AUSTRALIAN OS9 NEWSLETTER

EDITOR :
Gordon Bentzen
8 Odin Street
SUNNYBANK Qld 4109

(07) 345-5141

FEBRUARY 1990

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 = inhibi rms : d - directory file s - nc to owner w - write permit to own r - read permit to public pw - e permit to public BACKUP Syntax ge : Copies all data from one de ead error occurs s = single writes BASIC09 Syntax : Basic0 ge BUILD Syntax: Build filenar s from standard input CHD S nge working directory to specifi > Usage: Change execution 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 boot and system disks COPY Syntax data from one fil E Syntax : Date [t Opts: t = specify ame> Usage : Check directory for wor sters -m = save of unused cluster nly -o = print <devname> {<devn Del [-x] filenam s -x = delete x: Deldir directo yntax: Dir [e] [x] the file names y x=print executi] Usage : 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 makdir command o num K ECHO Syn tandard output ED riented text edito s text 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

EDITOR : Gordon Bentzen

HELPERS : Bob Devries and Don Berrie

SUPPORT : Brisbane OS9 Level 2 User Group.

Does OS-9 have a future? Or perhaps more appropriately the question should be, does OS-9 have a future in the personal computer world? If the enthusiasm of the dedicated OS-9ers is to have a significant bearing, then ØS-9 will certainly have a future. The membership of our National User Group continues to grow and most members seem eager for new knowledge and equally eager to help others.

While OS-9 is supported by a number of computers, the Tandy Color Computer is the most commonly used by our membership. Whilst many CoCo owners may have been a little concerned by the decision of Intertan Australia to drop the CoCo, we have continued to source hardware and software from overseas. Many of us continue to upgrade our favourite CoCo in one way or another.

There is an almost endless supply of quality programs and utilities available from the the U.S. and our members here in Australia continue to develop new software for all sorts of applications. The reputable suppliers in the U.S. provide a very professional approach to every request. In this edition our own Rob Mackay has submitted some hints on placing orders from U.S. suppliers. (Thanks Rob.)

When Intertan Australia discontinued the Color Computer from the Australian product range, we heard that Tandy in the U.S. also had plans to drop the CoCo. We understand that it was dropped from their range in Canada a few months ago, and we have correspondence from Intertan which confirms that the Color Computer has now been dropped from the product line in the U.S. So there you have it, no more CoCo's from Tandy. Intertan Australia have advised that a listing of companies and individuals dealing in Color Computer hardware and software products is at present being prepared. They have made a note to send us a copy in due course.

We have seen mention of a new CoCo 4 through the U.S. OS-9 user group. An OS-9 computer produced, not by Tandy, but by the U.S. OS-9 User Group. Details of such a machine are unconfirmed at this stage. We can only hope that a CoCo 4 is something more tangible than a wish list item.

Do you use a hard disk ? Up to now, I don't, but this is about to change. OS-9 is such a powerful and absolutely intriguing operating system that everybody seems to be making the investment (that's not what my wife would call it) in a hard disk system. It almost seems that if I don't go hard disk, I will soon be the only one in Australia running OS-9 from floppies. So, OS-9 is far from dead, even the CoCo 3 is far from dead.

In this edition, we present a submission from Jules Ambrosi which, as he describes, is a simple Database written in Basic09. Jules provided a disk with the source code and 'Docs' as well as a Packed version for us to trial. I have run his Database and must report that I find it very effective. (Thanks Jules.) This Basic09 program is well worth the effort it will take to type out the source code listing. Jules comments that when running from a floppy, you can expect a bit of a wait when using such things as "find" and "sort". I used this Database with the the file in a ramdisk, and in this mode it seems to run like a 'blur'.

I know that we make a lot of reference to the CoCo and present articles which are in many cases CoCo specific. We would like to include OS-9 articles for other systems as well, so how about sending us something about your favourite machine.

I will leave you to ponder the question of OS-9's future in the personal computer world and trust that you find something of interest in this month's edition.

Gordon Bentzen.

A BASIC09 DATABASE PROGRAMME

The following programme and its accompanying description were sent to us by one of the more experienced members of the National User Group. We applaud his efforts, and his decision to share with all of us the fruits of his labours.

We feel obliged, however, to point out that the structured nature of Basic09 really cries out for MODULAR programming by the use of procedures. In Jules' programme, he makes considerable use of line numbers, and GOSUB's to those line numbers. This makes the programme fairly difficult to read, and extremely difficult to debug. The use of procedures would have certainly made the programme easier to understand, much easier to debug, and allow for easy future modifications and enhancements.

