WAS MS-DOS CODE COPIED FROM CP/M?

Bob Zeidman

ZEIDMAN CONSULTING

AGENDA

- History of DOS and CP/M
- Forensic Analysis of Software
- Comparison of DOS and CP/M Code
- Comparison of Command Line Interfaces
- Comparison of System Calls
- Conclusion
  - What, if anything, was copied?
  - Was there copyright infringement?
- Q & A

HISTORY OF DOS AND CP/M

- Kildall invented CP/M, the basis for the microcomputer OS
- That fateful day 36 years ago: August, 1980
- Gates sent IBM to DRI
- Kildall didn’t meet
- IBM went back to Gates
- Microsoft bought QDOS from Seattle Computer Products
- Microsoft hired Tim Paterson to modify it and create PC-DOS (MS-DOS)
- Seattle Computer Products profited
- Gary Kildall and DRI profited

BUT…

- Was DOS an illegitimate copy of CP/M?
- Was code copied?
- Were commands copied?
- Were system calls copied?
- Is there a secret command in CP/M and DOS that proves copying?

FORENSIC ANALYSIS OF SOFTWARE

- Scientific analysis
- Quantifiable
- Not determined by legal issues
- Source code correlation
- Six reasons for correlation
  - Common algorithms
  - Commonly used identifier names
  - Common author
  - Third party code
  - Automatic code generation
  - Copying
- Iterative filtering

CLEANING THE CP/M CODE

- Removed things that are not source code
- Optical character recognition (OCR)
- Multiple passes of manual corrections
- Fix printer glitches
- Run CodeMatch of each file against itself
  - Comments as instructions
  - Instructions as comments
  - Strange identifiers
  - Incorrect OCR
- Reformatted code

COMPARISON OF DOS AND CP/M CODE

- Use the FileIdentify™ function of CodeSuite ®
- Load the source tree into the Understand tool from Scientific Toolworks; poke around
- Perform global searches within the source code files for the following terms:
  - The string “copyright”
  - Company names
  - Author names and initials
  - Any relevant terms

“CP/M” IN DOS

```
MSDOS.ASM:
     ; 1.12 10/09/81 Zero high half of CURRENT BLOCK after all (CP/M programs don't)
     STOSB ;Set it to zero (CP/M programs set low byte)
MSHEAD.ASM:
     STOSB ;Set it to zero (CP/M programs set low byte)
SYSCALL.ASM:
     STOSB ; Set it to zero (CP/M programs set low byte)
EXEC.ASM:
     XOR AX,AX     ; zero extent, etc for CPM
PRINT.ASM:
     DOCHAR:
          MOV     AL,BYTE PTR [BX]
          CMP     AL,1AH              ;^Z?
          JZ      FILEOFJ             ;CPM EOF
          CMP     AL,0DH              ;CR?
          JNZ     NOTCR
          MOV     [COLPOS],0
```

