

# OS-9 Newsletter

Volume II No. 11 <<< BELLINGHAM OS-9 USERS FORUM >>> November 30, 1991

## Basic09 Part-2

A Tutorial by Scott Honaker

In the September installment, we talked about the different data types available in Basic09. We even defined our own complex data type: `TYPE Employee=Name:STRING[25]; Street:STRING[20]; City:STRING[10]; Zip:REAL; Sex:BOOLEAN; Age:BYTE`

It is pretty obvious from our defined complex data type above, that a full blown database program could be put together by passing our `TYPE Employee` variables between several small basic09 procedures.

There are several key components to a database. The goal here is to break down the creation of a database into individual procedures, write a main core procedure and then systematically write individual procedures to expand the program. We'll call our database program Personal Database System or PDS for short.

To begin with, we will need a main menu, internal to PDS. This main menu will call all the procedures listed.

In this month's installment, I will include the code for `Menu`, `View`, `Create`, and `Add`. The following `TYPE` statement must be the first line in any procedure that will directly access the records in the database (most of them):

```
TYPE address=FName:STRING[10]; LName:STRING[15]; address1:STRING[20]; address2:STRING[20]; city:STRING[15]; state:STRING[2]; zip:STRING[10]; phone:STRING[14]
```

This `TYPE` statement is different than our earlier defined `TYPE` but a little more practical for a standard "rolodex" type database.

### MAIN MENU:

#### PROCEDURE PDS

This procedure will display the current record and database name by calling the `VIEW` procedure. It will also allow movement through the database up and down records and calling of the other procedures/functions. This procedure is also responsible for closing the database.

REM Personal Database System - Main Procedure

```
DIM Top,Current:INTEGER
```

```
DIM DBPath,OutPath:BYTE
```

```
DIM DBName:STRING[32]; Menu:STRING[1]; Printer:STRING[5]
```

### - IN THIS ISSUE -

Basic09 Database (Part-2)	1
Public Domain Reviews	5
CoCo Speedup Upgrade	7
OS9 Disk Fragmentation	8
OS9 Community Network Int'l	9
Xmodem Padding Solution	10
Classifieds	10
Club Activities Report	11

### SUBSCRIPTION INFORMATION

The OS-9 Newsletter is compiled and printed monthly by the Bellingham OS-9 Users Group. Subscription rates are \$5 for 6 months or \$10 for 12 months. Mail your subscription check to:

Rodger Alexander  
3404 Illinois Lane  
Bellingham, WA. 98226

### BENEFITS TO SUBSCRIPTION MEMBERS

As a subscribing member of the Bellingham OS9 Users Group, you enjoy the following benefits:

1. Monthly Newsletter
2. Public Domain Library (6-Megs)
3. Technical assistance (734-5806)
4. FREE Classified Ads
5. On-Line OS-9 Conference (676-5787)

```
DIM tempstr:STRING[2]
```

```
Printer="/p"
```

```
DBName="Untitled" \Top=0 \Current=0 \DBPath=0
```

```
REPEAT
```

```
  RUN View(DBPath,Current,DBName)
```

```
  PRINT
```

```
  PRINT "Main Menu: " \ PRINT
```

```
  PRINT "<C>create database <O>open database  pac<K> database"
```

```
  PRINT "<A>add record    <D>delete record  <E>edit record"
```

```
  PRINT "<F>find record   Find <N>ext record <S>ort database"
```

```
  PRINT "<P>print database Print <L>abels   <W>rite to file"
```

```
  PRINT "E<x>it program" \ PRINT \ PRINT "Menu Choice: ";
```

```
5  RUN inkey(Menu) \ IF Menu="" THEN 5
```

```
IF Menu="C" OR Menu="c" THEN RUN CreateDB(DBPath,Current, DBName) \ ENDIF
```

```
IF Menu="O" OR Menu="o" THEN RUN OpenDB(DBPath,Top,Current, DBName) \ ENDIF
```

```
IF Menu="A" OR Menu="a" THEN RUN AddRec(DBPath,Top,Current) \ ENDIF
```

```
IF Menu="D" OR Menu="d" THEN RUN DelRec(DBPath,Top,Current) \ ENDIF
```

```
IF Menu="E" OR Menu="e" THEN RUN EditRec(DBPath,Current) \ ENDIF
```

```
IF Menu="F" OR Menu="f" THEN RUN Search(DBPath,Current,TRUE) \ ENDIF
```

```
IF Menu="N" OR Menu="n" THEN RUN Search(DBPath,Current,FALSE) \ ENDIF
```

```
IF Menu="P" OR Menu="p" THEN
```

```
  ON ERROR GOTO 100
```

```
  OPEN #OutPath.Printer:WRITE
```

```
  RUN PrintDB(DBPath,Top,OutPath)
```

```
  CLOSE #OutPath
```

```
ENDIF
```

```
IF Menu="L" OR Menu="l" THEN
```

```
  ON ERROR GOTO 100
```

```
  OPEN #OutPath.Printer:WRITE
```

```
  RUN PrintLB(DBPath,Top,OutPath)
```

```
  CLOSE #OutPath
```

```
ENDIF
```

```
IF Menu="K" OR Menu="k" THEN RUN PackDB(DBPath,Top) \ ENDIF
```

```
IF Menu="S" OR Menu="s" THEN RUN SortDB(DBPath,Top) \ ENDIF
```

```
IF Menu="W" OR Menu="w" THEN RUN WriteRec(DBPath,Top) \ ENDIF
```

```
IF Menu=CHR$(10) OR Menu=CHR$(8) THEN Current=Current-1 \ REM Down/Left arrow
```

```
  IF Current=0 THEN Current=1 \ ENDIF
```

```
  IF Current=-1 THEN Current=0 \ ENDIF
```

```
ENDIF
```

```
IF Menu=CHR$(12) OR Menu=CHR$(9) THEN Current=Current+1 \ REM Up/Right arrow
```

```
  IF Current>Top THEN Current=TOP \ ENDIF
```

```
ENDIF
```

```
10 UNTIL Menu="X" OR Menu="x"
```

```
  IF DBPath<>0 THEN CLOSE #DBPath \ ENDIF
```

```
END
```

```
100 PRINT CHR$(12) \ PRINT \ PRINT \ REM Clear Screen
```

```
  PRINT "ERROR - "
```

```
  PRINT "    Cannot open printer port: "; Printer
```

```
  PRINT \ PRINT \ INPUT "Press <Enter> to continue".tmpstr
```

```
  GOTO 10
```