Having said all that, this programme really is a fine effort. It should show to all of us just what can be done with perseverance and patience.

Thank you Jules, on behalf of all of the members, for your efforts. Ed.

*** HELP for Database09 *** December 1989.

Database09 is a basic database manager for OS-9 systems. It requires Basic09, Runb, Del, and Copy. It should run on any OS-9 Level 2 system but don't hold me to that. It was written on a CoCo III with 30 meg Hard Drive. Your printer must be connected and turned on while using DataBase09. Records in Database09 have a fixed format, but this may be changed by altering the source code. You get 5 strings of 64 characters each, 200 records per file. Each string (item) has a title. Use the Create option to open a new file. Give it a filename, name each of the items, and decide how many records in the file.

Set the filename and printer path defaults with New filenames. A CR at the printer name input will default to '/P'. You can send printer output to a textfile by entering a filename here. Then you can include it in a word processor file. Use the View/Edit option to input data into the file. The prompts here are self-explanatory. You may edit the titles of the items by editing record #0 (zero) at any time.

Since a database may be several hundred records long, I included a Search for string function. Say you need to find a phone number in a mailing list, but can't remember the person's last name? Just type in the first name (get the capitals right!) or anything else you think is in their file. DataBase09 will search until it finds the first occurrence of the name anyplace in any item, in any record. Caution: be prepared for a wait in a long database! You can also print your findings, to create a mailing list of all your friends in a certain city, ect. A further note is that on my 30 meg hard drive, it does not take long to go through my big database, but on a floppy disk, be prepared for a wait. Sorry about that, but OS9 needs a hard drive.

You may wish to sort your data alphabetically. You can do this, too, and sort on any item. Two cautions here: the sort is limited to the first 10 characters of the item sorted on. Also, say you have two 'Smith, John' records, with different addresses. They will be sorted so one is after the other, but in the SAME order they were in the original, unsorted file.

DataBase09 was written out of necessity in about 4 hours, with another 4 hours of adding bells & whistles. I needed a way to keep track of all the people name and address for my BBS's list. I will not be constantly updating this program for new features, so if you want improvements, write to me. I'd appreciate it, if you would continue to give me credit for my effort by leaving my name intact in the source code. Thanks, and enjoy!

Address all questions and queries to;- Jules Ambrosi
P.O. Box 341,
Niddrie, 3042.