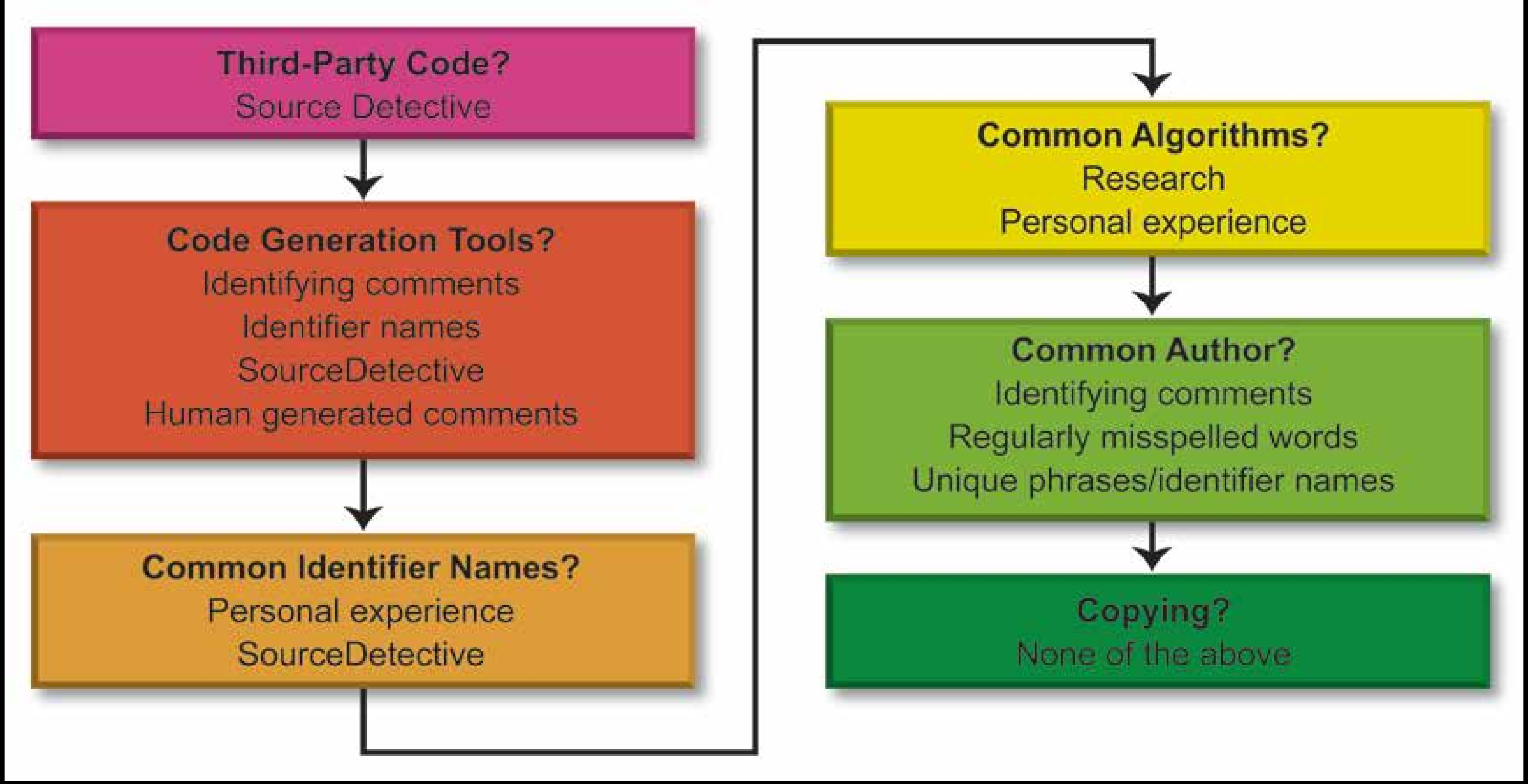
SIGNS OF COPYING?

- CP/M puts a Ctrl-Z character at the end of each file
- DOS does not
- Why is there code to read CP/M files in DOS
- DOS does not read CP/M files
- Research: Early DOS did read CP/M files
- **Not a sign of copying**

COMPARISON OF DOS AND CP/M CODE

- Run CodeMatch®
  - Inspect the most highly correlated file pairs
  - Run SourceDetective®
  - Focus on low search count matches
  - Examine partially matching identifiers
- Run CodeCross®
  - Run SourceDetective
  - Focus on low search count matches

RUNNING CODEMATCH



Third-Party Code?
Source Detective

Code Generation Tools?
Identifying comments
Identifier names
SourceDetective
Human generated comments

Common Identifier Names?
Personal experience
SourceDetective

Common Algorithms?
Research
Personal experience

Common Author?
Identifying comments
Regularly misspelled words
Unique phrases/identifier names

Copying?
None of the above

MATCHING STATEMENTS: ASM

DOS Code

In file PC-DOS\v11source\COMMAND.ASM:

```
FALSE    EQU     0
TRUE     EQU     NOT FALSE
```

