA 9,$1b,$5b,$40,$04,$0,$0,$0,$22,$2
(* "doubleheight" "off" code *)
230 DATA 9,$1b,$5b,$40,$04,$0,$0,$0,$11,$1
```

oooooooooooooooooooooooooooooooo

**Tuning a Burke & Burke Hard Disk Drive
by Peter Edwards**

This case history of tuning up a B&B-connected hard drive is offered in the hope that it might help someone else who is about to try it. Experts are encouraged to point out the errors, and (in particular) to offer explanations of the interleaving measurements described at the end.

After getting the B&B running, thanks to Andrew Donaldson, I decided to check that it was wasting as little space as possible, and running as fast as possible. The only specifications I had were the results of running performance programs on a PClone, before the drive was removed for use with the CoCo. This claimed that the drive, a CDC StorageMaster 518, had 3 heads, 495 cylinders and 17, 512 byte sectors per track (ie. 34, 256 byte ones).

First of all, I rebuilt the descriptor assuming one extra cylinder. It worked! So I got greedy and retried with another one (ie. 2 more than was used on the PClone). FORMAT finished, WITHOUT ANY MESSAGE, immediately after accepting the disk name. (BTW, you are aware, aren't you, that many 40 track floppy drives can go to 41 or 42 tracks? All you have to do is patch the descriptor, and remember not to use such disks for sending to friends.)

Buoyed up by this success, I tried 36 sectors per track, two more than MessDOS uses, and four more than B&B suggest. This time, FORMAT failed with error 241 - Bad system sector, FORMAT aborted. Fair enough -

B&B Step-code ('xx' above)	0	1	2	3	4	5	6	7
Step-rates (microsec)	3000	45	60	18	200	70	30	18
COPY time (seconds)	32	33	32	32	32	32	32	33
CMP time (seconds)	54	55	54	46	47	46	54	54

Clearly, using a COPY command to measure step-rates is no good; COPY's buffers must so big that it rarely needs to switch between files. On the other hand, CMP must use small buffers, perhaps just a sector or two for each file. It therefore swaps furiously between the two files. This produces timing differences which stand out clearly. For MY drive, not necessarily YOURS, I chose a step code of 5, which happens to be B&B's recommendation for those who don't know what to do. Note that the step-CODES are not in step-RATE order.

The last tests were for the optimal interleave (EXPERTS, PLEASE TUNE IN HERE). The interleave is only honoured when the drive is formatted. In normal

it must have plonked the last two sectors on top of the first one or two of the same track. (Note that there is no point in trying an odd number of OS-9 256 byte sectors, as B&B pack them in pairs into the 512 byte sectors as used by MessDOS.)

Then I tried with 4 heads, instead of 3, a real long shot. Drive manufacturers don't leave spare heads and disk surfaces inside their products! -- FORMAT hung.

At this point, I figured I had squeezed as much capacity as possible out of it. Not too bad; the extra two sectors was obvious, but the extra cylinder was a nice surprise. Guess the moral is "Don't believe all you're told".

Now for speed. First of all, the step-rate. My benchmark was the following line:

```
DMODE /hd stp=xx; TIME COPY x.ar y.ar; TIME CMP  
x.ar y.ar; DEL y.ar
```

All commands had been LOADED into memory. File x.ar was created by running the contents of my CMDS directory through AR; any decent sized file will do. DMODE is on the List; TIME is on Delphi. Someone should get permission to put it on the List.

I got the following average results (don't fuss about the numbers; just look at the trends):

use, it is ignored, as the sector headers contain the sector number, and that is what is important. THIS TEST THEREFORE DESTROYS THE DATA ON YOUR DISK. It is also bloody slow, due to the need to reformat the disk for each test.

The benchmark command was:

```
DMODE /hd ilv=xx; FORMAT /hd; FREE /hd; TIME MEGAREAD  
</hd@
```

(The "FREE" command is included to force a seek to track zero.) This was done with a temporary device descriptor which cut down the number of cylinders to describe a disk of about 2MB (69 cylinders for mine).