Notes about Version 2.0 to those who had version 1. Not too much has changed. A search function has been added to the View/Edit section, so you need not leave this part just to search for a new record. The old search function has been retained as Find and Print, which is what it does best anyway. It can print all records with the target text, ect.

```
PROCEDURE DataBase09

0000        REM * DataBase OS9 version 2.0
001D        REM * by Jules Ambrosi
0032        REM * With thanks to S. Robertson for
             \so much help
0063        REM * A simple database for OS-9
             \ Level 2
008B        REM * Look for the text help file
             \called Database09.doc
00C1        ON ERROR GOTO 340
00C7        TYPE file=one:STRING[64];
             \two:STRING[64];three:STRING[64];
             \four:STRING[64]; five:STRING[64]
0103        DIM rec:file
010C        TYPE index=name:STRING[10];
             \rnum:INTEGER
0122        DIM sort(200):index
0130        DIM temp$:index
0139        DIM cmp:STRING[10]
0145        DIM pass,j,top,bot,a,x,num,recnum,
```

```
          \recsize:INTEGER
016C      DIM interchange:BOOLEAN
0173      DIM title(5):STRING[64]
0184      DIM field(5):STRING[64]
0195      DIM prnt(5):STRING[1]
01A6      DIM path,output:BYTE
01B1      DIM a$,cls,ti$,prt$:STRING[1]
01C9      recsize=SIZE(rec)
01D3      cls=CHR$(12)
01DB      path=4
01E2      output=5
01E9      out$="/P"
01F2      REM FILES QUESTION
0203 10   GOSUB 280
020A      PRINT "        <N>ew file"
021F      PRINT "        <O>ld file"
0234      PRINT
0236      PRINT "         Choice? ";
024B      GET #0,a$
0254      PRINT
0256      IF a$="N" OR a$="n" THEN 220
026E      PRINT
0270      INPUT "Filename? ",fl$
0282      REM MAIN MENU
028E 100  GOSUB 280
0295      PRINT "       <V>iew/Edit records"
02B2      PRINT "       <N>ew filenames"
02CB      PRINT "       <F>ind and Print"
02E5      PRINT "       <S>ort database"
02FE      PRINT "       <P>rint database"
0318      PRINT "       <C>reate new database"
0337      PRINT "       <O>S-9 Shell Commands"
0356      PRINT "       <Q>uit"
0366      PRINT
0368      PRINT "       Choice ? >";
037D      GET #0,a$
0386      q$="NVFSPCOQnvfspcoq"
039D      FOR x=1 TO LEN(q$)
03AF        IF a$=MID$(q$,x,1) THEN 110
03C5      NEXT x
03D0      GOTO 100
03D4 110  ON x GOTO 120,150,230,260,200,220,140,
           \130,120,150,230,260,200,220,140,130
041E 120  GOSUB 280
0425      PRINT "Set your input and output
           \filenames."
044D      PRINT
044F      INPUT "Input filename? ",fl$
0467      IF fl$="" THEN
0473        PRINT "You must enter a filename."
0491        PRINT "Press ENTER ";
04A2        GET #0,a$
04AB        GOTO 120
04AF      ENDIF
04B1      PRINT "Enter pathname of printer(i.e.
           \'/P' or"
04DB      INPUT "output textfile:( i.e. filename
           \ ",out$
0503      IF out$="" THEN out$="/p"
0517      ENDIF
0519      GOTO 100
051D 130  GOSUB 280
0524      REM QUIT PROGRAM
0533      REM Open overlay window
0549 135  SHELL "display 1b 3d 01"
0560      SHELL "display 1b 22 01 C 8 30 B 02 00
           \00"
0586      REM Choose options *
0599      PRINT
059B      PRINT "     Enter <1> to shell to OS9"
05BD      PRINT
05BF      PRINT "     Enter <2> to restart
           \program"
05E4      PRINT
05E6      PRINT "     Enter <3> to quit"
0600      PRINT
0602      PRINT "    ";
060B      INPUT Z
0610      REM Close overlay window
0627      SHELL "display 1b 23"
0638      SHELL "display 1b 3d 00"
064C      ON Z GOTO 140,10,136
0660 136  PRINT "Sure you want to Quit (Y/N)? ";
0685      GET #0,a$
068E      PRINT
0690      IF a$="Y" OR a$="y" THEN
06A5        PRINT cls;
06AB 137    END
06B0      ENDIF
06B2      GOTO 100
06B6 140  PRINT cls; "Shell commands"
06CF      PRINT
06D1      INPUT "OS9>:",sh$
06DE      IF sh$="" THEN 100
06ED      SHELL sh$
06F2      PRINT