In file PC-DOS\v11source\MSDOS.ASM:

```
  IF          IBM
DELIM:
  ENDIF
  CMP         AL,":"        ;Allow ":" as separator in IBM version
  JZ          RET21
  IF          NOT IBM
DELIM:
  ENDIF

  CMP         AL,"+"
  JZ          RET101
  CMP         AL,"="
  JZ          RET101
  CMP         AL,";"
  JZ          RET101
  CMP         AL,","
  JZ          RET101
SPCHK:
  CMP         AL,9          ;Filter out tabs too
  JZ          RET101        ;WARNING! " " MUST be the last compare
  CMP         AL," "
RET101:       RET
```

CP/M Code

In file CPM\1.3\CCP.asm:

```
FALSE   EQU 8000H
TRUE    EQU NOT FALSE
```

In file CPM\1.3\CCP.asm:

```
DELIM:  ;LOOK FOR A DELIMITER
  LDAX D! ORA A! RZ ;NOT THE LAST ELEMENT
    CPI       ' '! JC COMERR            ;NON GRAPHIC
    RZ                                  ;TREAT BLANK AS DELIMITER
    CPI       '=‘ ! RZ
    CPI       LA ! RZ                   ;LEFT ARROW
    CPI       ',‘  ! RZ
    CPI       ',‘  ! RZ
    CPI       ';‘  ! RZ
    CPI       '<‘ ! RZ
    CPI       '>‘ ! RZ
    RET                                 ;DELIMITER  NOT FOUND
```

MATCHING COMMENTS & STRINGS: ASM

DOS

In file PC-DOS\v11source\ASM.ASM:

```
FPREG:
;Have detected "ST" for 8087 floating point stack register
  MOV         DL,0          ;Default is ST(0)
  CALL        SCANB         ;Get next character
  CMP         AL,"("        ;Specifying register number?
  JNZ         HAVREG
;Get register number
  CALL        NEXTCHR       ;Skip over the "("
  CALL        GETOP         ;A little recursion never hurt anybody
  CMP         AL,CONST      ;Better have found a constant
  MOV         CL,20         ;Operand error if not
  JNZ         ERRJ3
  CMP         [DLABEL],0    ;Constant must be defined
  MOV         CL,30
  JNZ         ERRJ3
  MOV         DX,[DATA]     ;Get constant
  CMP         DX,7          ;Constant must be in range 0-7
  MOV         CL,31
  JA          ERRJ3
  MOV         AL,[SYM]
  CMP         AL,")"
  MOV         CL,24
  JNZ         ERRJ3
HAVREG:
  MOV         DH,FREG
  XOR         AL,AL         ;Zero set means register found
  RET
```

CP/M

In file CPM\2.0\cpm20_code\as4sear.asm:

```
NEXTS:        ;LOOK AT NEXT SUFFIX
  LXI         H,ACCUM+1     ;SUFFIX POSITION
  LDAX        D             ;CHARACTER TO ACCUM
  CMP         M
  INX         D             ;READY FOR NEXT CHARACTER
  JNZ         NEXT0         ;JMP IF NO MATCH
  LDAX        D             ;GET NEXT CHARACTER
  INX         H             ;READY FOR COMPARE WITH ACCUM
  CMP         M             ;SAME?
  RZ          ;RETURN WITH ZERO FLAG SET, B IS SUFIX
```

MATCHING IDENTIFIERS: ASM

DOS Code

In file PC-DOS\v11source\COMMAND.ASM:

```
 CRLF:
               MOV   DX,OFFSET RESGROUP:NEWLIN
               PUSH  AX
               MOV   AH,PRINTBUF
               INT   33
               POP   AX
RET10:         RET
```