#### VIEW RECORD:

PROCEDURE View

View will display the current database name, record number and record contents. It is smart enough to

recognize when there are no records in the database and when it's looking at a deleted one.

```
TYPE address=FName:STRING[10]; LName:STRING[15]; address1:STRING[20]; address2:STRING[20]; city:STRING[15];
state:STRING[2]; zip:STRING[10]; phone:STRING[14]
PARAM DBPath:BYTE, Top:INTEGER; Current:INTEGER; DBName:STRING[32]
DIM rec:address

PRINT CHR$(12) \ REM Clear Screen
PRINT "Current Database: "; DBName \ PRINT
PRINT "Record number: "; Current
IF Current=0 THEN PRINT \ PRINT \ PRINT \ PRINT \ PRINT \ PRINT \ PRINT \ PRINT \ GOTO 10 \ ENDIF
SEEK #DBPath,(Current-1)*SIZE(rec)
GET #DBPath,rec
PRINT
PRINT rec.FName; " "; rec.LName
PRINT rec.
DIM rec:address
SEEK #DBPath.SIZE(rec)*(Current-1)
GET #DBPath,rec
PRINT
PRINT rec.FName; " "; rec.LName
PRINT rec.address1
PRINT rec.address2
PRINT rec.city; " "; rec.state; " "; rec.zip
PRINT "Phone: "; rec.phone
PRINT \ PRINT
10 END
```

#### CREATE DATABASE:

##### PROCEDURE CreatedB

This procedure will create a database of a specific name and return its path number (in DBPath) and the "current" record in Current (obviously zero).

```
PARAM DBPath:BYTE; Current:INTEGER; DBName:STRING[32]
DIM Option:STRING[1]

Current=0
ON ERROR GOTO 100
PRINT \ INPUT "Enter new database name: ",DBName
10 CREATE #DBPath,DBName:UPDATE
END

100 IF ERR=218 THEN \REM Error 218 - File already exists
PRINT CHR$(12) \ PRINT \ PRINT
PRINT "File: "; DBName: " already exists"
PRINT \ PRINT "Overwrite (Y/N)? ";
200 RUN inkey(Option) \ IF Option="" THEN 200
IF Option="Y" OR Option="y" THEN
SHELL "del "+DBName
GOTO 10
ENDIF
ELSE
PRINT CHR$(12) \ PRINT \ PRINT
PRINT "Error "; ERR: " has just occurred."
INPUT "Press <Enter> to continue",Option
ENDIF
END
```

### ADD RECORD:

PROCEDURE AddRec

This procedure will allow a record to be added at the end of the specified path and then increment the Top value and update the current record pointer.

```
TYPE address=FName:STRING[10]; LName:STRING[15]; address1:STRING[20]; address2:STRING[20]; city:STRING[15];  
state:STRING[2]; zip:STRING[10]; phone:STRING[14]
```

```
PARAM DBPath:BYTE; Top:INTEGER; Current:INTEGER
```

```
DIM tmp:STRING[32]; tmprec:address
```

```
IF DBPath=0 THEN END \ ENDIF \ REM No valid output path  
PRINT CHR$(12) \ PRINT "PDS - Add Record to open database:"
```

```
\ PRINT \ PRINT
```

```
10 INPUT "Enter first name: ",tmp
```

```
IF LEN(tmp)>10 THEN 10
```

```
tmprec.FName=tmp
```

```
20 INPUT "Enter last name: ",tmp
```

```
IF LEN(tmp)>15 THEN 20
```

```
tmprec.LName=tmp
```

```
30 INPUT "Enter first address line: ",tmp
```

```
IF LEN(tmp)>20 THEN 30
```

```
tmprec.address1=tmp
```

```
40 INPUT "Enter second address line: ",tmp
```

```
IF LEN(tmp)>20 THEN 40
```

```
tmprec.address2=tmp
```

```
50 INPUT "Enter city name: ",tmp
```

```
IF LEN(tmp)>15 THEN 50
```

```
tmprec.city=tmp
```

```
60 INPUT "Enter state name: ",tmp
```

```
IF LEN(tmp)>2 THEN 60
```

```
tmprec.state=tmp
```

```
70 INPUT "Enter zip code: ",tmp
```

```
IF LEN(tmp)>10 THEN 70
```

```
tmprec.zip=tmp
```

```
80 INPUT "Enter phone number: ",tmp
```

```
IF LEN(tmp)>14 THEN 80
```

```
tmprec.phone=tmp
```

```
Top=Top+1
```

```
SEEK #DBPath,(Top-1)*SIZE(tmprec)
```

```
PUT #DBPath,tmprec
```

```
Current=Top
```

```
END
```