06F4      PRINT "Completed. Press ENTER ";
0710      GET #0,a$
0719      GOTO 100
071D 150  GOSUB 280
0724      PRINT "View, Edit, or Search records"

0745      PRINT
0747      PRINT "Working";
0753      ot$=""
075A      IF out$<>"/p" AND out$<>"/P" AND
           \ out$<>"/P1" THEN
077B        CREATE #output,out$
0785        ot$="FILE"
0790      ENDIF
0792      recnum=0
0799      OPEN #path,fl$
07A3      SEEK #path,0
07AC      GET #path,rec
07B6      GOSUB 290
07BA      WHILE NOT(EOF(#path)) DO
07C5        PRINT ".";
07CB        GET #path,rec
07D5        IF rec.one=" " THEN 160
07E8        recnum=recnum+1
07F3        SEEK #path,recnum*recsize
0801      ENDWHILE
0805 160  top=recnum-1
0813      recnum=1
081A      SEEK #path,recnum*recsize
0828      WHILE NOT(EOF(#path)) DO
0833        GET #path,rec
083D        IF recnum=0 THEN
0849          GOSUB 290
084D        ENDIF
084F        GOSUB 300
0853        GOSUB 330
0857        PRINT "<N>ext, <B>ack, <P>rint,"
0873        PRINT "<G>oto, <L>ast, <F>irst,"
088F        PRINT "<E>dit, <Q>uit, <S>earch: ";
08AE        GET #0,a$
08B7        PRINT
08B9        IF a$="Q" OR a$="q" THEN 190
08D1        IF a$="S" OR a$="s" THEN
08E6          oldrec=recnum
08EF          GOSUB 330
08F3          INPUT "Text to search for: ",find$
090F          PRINT "Working";
091B          FOR recnum=1 TO top
```

```
092C            PRINT ".";
0932            SEEK #path,recnum*recsize
0940            GET #path,rec
094A            GOSUB 300
094E            FOR i=1 TO 5
0960              IF SUBSTR(find$,field(i))<>0
                \THEN 180
0977              NEXT i
0982            NEXT recnum
098D            recnum=oldrec
0996            PRINT
0998           PRINT "Not found. Press ENTER ";
09B4            GET #0,a$
09BD          ENDIF
09BF          IF a$="P" OR a$="p" THEN
09D4            IF ot$="FILE" THEN
09E4              GOSUB 320
09E8            ELSE
09EC              OPEN #output,out$:WRITE
09F8              GOSUB 320
09FC              CLOSE #output
0A02            ENDIF
0A04          ENDIF
0A06          IF a$="N" OR a$="n" THEN
0A1B            recnum=recnum+1
0A26          ENDIF
0A28          IF a$="B" OR a$="b" THEN
0A3D            recnum=recnum-1
0A48          ENDIF
0A4A          IF a$="F" OR a$="f" THEN
0A5F            recnum=1
0A66          ENDIF
0A68          IF a$="L" OR a$="l" THEN
0A7D            recnum=top
0A85          ENDIF
0A87          IF a$="G" OR a$="g" THEN
0A9C            INPUT "Goto Record #",recnum
0AB1          ENDIF
0AB3          IF a$="E" OR a$="e" THEN
0AC8 170        GOSUB 330
0ACF            PRINT "Edit which field (Q=Quit)?
                \";
0AEF            GET #0,a$
0AF8            PRINT
0AFA            IF a$="Q" OR a$="q" THEN
0B0F              GOSUB 310
0B13              SEEK #path,recnum*recsize
0B21              PUT #path,rec
0B2B              GOTO 180
0B2F            ENDIF
0B31            ac=ASC(a$)
0B3B            IF ac>53 OR ac<49 THEN 170
0B53            a=VAL(a$)
0B5D           PRINT cls; "Make changes. ENTER=No
               \change."
0B83            PRINT
0B85            PRINT a; " "; title(a)
0B95            PRINT field(a)
0B9D            READ #0,b$
0BA6            IF b$="" THEN 170
0BB5            field(a)=b$
0BC1            GOTO 170
0BC5          ENDIF
0BC7 180      SEEK #path,recnum*recsize
0BD8        ENDWHILE
0BDC 190    CLOSE #path
0BE5        IF ot$="FILE" THEN
0BF5          CLOSE #output
0BFB        ENDIF
0BFD        GOTO 100
0C01 200    GOSUB 280
0C08       PRINT "Print records in '"; fl$; "'."
0C27        PRINT
0C29        PRINT "Print titles (Y/N)? ";
0C42        GET #0,ti$
0C4B        PRINT
0C4D        IF ti$="" THEN 100
0C5C        INPUT "Starting record? ",start
0C75        INPUT "  Ending record? ",ender
0C8E        IF ti$<>"Y" AND ti$<>"y" THEN
0CA3          FOR x=1 TO 5
0CB3            PRINT "   Print field "; x; "? ";