In file PC-DOS\v11source\COMMAND.ASM:

```
RENAM  EQU  23
```

In file PC-DOS\v20source\PRINT.ASM:

```
;WARNING DANGER WARNING:
;   PRINT is a systems utility. It is clearly understood that it may have
;   to be entirely re-written for future versions of MS-DOS. The following
;   TWO vectors are version specific, they may not exist at all in future
;   versions. If they do exist, they may function differently.
; ANY PROGRAM WHICH IMITATES PRINTS USE OF THESE VECTORS IS ALSO A SYSTEMS
; UTILITY AND IS THEREFORE NOT VERSION PORTABLE IN ANY WAY SHAPE OR FORM.
; YOU HAVE BEEN WARNED, "I DID IT THE SAME WAY PRINT DID" IS NOT AN REASON
; TO EXPECT A PROGRAM TO WORK ON FUTURE VERSIONS OF MS-DOS.
SOFTINT EQU 28H;Software interrupt generated by DOS
COMINT  EQU 2FH;Communications interrupt used by PRINT
               ;  This vector number is DOS reserved. It
               ;  is not generally available to programs
               ;  other than PRINT.

BLKSIZ   EQU 512;Size of the PRINT I/O block in bytes
FCBSIZ   EQU 40 ;Size of an FCB
```

CP/M Code

In file CPM\1.3\CCP.asm:

```
CRLF:          MVI   A, CR! CALL PRINTCHAR
               MVI   A, LF! JMP PRINTCHAR
```

In file CPM\2.0\cpm20_code\os2ccp.asm:

```
renam:         ;rename the file given by d,e
```

In file CPM\2.0\cpm20_code\deblock.asm:

```
;*****************************************************
;*                                                   *
;*         CP/M to host disk constants               *
;*                                                   *
;*****************************************************
blksiz         equ            2048          ;CP/M allocation size
hstsiz         equ            512           ;host disk sector size
hstspt         equ            20            ;host disk sectors/trk
hstblk         equ            hstsiz/128    ;CP/M sects/host buff
cpmspt         equ            hstblk * hstspt;CP/M sectors/track
secmsk         equ            hstblk-1      ;sector mask
               smask          hstblk        ;compute sector mask
secshf         equ            @x            ;log2(hstblk)
```

PARTIALLY MATCHING IDENTIFIERS: ASM

DOS	CP/M	Common
blank blankzer isblank	deblank blank	blank
zexeccodeend zexeccodesize	ccode	ccode
conchng	concha conchar oconch	conch
dollar	pdollar	dollar
extcom	nextcom	extcom
smallddsect	olddsk	ldds
nomod	nomove	nomo
noover	noovf	noov
drvnoset movnamenoset nosetbuf nosetcasc nosetdir nosetsing nosetsing2 nosetudrv nosetver nosetver2 nosetwrperr	noselect	nose

MATCHING INSTRUCTION SEQUENCES: ASM

DOS	CP/M
In file PC-DOS\v11source\IO.ASM:	In file CPM\cpm20_code\os4bios.asm:
`JMP   INIT`	`jmp   const`
`JMP   STATUS`	`jmp   conin`
`JMP   INP`	`jmp   conout`
`JMP   OUTP`	`jmp   list`
`JMP   PRINT`	`jmp   punch`
`JMP   AUXIN`	`jmp   reader`
`JMP   AUXOUT`	`jmp   home`
`JMP   READ`	`jmp   seldsk`
`JMP   WRITE`	`jmp   settrk`
`JMP   DSKCHG`	`jmp   setsec`
`JMP   SETDATE`	`jmp   setdma`
`JMP   SETTIME`	`jmp   read`
`JMP   GETTIME`	`jmp   write`
`JMP   FLUSH`	`jmp   listst      ;list status`
`JMP   MAPDEV`	`jmp   sectran`

MATCHING STATEMENTS: PL/M

DOS

In file PC-DOS\v11source\ASM.ASM:

```
LOAD:
  MOV          DH,25
  CMP          AL,BH          ;Check if memory-to-memory
  JZ           MRERR
  MOV          AL,BH
  CMP          AL,REG         ;Check if 8-bit operation
  JNZ          XRG
  MOV          DH,22
  TEST         CL,1           ;See if 8-bit operation is OK
  JZ           MRERR
```