THAT'S IT FOR THIS MONTH! To type in the procedures above, make sure Basic09 (merged with INKEYS, SYSCALL, GFX, GFX2) is in your commands (CMDS) directory, then type in the following from the OS9 prompt:

```
OS9>basic09
```

```
B:edit PDS
```

Now type in the Main Menu procedure PDS. Remember that the Basic09 Editor operates the same as the OS9 Line Macro Editor. You must leave a space between the prompt and the line of text. When you finish entering the PDS listing, enter **q** at the basic09 editor's "E" prompt.

```
E:q
```

DO NOT ENTER A SPACE FIRST. This will quit the edit mode. If you typed in any mistakes, they will be listed at this time with their address locations and the error types. You'll have to go back and find your mistakes. The listing must be error free.

After you have entered all four procedures (error free), run the program:  
 Brun PDS

#### HOMEWORK ASSIGNMENT:

The next most obvious procedure that needs to be written is one that will open the database records that you create. After all, what good does it do you to create a database and not be able to view it the next time you turn on your computer?

#### OPEN DATABASE:

PROCEDURE OpenDB

PARAM DBPath:BYTE; Top:INTEGER; Current:INTEGER; DBName: STRING[32]

This procedure will open a database of a specific name and return its path number (in DBPath), the highest record number (in Top) and the current record (in Current). It will be necessary to read the entire database to determine the highest record number so that the top value can be returned to the main procedure for use in other procedures. The current value is not really important here and can be set to any value from 1 to Top.

#### Additional optional features:

1. A list of available files would be a nice touch for an open menu.
2. Since the entire database will have to be read to determine the last record number, it would be a good opportunity to use a RAM disk if available. As it reads the disk version, it could write it to the RAM disk, then return the path number of the open RAM disk file to the main procedure. This would require an additional parameter so the main procedure will know to update the disk version before it exits.

If you have any questions, pull out your OS9 Level Two System Manual and start reading the Basic09 section.

## Public Domain

### *Reviews . . .*

by Rodger Alexander

#### DIVVY - by Mike Shook

This is a windowing utility that you cannot be without if you have more than 128K of memory. Divvy will create multiple windows on one screen, or more correctly, it will create multiple overlay windows "on the fly" with an active shell in each overlay.

Besides all of the "cool" things you can do with multiple windows on one screen, think of the really practical advantages of having multiple screens/processes. I find it extremely useful when I'm writing an article like this one.

For example, while I am using a text editor to write a review on a particular piece of software on one window using the lower half of the screen, I'm also viewing the documentation file for that software on a second window on the upper half of the screen.

I use to accomplish the same thing by toggling between two windows pressing the CLEAR and SHIFT-CLEAR keys. Better than MS-Dos, but still a pain. I could accomplish double overlay windows with multivue, but that was a bit of a hassle too.

Not only does Divvy create up to four overlay windows, but you can also specify optional colors.

The command line options are as follows:

Divvy ncnn n n n

||| | |Number: Optional Border Color  
 ||| | |Number: Optional Background Color  
 ||| | |Number: Optional Foreground Color  
 ||| | |Number Of Right/Bottom Windows (1 Or 2)  
 ||| | |Number Of Left/Top Windows (1 Or 2)  
 || |Letter Horizontal Or Vertical Division (H Or V)  
 | |Number Type of Screen (1,2, 5 thru 8)

#### POP Version 4.4 by Chris Swinefurth

This is another "on the fly" windowing utility. By simply typing "pop" from the OS9 prompt, a new 80 column text window is created with a window number label across the top of the screen. Each new "popped" window has different colors. Options are also available to "pop" VDG windows and select your own colors. An accompanying popconfig file makes it possible to change the default settings.

#### CMPFIX - by Roger Krupski

This is a handy little utility to have around, especially if you are one of those type who is always modifying binary programs. What this program does is to create a MODPATCH script file from your CMP output. Use CMPFIX instead of CMP to compare the original program with your modified version:

CMPFIX original < modified > CMPfile  
 MODPATCH < CMPfile

#### MELT - by Andy DePue

This one is just for fun and it does just what the name says... it melts! Well, actually the program itself does

not melt... it melts your text screen. And after the screen is dead and gone, it does it's best to restore the screen! You'll get a big kick out of this one.

#### ALIAS - by Tim Koonce (Tim Kientzle)

This is definitely not a new program, but often overlooked by OS9'ers due to a misconception. This program does not only create a new command name in your execution directory that executes an already existing command file, but it includes the specified options.

##### ALIAS ls dir e

Creates ls which executes dir with the "e" option. ls is now the same as dir e.