AUSTRALIAN OS9 NEWSLETTER

This is to minimize the time spent reformatting. Once again, all commands were memory resident. Like TIME, MEGAREAD is from Delphi. I won't give the figures for this, as I did it for every second value between 0 and \$1E (DMODE uses hexadecimal!). These were graphed, and the dip in the graph was confirmed by checking the odd-numbered interleaves, and by remeasuring where it looked funny. All in all, 33 times :- (The graph looked like that shown in Fig 2 below.

The key features are:

- Horizontal from interleaves from 0 to 3 at 54 secs
- Near-vertical from 4 to the minimum of 6, where it got down to 40 secs
- Sloping line from 6 to \$1E, where it was 65 secs

I was unhappy with a few aspects of this, so I measured again with different commands. Because a large proportion of sequential multisector disk I/O, which is when interleaves matter most, is in the loading of binary modules into memory, I felt a benchmark involving loads should be tried too. So I wrote a Mickey Mouse program ("MEGALOAD") which repeatedly loads a large module from disk. The whole measuring process was then repeated.

For what it's worth, while testing MEGALOAD, it was obvious from the light on the disk drive that the CRC checking on module loading is done after the module is loaded, not while it happens. Disabling the CRC with the following modpatch directives:

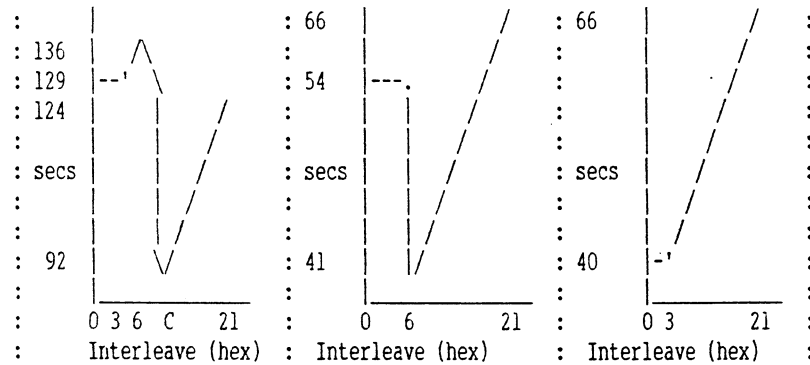
```
l os9pl
c 05b7 cc 4f
c 05b8 00 5f
c 05b9 02 39
v
```

speeded it up, and made it a better test of disk read performance. (I run without CRCs all the time now, because of the performance improvement. If my disk goes bad, loading a dud module or two is the least of my worries!)

First, the results of (if I remember correctly)

```
ECHO yyy ! TIME FORMAT /hd "dummy name" -r
```

or whatever params you need to have the whole thing run without any interactive prompts. See Fig 1. Note the small but definite hump between the plateau and the cliff.



FORMAT
Fig 1

MEGAREAD
Fig 2

MEGALOAD
Fig 3

Now for "TIME MEGAREAD </hd@" (Fig 2). Note that there is no hump.

Finally, "TIME MEGALOAD" (Fig 3). Note, no plateau or cliff.

The best spot (the start of the straight line on the right), varies: 3 for MEGALOAD, 6 for MEGAREAD (right where FORMAT's hump is) and \$C for FORMAT. This is what we expect, considering the different amount of

work being done after each read. The optimal spot according to FORMAT has a 20% performance hit if it's MEGALOAD which is telling the truth.

It would be interesting to alter MEGAREAD, so instead of reading in 1MB in 1024B chunks, it used 256 byte (ie. one logical disk sector) or 512 byte (one physical sector) instead. I have a feeling it would be closer to MEGALOAD then. I think the load system calls would read in single sector units, and probably

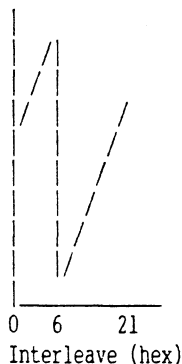
direct to the data's final location. C programs probably read into an RBF buffer first, then get copied to the user's program data area, and in the case of MEGAREAD, re-blocking would be needed as well. Unforch, I can't try this - my compiler is on my backup floppies, and I don't feel like reloading them, modding MEGAREAD, then scrapping the HD again!

I guess the hump in the FORMAT curve is because I am measuring the format time as well as the verify time; I probably have the sum of two quite different curves. A low-grade benchmark. A pity, because it's so easy.