CP/M

In file CPM\2.0\cpm20_code\load.plm:

```
LOAD:
DO;
  /* C P / M   C O M M A N D   F I L E   L O A D E R

  COPYRIGHT (C) 1976, 1977, 1978
  DIGITAL RESEARCH
  BOX 579 PACIFIC GROVE
  CALIFORNIA 93950

  */

DECLARE
  TPA   LITERALLY '0100H', /* TRANSIENT PROGRAM AREA */
  DFCBA LITERALLY '005CH', /* DEFAULT FILE CONTROL BLOCK */
  DBUFF LITERALLY '0080H'; /* DEFAULT BUFFER ADDRESS */
```

DOS

In file PC-DOS\v11source\ASM.ASM:

```
L0014:
  POP          BX
  MOV          AL,[BX]
  INC          BX
  MOV          CH,AL
  ADD          AL,24
  SHR          AL
  SHR          AL
  SHR          AL
  MOV          CL,AL
  INC          CL             ;Invert last bit
  AND          CL,1           ;Number of extra tabs needed (0 or 1)
  SHR          AL             ;Number of positions wide this symbol needs
  SUB          [SYMLIN],AL
  JNC          WRTSYM         ;Will it fit?
  SUB          AL,SYMWID
  NEG          AL
  MOV          [SYMLIN],AL
  CALL         CRLF           ;Start new line if not
```

CP/M

In file CPM\2.0\cpm20_code\load.plm:

```
PRINT: PROCEDURE(A);
  DECLARE A ADDRESS;
  /* PRINT THE STRING STARTING AT ADDRESS A UNTIL THE
  NEXT DOLLAR SIGN IS ENCOUNTERED WITH PRECEDING CRLF */
  CALL CRLF;
  CALL PRINTM(A);
  END PRINT;
```

MATCHING COMMENTS & STRINGS: PL/M

DOS

In file PC-DOS\v20source\DEBCOM1.ASM:

```
NOHEX:
  CMP         AL,8                          ; Backspace
  JZ          BS
  CMP         AL,7FH                        ; RUBOUT
  JZ          RUB
  CMP         AL,"-"                        ; Back   CLDto previous address
  JZ          PREV
  CMP         AL,13                         ; All done with command?
  JZ           EOL
  CMP         AL," "                        ; Go to next address
  JZ          NEXT
  MOV         AL,8
  CALL        OUT                           ; Back   CLDover illegal character
  CALL        BACKUP
  JCXZ        DWAIT
  JMP         SHORT GETDIG
```

CP/M

In file CPM\1.1\bdos.plm:

```
IF (C := CONIN) = CTLC THEN
  DO; CALL CTLOUT; CALL CRLF;
  GO TO BOOT;
  END;
IF C = CTLE THEN /* PHYSICAL RETURN */
  CALL CRLF; ELSE
IF C = CR THEN
  DO; BUFFER(1) = COMLEN;
  CALL CONOUT(CR);
  RETURN;
  END;
IF C = CTLU THEN
  DO; CALL CTLOUT; CALL CRLF; COMLEN=0;
  END; ELSE
IF C = 7FH THEN /* RUBOUT */
  DO;
  IF COMLEN > 0 THEN
    CALL CONOUT(BUFFER((COMLEN:=COMLEN-1)+2));
  END; ELSE
  DO;
  IF (C AND 01100000B) = 0 THEN /* CONTROL CHARACTER */
    CALL CTLOUT; ELSE
  CALL CONOUT(C);
  BUFFER ((COMLEN:=COMLEN+1)+1) = C;
  END;
END;
```

MATCHING IDENTIFIERS: PL/M

DOS	CP/M
In file PC-DOS\v11source\ASM.ASM:	In file CPM\1.1\bdos.plm:

SETDMA: EQU 26	SETDMA: PROCEDURE(A);
	  DECLARE A ADDRESS;
	  DATAA=(SECTORA:=(TRACKA:=(BUFFA:=A)-3)+1)+1;
	  END SETDMA;