#### FREE - by Tim Kientzle (Tim Koonce)

This is a replacement for the Microware FREE command with several added features. **FASTER:** Up to 4 times faster than the original free command. **USER FRIENDLY:** Outputs relevant information in kilobytes, rather than in sectors, and decimal instead of hexadecimal. Also give total disk usage as a percent of the disk capacity. Options are available to force the output to be in bytes, sectors, clusters, or megabytes. **USER FRIENDLY INPUT:** Will accept device names with or without a leading slash, or directory names such as "." Will accept multiple device names on a single command line.

OS9> free /d0 /d1 /d2

Device	Capacity	Available	Used	%Full	Volume Name
/D0	360	265	94	26%	"DISTO HDBoot"
/D1	180	10	169	94%	"UTILITY6A"
/D2	720	483	236	32%	"HDSYS.DEFS.LIB"

OS9> FREE /H0

Device	Capacity	Available	Used	%Full	Volume Name
/H0	40,992	29,922	11,069	27%	"40MegHD"

#### KFORMAT - by Kelly Thompson

KFORMAT was written to make it easy to format a slew of disks. It allows you to select a single name for all the disks to be formatted, and allows you to select all format options (number of sides, tracks, etc.) It allows you to format any number of disks that you like, so it is even useful for formatting a single disk in a particular way, hence you don't have to look up how to do it.

**Kelly's Handy Formatter**

Loading OS-9 FORMAT command...

Please fill in the following information:  
(Default values flash !)

Enter device to format [eg. /dd, /f0, etc.] :

Enter number of sides [1 or 2] :

Enter number of tracks (cylinders) [35, 40, etc.] :

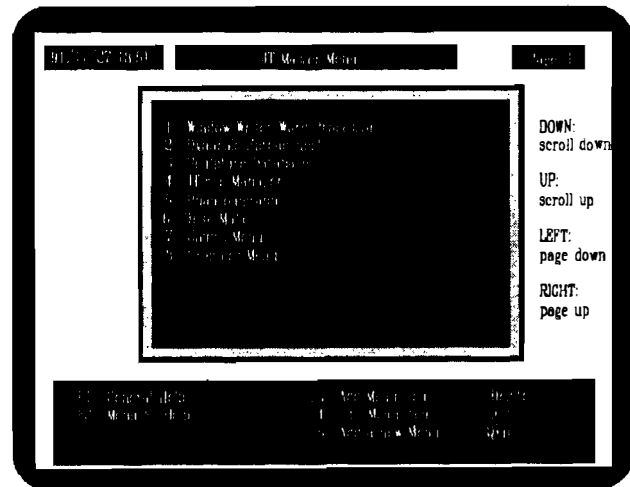
Logical format only? [Y/N] :

If you are formatting more than one disk:  
Should all disks have the same name? [Y/N] :

KFORMAT really shines when you want to format several disks. Ask you know, formatting a double-sided 40 (or 80) track disk can take a while! So, if you would like to leave the room...turn up the volume! KFORMAT will "whistle" at you when it's ready for the next disk. Literally!

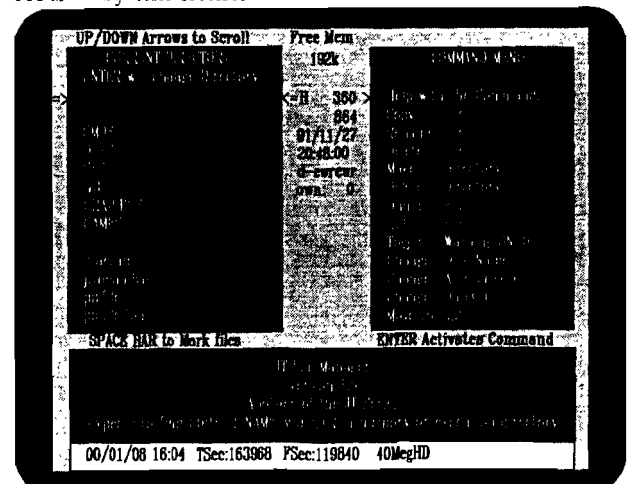
#### JTMENU - by Jim Hollier

JTMENU is a menuing interface for hard drive systems (or floppy) permitting you to create on-screen menus to make accessing and executing program applications and utilities so simple that even your 5 year old will have no problem with OS9. Unlike other "MENU" programs that read script files that you create with a text editor,



JTMENU has it's own built in formatted editor that prompts for each line of entry. PLUS you are not limited to only 3 or 4 shell command lines. This allows you much more power and flexibility. A complete text of instruction and sample script command files is provided from within the menu. Help file options for each menu item is possible and password protection for each menu screen or individual file is optional.