0CD0            GET #0,prnt(x)
0CDD            PRINT
0CDF          NEXT x
0CEA        ENDIF
0CEC        IF out$="/p" OR out$="/P" OR
             \out$="/P1" THEN
0D0D          OPEN #output,out$:WRITE
0D19        ELSE
0D1D          CREATE #output,out$
0D27        ENDIF
0D29        PRINT "Working..."
0D37        OPEN #path,fl$
0D41        SEEK #path,0
0D4A        GET #path,rec
0D54        GOSUB 290
0D58        FOR recnum=start TO ender
0D6C          SEEK #path,recnum*recsize
0D7A          IF EOF(#path) THEN 210
0D87          GET #path,rec
0D91          IF rec.one=" " THEN 210
0DA4          GOSUB 300
0DA8          IF ti$="Y" OR ti$="y" THEN
0DBD            WRITE #output,"RECORD #"; recnum
0DD2          ENDIF
0DD4          FOR num=1 TO 5
0DE4            IF ti$="Y" OR ti$="y" THEN
0DF9              WRITE #output,title(num); " ";
               \field(num)
0E11            ELSE
0E15              IF prnt(num)="Y" OR
               \prnt(num)="y" THEN
0E30                WRITE #output,field(num)
0E3D              ENDIF
0E3F            ENDIF
0E41          NEXT num
0E4C          WRITE #output," "
0E56        NEXT recnum
0E61 210    CLOSE #output
0E6A        CLOSE #path
0E70        GOTO 100
0E74 220    GOSUB 280
0E7B        PRINT "Create a new file."
0E91        PRINT
0E93        INPUT "Filename? ",fl$
0EA5        IF fl$="" THEN 100
0EB4       PRINT "Enter titles for the 5 fields."
0ED6        INPUT "Field 1? ",rec.one
0EEB        INPUT "Field 2? ",rec.two
0F00        INPUT "Field 3? ",rec.three
0F15        INPUT "Field 4? ",rec.four
0F2A        INPUT "Field 5? ",rec.five
0F3F        INPUT "How many records (200 max)?
             \",num
0F63        CREATE #path,fl$:UPDATE
0F6F        PRINT "Working";
0F7B        SEEK #path,0
0F84        PUT #path,rec
0F8E        rec.one=" "
0F9A        rec.two=" "
0FA6        rec.three=" "
0FB2        rec.four=" "
0FBE        rec.five=" "
```

```
0FCA        FOR sk=1 TO num
0FDD          PRINT ".";
0FE3          SEEK #path,sk*recsize
0FF2          PUT #path,rec
0FFC        NEXT sk
1007        CLOSE #path
100D        GOTO 100
1011 230    GOSUB 280
1018        PRINT "Find records and print."
1033        PRINT
1035        INPUT "Text to find: ",find$
104B        IF find$="" THEN 100
105A        PRINT "Print fields that have it
             \(Y/N)? ";
1080        GET #0,prt$
1089        PRINT
108B        IF prt$="Y" OR prt$="y" THEN
10A0          PRINT "Print titles (Y/N)? ";
10B9          GET #0,ti$
10C2          PRINT
10C4          prt$="Y"
10CC          IF out$="/p" OR out$="/P" OR
             \out$="/P1" THEN
10ED            OPEN #output,out$:WRITE
10F9          ELSE
10FD            CREATE #output,out$
1107          ENDIF
1109        ENDIF
110B        recnum=0
1112        OPEN #path,fl$
111C        PRINT "Working";
1128        SEEK #path,recnum
1132        WHILE NOT(EOF(#path)) DO
113D          PRINT ".";
1143          GET #path,rec
114D          IF rec.one=" " THEN 240
1160          IF recnum=0 THEN
116C            GOSUB 290
1170          ENDIF
1172          GOSUB 300
1176          GOSUB 250
117A          recnum=recnum+1
1185          SEEK #path,recnum*recsize
1193        ENDWHILE
1197 240    CLOSE #path
11A0        PRINT
11A2        IF prt$="Y" THEN
11AF          CLOSE #output
11B5          PRINT "Done. Press ENTER";
11CB        ELSE
11CF          PRINT "Not found. Press ENTER";
11EA        ENDIF
11EC        GET #0,a$
11F5        GOTO 100
11F9 250    FOR i=1 TO 5
120E          IF SUBSTR(find$,field(i))<>0 THEN
1222            IF prt$="Y" THEN
122F              IF ti$="Y" OR ti$="y" THEN
1244                GOSUB 320
1248              ELSE
124C                FOR x=1 TO 5
125C                  WRITE #output,field(x)
1269                NEXT x
1274                WRITE #output,""
127D              ENDIF
127F            ELSE
1283              GOSUB 330
1287              PRINT