In file PC-DOS\v11source\COMMAND.ASM:	In file CPM\1.3\BDOS.plm:

SETDMA EQU 26	SETDMA: PROCEDURE(A);
	  DECLARE A ADDRESS;
	  CALL SELDMA(BUFFA.= A);
	  END SETDMA;

In file PC-DOS\v11source\HEX2BIN.ASM:	In file CPM\1.3\ED.plm:

SETDMA: EQU 26	SETDMA: PROCEDURE(A);
	  DECLARE A ADDRESS;
	  /* SET DMA ADDRESS */
	  CALL MON1(26,A);
	  END SETDMA;

In file PC-DOS\v11source\MSDOS.ASM:	In file CPM\1.3\PIP.plm:

SETDMA: ;System call 26
  MOV   CS:[DMAADD],DX
  MOV   CS:[DMAADD+2],DS
  RET
	SETDMA: PROCEDURE(A);
	  DECLARE A ADDRESS;
	  CALL MON1(26,A);
	  END SETDMA;

In file PC-DOS\v11source\TRANS.ASM:	In file CPM\1.4\bdos.plm:

SETDMA: EQU 26	SETDMA: PROCEDURE;
	  /* SELECT DATA DMA ADDRESS */
	  IF DIRSET THEN CALL SELDMA(DMAAD);
	  END SETDMA;

In file PC-DOS\v20source\PROFIL.ASM:	In file CPM\2.0\cmp20_code\ed.plm:

SETDMA EQU 26	SETDMA: PROCEDURE(A);
	  DECLARE A ADDRESS;
	  /* SET DMA ADDRESS */
	  CALL MON1(26,A);
	  END SETDMA;

	In file CPM\2.0\cmp20_code\pip.plm:

	SETDMA: PROCEDURE(A);
	  DECLARE A ADDRESS;
	  CALL MON1(26,A);
	  END SETDMA;

	In file CPM\2.0\cmp20_code\stat.plm:

	setdma: procedure(dma);
	  declare dma address;
	  call mon1(26,dma);
	  end setdma;

PARTIALLY MATCHING IDENTIFIERS: PL/M

DOS	CP/M	Common
baddisk baddisklen	ddisk	disk
dmaadd dmaaddr	dmaad	dmaad
needbat	feedbase	eedba
intbase	printbase	intbase
findfile	endfile	ndfile
rloopentry	pipentry	pentry
fcb_random_read fcb_random_read_block fcb_random_write fcb_random_write_block random	read$random readrandom set$random setrandom write$random	random
crename fcb_rename	rename	rename
simped	simplecom simplecopy	simp
args_missing nobatsing nosetsing processing	singlecom singlercom	sing
setabort	tabout	tabo
addr_int_terminate int_terminate	terminate	terminate

SOURCEDETECTIVE RESULTS

Comment or string	Search Score
Save DMA address	51
decrement character count	273
Restore opcode	545
No, get next character	930
DOS entry point	1180

Identifier	Search Score
lstfcb	8
FLGTAB	327
recsiz	878
CHKSIZ	1460
rdloop	2010
setdma	2140
CHKSIZ	1890

Statement	Search Score
JC COMERR	1
JZ GETOP	1
jnz se2	2
CALL NOWRITE	4
JNZ STERR	4
call DISKWRITE	5
JMP SETFCB	5
jmp comerr	11
JNZ COMERR	12
JNZ RDLOOP	12
call SETFCB	14
CALL GETOP	15
TRUE EQU 0FFFFH	90

RUNNING CODECROSS AND SOURCEDETECTIVE

Comment/Statement	Search Score
ENDM	55600
CALL PRINT	167000
endif	1360000
XCHG	1980000
NOP	10100000
DAA	11800000
STC	12400000
CMC	13000000
RET	13500000
ELSE	14500000
NOTE:	53000000
END	253000000

BITMATCH

- Compared MS-DOS 1.0 binary to MS-DOS 1.1 source code
  - High correlation
- Compared MS-DOS 1.0 binary to Q-DOS source code
  - High correlation
- Compared MS-DOS 1.0 binary to CP/M source code
  - Low correlation

COPIED CODE?