#### JTFM - by Jim Hollier



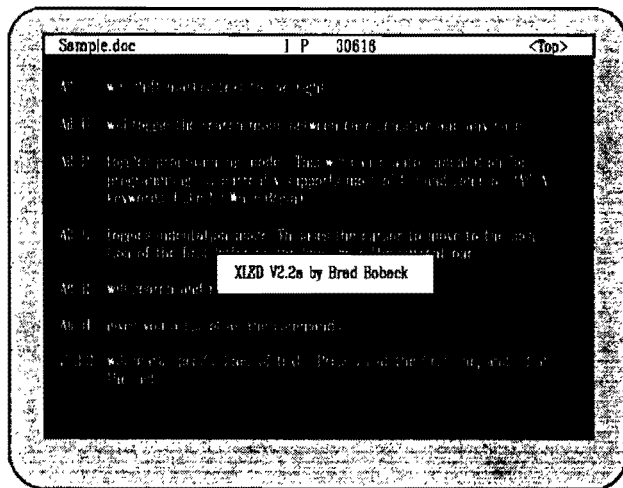
This is a file management program in which the data directory is listed in a window on the left side of the

screen and a list of command options is displayed in a window on the right side of the screen. Between the two windows is a file stat (fstat) display of the file that is pointed at in the left window. That's nice! But the real value of this file manager is the ability to **mark** multiple files to be copied or deleted or attributed or moved. WOW!

This program is so simple yet complete, you might want to consider using it as a "new user" interface for the OS9 operating system. OR, as a complete OS9 utility application.

**XLED - by Brad Boback**

Another text editor? YES it is. YES it **does** have the standard options such as block commands and global search and replace commands, etc. However, the one additional feature of this text editor over all other is that it can be set to Programming Mode which will cause auto-indentation when using keywords like: **{, If, While, Begin)**



**SURFACE CHECK UTILITIES:**

I've run across three utilities of this type. **STEST** by Bruce Isted, **DSCAN2** by E.R. Anderson and Tom Birt, and **SCAND** by Paul Pollock. **DSCAN** version 2.0 is the most attractive on the screen and is written in C, but **SCAND**, written in Basic09 is 5 times faster, reading a 720K floppy in just over 2 minutes. On the other hand, **STEST** runs 4 test on each sector and operates at a rate of 12.5 minutes per MegaByte on a hard drive or 2.5 hours/MegaByte on a floppy. **DSCAN2** runs at 12 minutes/720K floppy. The **DSCAN2** archive also includes a text file "BITMAP.DOC" that explains how to use a disk editor to modify your bitmap so OS9 will not write to sectors that have gone bad since the initial format.

**SDUMP - by Marie-Louis Marcoux**

This program is designed to run in background (sdump &) when started. Pressing **SHIFT/CTRL/ALT** keys simultaneously activates the dump of the current text window (80 or 40 columns) to the printer.

**MD - by Tom Birt**

If you're running OS9 Level Two on a color monitor, you'll love this one. This public domain utility is very similar to microware's **MDIR**, except it's "color coded". The **SYSTEM** modules are purple, **DRIVERS** are in green (flickering), **DESCRIPTORS** are in yellow, **MANAGERS** are in white, **PROGRAMS** are in turquoise, **SUBROUTINES** are in red and **DATA** files are in green (non-flickering). You can even specify a module name to be highlighted as a search aid. It's even faster than **MDIR**! What more can I say? You gotta get this one!

## Hi-Speed Upgrade

by Mark Johnson

I found out about this upgrade through the OS-9 echo on our local fido-net bulletin board, Columbia Hts. BBS (206-425-5804). The modification to your coco is simple: replace the original crystal on the CoCo motherboard with a 32MHz crystal. That's it! Almost too simple to be true... but it works!

One problem with the upgrade though is that it won't work on a television. The RFI unit will only go to 1.78MHz and that's why I think Tandy limited the CoCo's speed. There are also some reports that it is supposed to cause some troubles on some monitors, but people were putting diodes or something on the monitor cord to clear it up. On the CM-8 RGB monitor that Tandy sells, it is supposed to work fine with the upgrade. It worked ok for me and another member of our Longview/Kelso CoCo Club. Also someone got the RFI unit working by using both crystals. They put the old, slower crystal in the RFI unit itself so the screen updates would be at the old speed while the new 32MHz crystal replaced the old crystal on the motherboard ran the rest of the system at a full 2MHz.

- Mark Johnson, Columbia Hts. BBS -



# Fragmentation

OS9, as a disk-based operating system, is constantly moving files between main memory and your hard or floppy disks. The part of OS9 that is responsible for disk access is called RBF -- short for Random Block File manager.

RBF keeps track of where data is stored on your disks. It does this by dividing each disk into three regions:

- 1) The **IDENTIFICATION SECTOR**. The information in this region tells OS9 what type of disk you are using.
- 2) The **ALLOCATION BITMAP**. Each bit in this region represents one or more disk sectors. A set bit indicates that the sectors are being used to store data; a clear bit indicates that the sectors are available for storing additional data. Called **ABM** for short.
- 3) The **FILE SYSTEM**. Defines directories, files, and the data stored in each file.

As you can see, each of these regions serves a unique and essential purpose. Most computer disk systems, from CoCo BASIC to UNIX to large mainframe computers, keep track of data by similarly dividing their disks.

The **ABM** and file system change slightly every time you create, modify, or delete a disk file. When a file is created, OS9 searches the **ABM** for an unused area of the disk and places the file there; OS9 also sets additional **ABM** bits to indicate that the area is now in use. Extending the file causes OS9 to set more **ABM** bits; shortening the file clears the **ABM** bits for parts of the disk that are no longer needed. All of a file's **ABM** bits are cleared when the file is deleted.

Most of the **ABM** bits are zero on a blank, newly formatted disk. As you use the disk, the **ABM** and file system are revised over and over again. The **ABM** eventually becomes a random pattern of set and cleared bits.