1289              PRINT "Is this it (Y/N)? ";
12A0              GET #0,a$
12A9              PRINT
12AB              IF a$="Y" OR a$="y" THEN
12C0                CLOSE #path
12C6                PRINT "Press ENTER when done.
             \";
12E2                GET #0,a$
12EB                GOTO 100
12EF              ENDIF
12F1              PRINT "Working";
12FD            ENDIF
12FF          ENDIF
1301        NEXT i
130C        RETURN
130E 260    GOSUB 280
1315        PRINT "Sort '"; fl$; "'
             \alphabetically."
1337        PRINT
1339        PRINT "Sort on which field (1-5)? ";
1359        GET #0,a$
1362        itm=VAL(a$)
136B        fl2$=fl$+"_BAK"
137A        PRINT
137C        PRINT "Making unsorted backup file: ";
             \fl2$
13A1        sh$="copy "+fl$+" "+fl2$
13B9        del$="del "+fl$+"_BAK"
13CF        SHELL del$
13D4        SHELL sh$
13D9        PRINT "Loading file."
13EA        OPEN #path,fl$
13F4        GET #path,rec
13FE        GOSUB 290
1402        recnum=1
1409        SEEK #path,recsize
1413        WHILE NOT(EOF(#path)) DO
141E          GET #path,rec
1428          IF rec.one=" " THEN 270
143B          GOSUB 300
143F          sort(recnum).name=field(itm)
1452          sort(recnum).rnum=recnum
1461          recnum=recnum+1
146C          SEEK #path,recnum*recsize
147A        ENDWHILE
147E 270    CLOSE #path
1487        bot=1
148E        top=recnum-1
1499        interchange:=TRUE
149F        pass=1
14A6        PRINT "Sorting file."
14B7        WHILE pass<=top-1 AND interchange DO
14CB          interchange:=FALSE
14D1          FOR j=1 TO top-pass
14E6            IF sort(j).name>sort(j+1).name
             \THEN
1502              interchange=TRUE
1508              temp$=sort(j)
1513              sort(j)=sort(j+1)
1525              sort(j+1)=temp$
1534            ENDIF
1536          NEXT j
1541          pass=pass+1
154C        ENDWHILE
1550        PRINT "Writing sorted file: "; fl$
156D        OPEN #output,fl$
1577        recnum=1
157E        SEEK #output,recsize
1588        OPEN #path,fl2$
1592        FOR idx=1 TO top
15A5          SEEK #path,sort(idx).rnum*recsize
15BA          GET #path,rec
15C4          PUT #output,rec
15CE          recnum=recnum+1
15D9          SEEK #output,recnum*recsize
15E7        NEXT idx
```

```
15F2      CLOSE #output
15F8      CLOSE #path
15FE      PRINT
1600      PRINT "Delete backup copy (Y/N)? ";
161F      GET #0,a$
1628      PRINT
162A      IF a$="Y" OR a$="y" THEN
163F        sh$="del "+fl2$
164E        SHELL sh$
1653      ENDIF
1655      PRINT "Done, press ENTER";
166B      GET #0,a$
1674      GOTO 100
1678 280  PRINT cls; "     *** DataBase 09 ***
           \"
16A1      PRINT "      by Jules Ambrosi      "
16C1      PRINT
16C3      recnum=0
16CA      RETURN
16CC 290  title(1)=rec.one
16DD      title(2)=rec.two
16EB      title(3)=rec.three
16F9      title(4)=rec.four
1707      title(5)=rec.five
1715      RETURN
1717 300  field(1)=rec.one
1728      field(2)=rec.two
1736      field(3)=rec.three
1744      field(4)=rec.four
1752      field(5)=rec.five
1760      RETURN
1762 310  rec.one=field(1)
1773      rec.two=field(2)
1781      rec.three=field(3)
178F      rec.four=field(4)
179D      rec.five=field(5)
17AB      RETURN
17AD 320  WRITE #output,"FILE: "; fl$; " **
           \RECORD #"; recnum
17D6      FOR x=1 TO 5
17E6        WRITE #output,title(x); " ";
           \field(x)
17FE      NEXT x
1809      WRITE #output,""
1812      RETURN
1814 330  PRINT cls; "Record #"; recnum
182B      FOR d=1 TO 5
183D        PRINT d; " "; title(d); " ";
           \field(d)
185A      NEXT d
1865      PRINT
1867      RETURN
1869 340  eror=ERR
1873      PRINT
1875      PRINT "Error #"; eror
1884      IF eror=246 THEN
1891        PRINT "Turn your printer on."
18AA      ENDIF
18AC      PRINT "Press ENTER";
18BC      GET #0,a$
18C5      GOTO 100
```