Anyway, I chose an interleave of 7, just to get a bit of distance from the vertical section. (B&B recommend a value of 22 (\$16).)

However, I am unhappy about a few aspects of these measurements. At 6, the next sector arrives at the read-head just when MEGAREAD wants it. At 7, it has to wait for one sector time; at 8, it waits for two sector times. So far, so good - this explains the straight line as the interleave increases past a value of 6. Going the other way, at an interleave of 5, the next sector has just been missed, and an entire revolution has been wasted. Hence the vertical bit.

My problem is with the horizontal portion. I would have thought that going to an interleave of 4 means you waste one revolution, less one sector, and going to 3 would mean 1 rev less 2 sectors etc. This would produce a graph like:



This would mean that the graph wraps around.

Any ideas anyone? I have a few, but I'm unsure what to make of them:

One is the fact that pairs of OS-9 256 byte sectors are packed into single 512 byte ones before being written. I assume that reading consecutive 256 byte sectors results in a disk access on every second 'logical' read. Or is a 512 byte sector read each time, with half of the data being discarded?

What happens when you write randomly? Sequentially? Do you get 'lost' revolutions when writing a 256 byte sector due to the necessity to read its neighbour first?

Do PC disk controllers include some sort of buffer, which acts as a cache at low interleave values but not at high ones? (Off the wall, this one.)

At the time of writing, I have not had time to try asking the experts.

The MEGALOAD source and its build script (bld) have been tested with both asm and rma. In the source, the assembler-specific lines are marked in the right margin. Just (un)comment the ones you need. C.asm & c.link should be the same as rma & rlink - I haven't tested them.

As written, the file which is repeatedly loaded is /h0/compiler/c.pass1, which is just pass 1 of the C compiler - I'm sure you've got that! I chose it because

- it's one of the biggest modules I have (31kB I think)
- most people who would want to tune a HD will have it
- it's a single module file
- the module name matches the filename

In any case, anyone who doesn't like it can edit the source and re-assemble.

oooooooooooo0000000000oooooooooooo

The future of the PD Library
by Bob Devries

As you are no doubt aware, this is your last newsletter from us. With that, of course, all announcements of new Public Domain or commercial software for OS-9 will also cease.

necessary information about updates. Perhaps a three monthly mailing to interested OS-9 users would be a start.

I will be downloading whatever new software is available for OS-9/6809, OS-9/68000, and OS9000, from the InterNet FTP sites, and including these in the PD library.

The PD library also contains about 4 MB of RSDOS software. These files have actually always been there, but I haven't made an issue of it before.

Interested people should contact me at the address below, or phone me.

While I can't promise to maintain a complete commented list of the software, I can, and will supply to anyone who asks (and sends a disk, as before) a list of the names of the files.

Bob Devries
21 Virgo Street,
INALA. Qld. 4077

For those who are interested, and who are willing to keep up a communication with me, I will send the

PH: (07) 2787209

oooooooooooo0000000000oooooooooooo

IIIIII NNNN NN FFFFFF 000000 !!
II NN NN NN FF 00 00 !!
II NN NN NN FFFF 00 00 !!
II NN NN NN FF 00 00 !!
II NN NN NN FF 00 00
IIIIII NN NNNN FF 000000 oo

By Rod Holden

Hi, and welcome to Info. This particluar piece of software is in a file called Qtip4l.ar which is available under the OCN OS9_UTI directory, please read on;

Qtip was written out of need for a easy to use disk editor that was capable of aiding in the repair of a disk too. Everything has been made to be operated with the arrow keys and the spacebar, except for the places you are asked to enter a string, decimal or a hex value.

** Q T I P **

What is required: QTIP, the main program.

Written by Frans Lichtenberg

STARTING QTIP

(c) 1988,1989,1990 CFL Software

QTIP can be loaded with or without an argument. If omitted QTIP will prompt for a filename when the main screen is displayed. If a '-?' is used a argument a help message will appear. The command line for QTIP can look like this:

A disk management utility for the Colorcomputer 3 under the OS9 Level II operating system.

qtip /h0e

--> Open the hole media.

INTRODUCTION