OS9 must search through this

random **ABM** pattern whenever it creates or extends a file. Since the pattern is random, its "even odds" that when a cleared bit is finally located, the next bit will be set. If OS9 needs to extend the file by more than a few sectors, the unused sectors chosen by RBF are pretty likely to be scattered all over the disk!

This scattering effect is called **FRAGMENTATION**. There aren't any significant good results of fragmentation, but there are several bad ones:

- 1) OS9 must record the location of each sector used by a file in a part of the file called a **FILE DESCRIPTOR SECTOR (FDS)**. The **FDS** contains as many as 48 "segments", each of which is represented by a starting disk address and a sector count. Each time OS9 encounters a set bit in the **ABM**, it must start a new file segment by searching for a cleared bit. Once a file has 48 segments, OS9 can't make the file any larger -- even if each segment is only a single sector!
- 2) A file's segments may be scattered all over the disk. OS9 must constantly move the disk heads back and forth across the disk as it moves from segment to segment. The resulting overhead slows down disk access significantly.
- 3) All of that head motion creates excessive wear on your disk drive's mechanical parts.

Minor fragmentation is a natural side effect of disk operation, but when left untreated it gets worse and worse until an individual file or disk is stored in the most inefficient way possible.

## Common Fragmentation Myths

Many people believe that they have a fragmentation problem when there is a significant difference between the number of free sectors on a disk and the size of the largest free block (as reported by **DCHECK**). The truth is, the size of the largest free block has very little to do with the amount of disk fragmentation until it falls below about 2% of the device capacity. It's the size of the

other, smaller blocks that's more important, and **DCHECK** won't tell you about these. Several public domain utilities that better display fragmentation are available through the electronic information services. Another common myth is that you can eliminate fragmentation by backing up a disk, reformatting, and restoring. This is only true if you use a file-by-file backup utility; device-based utilities like **BACKUP** and **HDB** won't effect fragmentation at all. In addition, a certain amount of uncorrectable fragmentation is created by **ABM** bits set to conceal media defects. **DSAVE**, **FBU**, and **HDKIT** are typical file-by-file backups that can be used to reduce or eliminate fragmentation.

If you have two hard drives, you may be tempted to try to save time by using **BACKUP** to copy one drive to the other, then reformatting and using **DSAVE** to get rid of the fragmentation. This practice sometimes works, but it is dangerous if the two drives have different defective areas: **BACKUP** may destroy some data by copying it to a defective sector.

The three most reliable methods for reducing or eliminating fragmentation are:

- 1) If you have a spare hard drive, **FORMAT** it and mount it on your system. Then **DSAVE** your main drive to the spare drive. Now you can **DSAVE** the spare drive back to the main drive, eliminating fragmentation in the process.
- 2) For single drive systems, use a file-by-file backup utility to save the contents of the drive to multiple floppy disks. Then reformat the drive and restore it from floppies.
- 3) Use a hard disk optimizer (e.g. **REPACK**) to repack the disk without reformatting or saving to floppies.

## Symptoms of Extreme Fragmentation

When OS9 displays **ERROR 217** (Segment list full), this indicates that a file has reached the built-in fragmentation limit of 48 segments. Your disk itself may not be badly



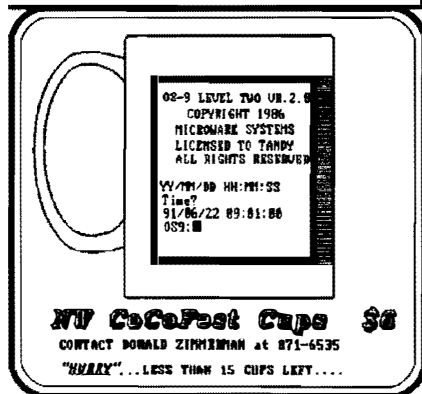
fragmented; you can find out by trying to reduce the file's fragmentation. Do this by renaming the file, copying it to its old name, and deleting the renamed version. If you're lucky and the disk isn't badly fragmented, the copy will have fewer segments than the original. Otherwise, it may be time to defragment the disk using one of the above procedures.

ERROR 248 (Device full) when running EZGen, Cobbler, Config, or OS9Gen is another symptom of fragmentation; this error means that OS9 couldn't find any group of clear ABM bits large enough to create a single-segment file. The disk may actually be full, or it may be badly fragmented. When DCHECK shows that a device's largest free block is smaller than about 2% of the device's capacity, the device may be badly fragmented. This is especially true if the total number of free sectors is much larger than 2% of capacity.

### Summary

Fragmentation is a natural side effect of disk operation. It is caused by the constant rearrangement of disk data as files are created, modified, and deleted.

Three problems caused by fragmentation are disk errors (e.g. "Segment List Full", "Device Full"), slower disk access, and increased mechanical wear. Although device-based backups don't effect fragmentation, you can reduce or eliminate it by periodically backing up and restoring your hard drive, or by using a hard disk optimizer program.



## OS9 Community Network International

The "OS9 Community Network International" is getting organized and several of our subscribers and local OS9'ers are actively involved. In the August issue I reported that a NEW "OS9 Users Group" was starting up to replace the old "OS9 Users Group". Since that report three months ago, Constitutions and By Laws have been written and the new name should advert some confusions in regards to the original OS9 Users Group.