oooooooooo0000000000ooooooooooo

SHELLSCRIPTS WITH BELLS AND WHISTLES

Well, I think it's finally time that we came really to grips with all of the features of both the Windint windowing system, and all of the goodies that go with Shellplus 2.x.

One of my pet hates about any computing system, and the programmes written for it, is the tendency for programmers to write code for specific applications, and system environment setups. I guess that some of that criticism should be even levelled at the following shellscript, as it assumes a number of things about your system. (More of that later.)

I use my computer for a number of different activities. These include programming, letter writing, hacking, writing articles for this newsletter, and yes, even occasionally playing games. I spend a great deal of my time at the keyboard, and consequently I decided to purchase a hard drive to eliminate the constant need for swapping disks. For some time, it seemed that all I had done was to swap from one problem to another. Because now I think I am getting RSI from typing long directory names.

You see, the complexity of my hard drive has increased to the stage where I have some 120 directories containing more than 1000 separate files. I take pride in the fact that my hard disk is well structured, and I maintain a regular backup schedule. I always delete any temporary files, and in general try to keep my system tidy.

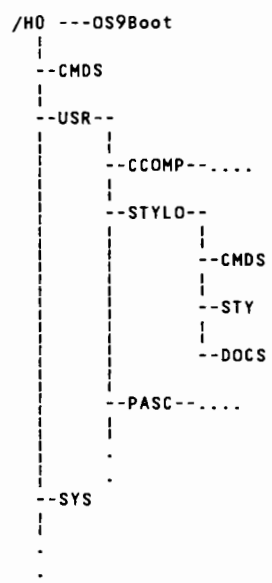
An unfortunate side effect of this is that I now have got to the stage of having some very long pathnames that lead to particular programmes that I frequently use. One of these is the Stylograph (C) wordprocessing system.

I get quite sick of typing lines that look like :

```
chd /H0/USR/STYLO/DOCS
chx /H0/USR/STYLO/CMDS
```

So I decided to do something about it and in doing so, I decided that I should try to use some of the features of both shellplus, and our great little system.

The structure of my hard disk looks (in part) like this :



.... and this leads to the problem of long directory names.

But first of all, lets take a look at the source code for the shellscript.

```
* Shellscript for starting an Editor
* in a Complex Directory Structure
*
* (C) D.A. Berrie 1990
*
*Start
load prompt
onerr goto erlp2
var.1="/d1/docs" ** Change this string to point
*                  to your document directory
display 1b 22 01 08 04 46 0f 02 02
display 1b 22 01 06 03 46 0f 05 07
display 1b 22 01 08 04 42 0d 02 00
display 1b 22 00 0a 05 3e 0b 02 00
tmode .1 pause pag=11
*loop1
dir %1
display a
display 1b 32 04
echo Type Full Directory Pathlist and press ENTER
echo or Type Filename and press ENTER to accept
display 1b 32 02
prompt:
var.1
if %1 > 0
cls
if -D %1
chd %1
goto loop1
endif
endif
*loop2
display 1b 23
display 1b 23
display 1b 23
display 1b 23
chx /d1/cmds ** Change this string to point to
*              your editor execution directory
path=/d0/cmds ** To allow for subshell access to
*                normal execution directory
```

```
Iniz W7
display 1b 20 02 00 00 50 18 00 01 01 >/w7
onerr goto erlp1
~~iniz w7~~
display 1b 21 >/w7
display 05 20 >/w7
display 1b 22 00 12 10 30 05 02 02 0c >/w7
display 1b 22 00 10 0f 30 05 03 04 0c >/w7
display 1b 22 00 12 10 2c 03 02 00 0c >/w7
display 1b 22 00 14 11 28 01 02 00 0c >/w7
prompt Edit File : %1 >/w7
display 1b 23 >/w7
display 1b 23 >/w7
display 1b 23 >/w7
display 1b 23 >/w7
display 02 36 2c >/w7
display 1f 24 >/w7
echo PLEASE WAIT WHILE MODULES ARE LOADED >/w7
display 1f 25 >/w7
stylo %1 <>>>/w7 ** Insert your normal editor  *
                      command filename
*erlp1
display 1b 21 >/1
display 1b 24 >/w7
deiniz w7
goto end
*erlp2
display 1b 23
display 1b 23
display 1b 23
display 1b 23
*end
unlink prompt
tmode .1 pause pag=24
```

I guess that I had better try to give some explanation as to what exactly happens in this shellscript, where changes might be made, and some of the 'idio(t)syncrasies' of both my programming and the system.

The first thing to note is that the system (I don't know which part) does not allow changing of the current device window (using the window select string: display 1b 21 >/wX) from a procedure file!!

If you type the command string manually (ie from your keyboard) it works fine. If you use calls to device window select routines from programmes written in higher level languages, they too work fine. If, however, you include the device window select command in a procedure file, is does not work. There is a way to get around this limitation however. The method is provided by the use of the data module feature of shellplus. You will, however, need to have access to the datamod utility, in order to change the textfile into a data module. This module, and the prompt utility mentioned in the next paragraph are public domain programmes, and are included with the latest shellplus archive. Because the datamod utility creates in memory data modules, you must load these packed shellscript modules into memory in order to execute them.

You will need access, either in memory, execution directory or execution path, to the following executable programmes in order to successfully run this shellscript.

load; prompt; display; tmode; dir; echo; cls; iniz; deiniz and unlink.

As the shellscript stands, it is setup to run the stylograph wordprocessor, but it could be equally used to run Sled, Edit or perhaps even window writer using my "window tidy" Basic09 procedure.

The first thing that the script does is to load the prompt utility (must be in current execution directory), to speed up file selection prompt writing. Prompt is simply an echo facility, but without a <CR> at the end!