NO

COMPARISON OF COMMAND LINE INTERFACES

DOS	CP/M	VMS	Apple DOS
	ASSIGN		
			CLOSE
COPY		COPY	
DATE			
DEL		DELETE	DELETE
DIR	DIRECT	DIR	CATALOG
ERASE	ERASE		
			INIT
			LOCK
			OPEN
			READ
PAUSE			
REM			
RENAME	RENAME	RENAME	RENAME
	SAVE		WRITE
TIME			
TYPE	TYPE	TYPE	
			UNLOCK
			VERIFY

COPIED COMMANDS?

NO

COMPARISON OF SYSTEM CALLS

DOS

In file PC-DOS\v11source\MSDOS.ASM:

```
; Standard Functions
DISPATCH DW     ABORT   ;0
         DW     CONIN
         DW     CONOUT
         DW     READER
         DW     PUNCH
         DW     LIST     ;5
         DW     RAWIO
         DW     RAWINP
         DW     IN
         DW     PRTBUF
         DW     BUFIN    ;10
         DW     CONSTAT
         DW     FLUSHKB
         DW     DSKRESET
         DW     SELDSK
         DW     OPEN    ;15
         DW     CLOSE
         DW     SRCHFRST
         DW     SRCHNXT
         DW     DELETE
         DW     SEQRD   ;20
         DW     SEQWRT
         DW     CREATE
         DW     RENAME
         DW     INUSE
         DW     GETDRV  ;25
         DW     SETDMA
         DW     GETFATPT
         DW     GETFATPTDL
         DW     GETRDONLY
         DW     SETATTRIB  ;30
         DW     GETDSKPT
         DW     USERCODE
         DW     RNDRD
         DW     RNDWRT
         DW     FILESIZE   ;35
         DW     SETRNDREC
; Extended Functions
         DW     SETVECT
         DW     NEWBASE
         DW     BLKRD
        DW     BLKWRT     ;40
        DW     MAKEFCB
        DW     GETDATE
        DW     SETDATE
        DW     GETTIME
        DW     SETTIME    ;45
        DW     VERIFY
```

CP/M

In file CPM\1.1\bdos.plm:

```
  DO CASE FUNC;
  GO TO BOOT;                              /* 0: SYSTEM RE-BOOT */
  DO; RET = CONIN; CALL CONOUTA(RET);      /* 1: READ CONSOLE */
    END;
  CALL CONOUT(LINFO);                      /* 2: WRITE CONSOLE */
  RET = OCTIN;                             /* 3: READ OCTOPUS (INFO=0), OR RETURN STATUS
(INFO=1,2) */
  CALL OCTOUT(LINFO);                      /* 4: WRITE OCTOPUS */
  CALL LISTOUT(LINFO);                     /* 5: WRITE LIST DEVICE */
  ARET = 2900H;                            /* 6: INTERROGATE MEMORY SIZE */
  ARET = IOSTAT;                           /* 7: INTERROGATE DEVICE STATUS */
  IOSTAT = INFO;                           /* 8: CHANGE DEVICE STATUS */
  CALL PRINT(INFO);                        /* 9: PRINT BUFFER AT THE CONSOLE  */
  CALL READ;                               /* 10: READ BUFFER FROM THE CONSOLE */
  RET = CONBRK;                            /* 11: CHECK FOR CONSOLE INPUT READY */
  ;                                        /* 12: */
    DO; CURDSK,DLOG = 0;                   /* 13: RESET DISK SYSTEM, INITIALIZE TO DISK 0 */
    CALL SETDMA(80H);
    CALL SELECT;
    END;
    DO; CURDSK = LINFO;                    /* 14: SELECT DISK 'INFO' */
    CALL SELECT;
    END;
  CALL OPEN;                               /* 15: OPEN */
  CALL CLOSE;                              /* 16: CLOSE */
  CALL SEARCH(FNM);                        /* 17: SEARCH FOR FIRST OCCURRENCE OF A FILE */
  CALL SEARCHN;                            /* 18: SEARCH FOR NEXT OCCURRENCE OF A FILE NAME */
  CALL DELETE;                             /* 19: DELETE A FILE */
  CALL DISKREAD;                           /* 20: READ A FILE */
  CALL DISKWRITE;                          /* 21: WRITE A FILE */
  CALL MAKE;                               /* 22: CREATE A FILE */
  CALL RENAME;                             /* 23: RENAME A FILE */
  RET = DLOG;                              /* 24: RETURN THE LOGIN VECTOR */
  RET = CURDSK;                            /* 25: RETURN SELECTED DISK NUMBER */
  CALL SETDMA(INFO);                       /* 26: SET THE SUBSEQUENT DMA ADDRESS TO INFO */
  ARET = ALLOCA;                           /* 27: RETURN THE LOGIN VECTOR ADDRESS  */
  ;
                                           /* 28: UNUSED */
  ;
                                           /* 29: UNUSED */
  ECHO = LINFO;                            /* 30: ECHO CALL NO. 1 IF ARGUMENT IS TRUE */
  END; /* OF CASES */
```