I suspect the "Network" will be very successful due to its dependency on the FidoNet system available on RiBBS Bulletin Boards. FidoNet is truly an international telecommunications networking system, and almost every major city in North America has one or more Bulletin Boards tied into FidoNet (except Bellingham).

The goals of the OS9 Community Network International are to unify, foster, promote, improve, further and enhance, through its membership, high standards of perfection, efficiency and cooperation between all phases of the OS-9 6809, OS-9 68000 and OS-9000 Operating Systems, hereinafter referred to as the OS-9 Operating System; to do all such lawful acts and things necessary to further the best interests of the Network, and to formulate, determine and coordinate policies of mutual and international import and concern for such industry.

An OS9 Community Network International Newsletter, the "OS9 Community NetNews" will be posted on the CoCo and OS9 echos on FidoNet. Besides informational articles about OS9, the "NetNews" will include official meeting notices and serve as the official voice of the organization.

Below is a list of current officers. I did not list all of the officers, coordinators/librarians, only the board of directors and the individuals that are relevant to us.

### Officers:

International Coordinator: John Wight (FidoNet 1:345/200)

Secretary: Nancy Ward FidoNet 1:106/941

International Co-Coordinator: Don Vaillancourt (FidoNet 1:250/610)

International Librarian: Kevin Backs (FidoNet 1:163/239)

Newsletter Editor: Richard Lorbieski (FidoNet 1:106/941)

International Network Rep: Bill Nobel (FidoNet 1:140/26)

### Regional Librarians:

Region 11: Kerry Kowalski (FidoNet 1:154/888) {OS9 Newsletter Subscriber}

Region 17: (Wa. State) Mark Johnson (Columbia Hts. BBS) {OS9 Newsletter Subscriber}

Region 17: (British Columbia) Charles West

### Regional Coordinators:

Region 17: Dennis Mott (FidoNet 1:346/9) {DataWarehouse BBS Sysop}

Kerry Kowalski, better known as "Rocky" lives in Whitlaw, WI. If you have any questions and/or would like further information about the OS9 Community Network International, give Dennis Mott a call and leave a message on his BBS. His phone number is (509) 325-6787. Mark Johnson can be contacted on the Columbia Hts. BBS at (206) 425-5804.

Since the OS9 Newsletter is not associated with "the Network" our information is limited, but we will try to keep you posted as best we can. If you are really interested, call your local FidoNet BBS and register through the CoCo or OS9 echos.

## Classifieds: CoCo Stuff

### COCO HARDWARE:

512K CoCo-3	Multipak
CM-8 Monitor	20Meg Hard Dr.
Burke&Burke Intface	2-SingleSided Drives
TRS-80 Printer VII	Ser/Parallel Intface
Hi-Res Interface Deluxe	Joystick
Speech Sound Pak	Power Strip
CoCo Mouse	

### COCO SOFTWARE:

DynaCalc	Phantomgraph
T/S Spell	Deluxe Icon Ed.
TRSCOPY	OS9 C Compiler
Foxware C Lib.	OS9 Lev.II Dev Sys
OS9 Lev.II Op.Sys	OS9 Level I
Dala Master	Multi-Menu
FBU	The Zapper
B&B FileSysRepack	Multi-Vue
Thexder	Sub Battle
Flight Simulator-2	Rogue
Koronis Rift	Rescue on Factalus
Micro Illustrator	Disk Program Package
Packer	

### COCO MISCELLANEOUS:

Programming Books, OS9 Books, Misc.Floppy  
Disk, 60 issues of "The Rainbow"

**COMPLETE PACKAGE FOR \$999 (45% OF ORIGINAL PRICE)  
FREE DELIVERY**

Contact:

Dave Spencer  
P.O. Box 331  
Bridgeport, WA 98813  
(509) 688-7551 (after 5 p.m.)

## "How To" Video



**Installing "IRQ" Hack**  
**Installing 512K Ram Kit**  
**Installing B&B Hard Drive**  
**Upgrading Speech Sound Pak**  
**Installing a 2nd Floppy Drive**

**\$10**

**Plus \$2 Shipping**

**Call (206) 734-5886**

## Xmodem Padding Problems

- Delphi OS9 Sig -

Several people on the SIG have mentioned having problems after downloading binary files from Bulletin Boards when using XMODEM or YMODEM download protocol programs under OS-9.

The problem lies in the fact that XMODEM/YMODEM sends 128/1280-byte blocks of information at a time. When the end of the file is reached, any "short" block (less than 128/1280 bytes in length) is padded with some "invisible" character such as control-Z's, spaces, or nulls. This padding cause OS9 to reject the file as bad because the module header is incorrect, making OS9 think that the file is longer than the header says it should be.

The solution is simple:

- 1) Use ATTR and give the file the executable attribute (e).
- 2) Use the LOAD command and load the file into memory.
- 3) Use the SAVE command and save the file back to the disk using a new filename.
- 4) Use DEL and RENAME to get rid of the long file and rename the new file.
- 5) Use UNLINK to "unload" the file from memory.

That's it!