We then set the first error trap, and define variable.1 to a string representing the pathname to your document directory. Then we open a number of overlay windows to give the nifty shaded box effect. One thing that should be noted here is that I am assuming that the utility is run from an 80 column windint/grfint type window. It doesn't matter whether it is a graphics or text type window.

Then the page length is set to 11, to allow for the overlay window size, and an address marker for the start of a loop is incorporated. Then we do a dir of the directory that was set in variable.1 above. The following line, display a, simply writes a blank line. We then change the colour of the foreground, and display a message.

After printing a prompt, we ask for terminal input (the line is : var.1), and then check if the string entered is a directory. If it is a directory, we change to it, and return to loop1. If it is blank, or a filename (or anything else!) we proceed to loop2. The next four lines simply close the overlay windows.

You may set the strings referenced in the next two lines to suit your own system. After processing those two lines, the next sequence, display 1b 20 02 .... opens a device window using window descriptor W7 (in this case). You may change this to any available descriptor, however, it is important that, when you run your finished shellscript, that the window is not already defined, nor should there be a shell running in it!! This is important. After this we reset the error trap, so that if an error occurs in the remainder of the shellscript, we can close the device window, and deiniz it before quitting.

We then initialize the window. Dependant on how your system is setup, the window may already be initialized. It does not matter if it already is inized, but if it's not, then we have to do this in order to successfully select it. And select it is exactly what we do on the very next line. After that, the new window should appear on the screen. The cursor is turned off, and we then open some further overlay windows on the new device window.

You will note that the lines read "display 1b 22 00 .. .. 0c >/w7". The extra digit (0c) in these lines clears the window. In order for an overlay window to be displayed on a device window other that the one from which it was crested, something actually needs to be written to it. The 0c accomplishes this. After these overlay windows are opened, we use the prompt command to write a message (without a <CR> at the end) so that it does not scroll off a one-line window, and then print the filename.

The next four display sequences then close the overlay windows. Because these windows do not save the screen underneath them, they appear to be left on the screen. Next we reposition the cursor, make the text flash, write another message, and then turn the flashing off. The next line is the line which actually calls the editor programme (stylo in this case) with the selected filename as an argument to it.

The remainder of the script simply handles shutdown and returns the system to it's original state.

If you have any problems or questions you can call me on (07) 375-3236. Cheers Don Berrie.

oooooooooo0000000000oooooooooo

DATA ON CALLS TO THE U.S.A. & CANADA.

****

Reverse Charge Calls dial 0101
Charge Rates ect dial 0102
Access Code 0011 1

Normal Rates $1.70/minute
Off Peak Rates $1.17/minute (Times as follows)

8.00pm to 6.00am Sunday to Friday
All day Saturday

Time Differences

Washington State 19 hours behind E.D.S.T.	AUST	U.S.A.
	3.00am	8.00am (previous day)
	6.00am	11.00am
	9.00am	2.00pm
	12.00am	5.00pm
New York 16 hours behind E.D.S.T.	AUST	U.S.A.
	12.00mn	8.00am (previous day)
	3.00am	11.00am
	6.00am	2.00pm
	9.00am	5.00pm
Quebec 18 hours behind E.D.S.T.	AUST	CANADA
	2.00am	8.00am (previous day)
	5.00am	11.00am
	8.00am	2.00pm
	11.00am	5.00pm
Arizona 17 hours behind E.D.S.T.	AUST	U.S.A.
	1.00am	8.00am (previous day)
	4.00am	11.00am
	7.00am	2.00pm
	10.00am	5.00pm

* NOTE:
The above are for AUST.Eastern daylight saving time. At the return of standard time the difference will INCREASE by ONE hour. This will mean that where a difference of 17 hours now exists, it will become 16 hours. A change from 7 hours to 8 hours time difference. Also, I understand that "Daylight Saving" is used throughout the U.S.A. & possibly Canada during their summer months, although I am not aware how extensively. For the exercise this would be an example.

Aust Daylight Saving. - Arizona is 17 hours behind us.
Aust Standard Time.(E.S.T.) - Arizona is 16 hours behind us.
Aust Standard Time.(E.S.T.) with U.S.A.or Canada on daylight saving then
- Arizona is 15 hours behind us.

Footnote.

During the last eight months I have phoned and faxed the U.S.A. and Canada more times than my wallet would have liked. For this reason I found that the above information as a hard copy, was most useful.

I hope this information may be of assistance to other OS-9'ers in their quest to search out or purchase from the extensive overseas market.

Last, but not least, the assistance provided by most suppliers has been superb. In some cases, only seven days from placing an order to delivery.

Regards, Rob MacKay.
(Brisbane Users Group)