COPIED SYSTEM CALLS?

YES

COPYRIGHT INFRINGEMENT?

- I’m not a lawyer
- I’ve been an expert in over 175 IP cases
- I’ve written about IP
- A list of numbers is not by itself creative and thus not copyrightable
- But a list of numbers that arbitrarily represents specific functions is creative and thus copyrightable
- Furthermore, DRI appears to have indicated its copyright by putting a copyright notice on the CP/M Interface Guide
- Did DRI guard its system calls from copying?

COPYRIGHT INFRINGEMENT?

POSSIBLY

FAIR USE BY MICROSOFT?

- The purpose and character of the use, including whether such use is for nonprofit educational purposes
  - **No**
- The nature of the copyrighted work, especially whether it benefits the public
  - **No**
- Small amount or substantiality of the portion used in relation to the copyrighted work as a whole
  - **Probably not**
- Small effect of the use upon the potential market for or value of the copyrighted work
  - **Probably**

FAIR USE BY MICROSOFT?

PROBABLY

CONCLUSION: WHAT WAS COPIED?

Code was not copied

Command line interface was not copied

System calls were substantially copied

There is no secret routine in CP/M or DOS

CONCLUSION: WAS THERE COPYRIGHT INFRINGEMENT?

Code: no

Command line interface: no

System calls: maybe

WHAT ABOUT DOS 1.0?

- Does the source code exist?
- If so, I need to get a copy
- Since Q-DOS and DOS 1.1 are not copies of CP/M, and the DOS 1.0 binary is not highly correlated with CP/M, it is extremely unlikely that DOS 1.0 is copied from CP/M.

ZEIDMAN CHALLENGE AWARD 1

$100,000

Show that DOS code was copied from CP/M

Details to be announced soon

ZEIDMAN CHALLENGE AWARD 2

$100,000

Find secret routine in DOS that prints DRI or Kildall

Details to be announced soon

Q & A

CONTACT INFO

Bob Zeidman

Zeidman Consulting

www.ZeidmanConsulting.com

Bob@ZeidmanConsulting.com

Download results from www.ZeidmanConsulting.com/DOS_comparisons

ATTRIBUTIONS

- Photo of Bill Gates, courtesy of Shannon Ramos
- Photo of IBM Headquarters, courtesy of Simon Greig
- Photo of Gary Kildall, © Tom Munnecke/Hulton Archive/Getty Images
- Photo of Big Brother, courtesy of Apple Computer Corp.
- Photo of Bill Gates, courtesy of 60 Minutes
- Photo of Gazelle computer, courtesy of Seattle Computer Products
- Photo of Q-DOS manual, courtesy of The Register
- Photo of DOS manual, courtesy of Michael Holley