## Level-1 Modpatch Files

In the September issue of the Newsletter, page 5, was an extensive article on using MODPATCH for modifying OS9 Level-2 programs. However, the response I have received from several subscribers has been..."What about Level-1 patches". Who uses Level-1 anymore? Well, obviously there are several subscribers who are running OS-9 on CoCo-1's and CoCo-2's. So here are a couple of necessary MODPATCH files for Level-1.

### CC3DISK:

Since Tandy's version of OS-9 Level-1 doesn't really pay attention to the drive descriptors, the standard disk patches have to be made to the CCDISK driver module.

In the following patch files, insert the appropriate hexadecimal values for the desired results. Do not type in comments in parenthesis. Track step rates: 12=20msec, 11=12msec.



# Club Activities

Bellingham OS9 Users Forum - Longview/Kelso CoCo Club  
Mt. Rainier CoCo Club - Port O'CoCo Club - Seattle 68xxx Mug

## Seattle 68xxxMUG

The Seattle 68xxxMUG has it's monthly meeting on the top floor of Gugenheim Hall on the University of Washington Campus the first Tuesday of each month at 7:30 p.m. At the November 5th meeting, Donald Zongker brought his Tandy Color Computer-3 that was very recently installed in a horizontal baby AT PC case. Very compact with 3 floppy drives and 2 hard drives. All of the I/O port jack on the back were the standard DIN jack used on the back of the Color Computer so no adapters were needed.

A quick demonstration on the short comings of HOME PUBLISHER and the lack of support for Tandy's new DMP-135 printer was informally conducted, with a little input from everyone. Tandy's only response on using the DMP-135 with HOME PUBLISHER was to select the DMP-105 printer and set the parameters for 3 column mode and the DMP-135 would print 2 columns. The workable solution seems to be using a Disto 3-in-1 interface with the parallel printer output to the DMP-135, then selection one of the IBM printer drivers and setting the DIP switch on the DMP-135 to IBM mode.

Scott Honker then look over the meeting and presented his 4th installment on writing a Basic09 database. This session was to feature the presentation of the different procedures that individual's were assigned to

write. Unfortunately most of those members who had procedure assignments were not in attendance (they skipped class), so we were not able to see their work, although Scott did bring along his finished assignments as did Rodger, and we got a chance to see the new modules perform....Rodger's crashed!

The December meeting will feature another presentation of new Public Domain Utilities and Applications and hopefully a presentation of the "missing" Basic09 procedure files from our truant members.

- Rodger Alexander -

## Longview/Kelso CoCo Club

The Longview/Kelso CoCo Club will have their next meeting on December 10th at 7:00 and we will be setting up a FidoNet point/node address at the meeting. Also we plan on doing a bunch of 2MHz upgrades for the CoCo 3. In case you haven't heard about it yet, all you have to do it replace the clock crystal with a 32MHz crystal and thats it. Although you may have some interference on a TV and maybe a little interference on some monitors but the Tandy CM-8 has worked excellent so far without a glitch one. I just tested it and got about a 15% increase

- Mark Johnson -

No reports available from: Bellingham OS9 Users Forum, Mt. Rainier CoCo Club and Port O'CoCo Club.

10=6msec.

Track seek rates: 42=20mscc, 41=12msec, 40=6msec.

Head restore rates: 02=20msec, 01=12msec, 00=6msec.

CCDisk.pat

l ccdisk

c 1f9 13 xx (insert desired step rate)  
c 1ff 22 02 (shorten command settle delay)  
c 20b 04 41 (D0 double sided)  
c 20c 40 42 (D1 double sided)  
c 2dd 43 xx (insert desired seek rate)  
c 2e8 03 xx (insert desired head restore rate)  
c 2f9 a0 20 (shortn motor-on time delay)

v

Drive Descriptors:

Even though Tandy's version ignores the descriptors for all of the above information, it does look to the descriptors for the number of tracks. Therefore, it is necessary to patch the drive descriptor to change the track rates. The patch is the same for all drives (D0, D1, D2, etc.).

D0.pat

l d0

c 18 23 28 (change to 40 tracks)

v

Now simply type: modpatch ccdisk.pat and modpatch d0.pat to modify these two boot modules. To make these modifications permanent, cobbler a new system disk.

## Washington State BBS List

FAR POINT BBS - Seattle

RiBBS (Fido NET)  
(206) 285-8335

COLUMBIA HTS. BBS - Longview/Kelso

RiBBS (Fido NET)  
(206) 425-5804

DATA WAREHOUSE BBS - Spokane

RiBBS (Fido NET)  
(509) 325-6787

BARBEQUED RIBBS - Bellingham

PCBOARD (PC-NET)  
(206) 676-5787 Conference #5

OS-9 TACOMA BBS - Tacoma

RiBBS (Fido NET)  
(206) 566-8857

COCO EXPRESS BBS - Anacortes

RiBBS (Fido NET)  
(206) 293-1057

Bellingham OS-9 Users Group

presents . . .



Written by

Scott Honaker & Rodger Alexander

Mail \$2 + \$1 shipping fee to:  
3404 Illinois Lane, Bellingham, WA 98226

The OS-9 NEWSLETTER is published by The Bellingham OS-9 Users Group; Rodger Alexander, Editor. Back Issues are available for 75 cents (please specify month & year).

**OS-9 Newsletter**

3404 Illinois Lane

Bellingham, WA 98226