Journal of Software Engineering and Applications, 2014, 7, 513-529
Published Online May 2014 in SciRes. http://www.scirp.org/journal/jsea
http://dx.doi.org/10.4236/jsea.2014.76048

Scientific Research

A Code Correlation Comparison of the DOS and CP/M Operating Systems

Robert Zeidman

Zeidman Consulting, Cupertino, USA
Email: Bob@ZeidmanConsulting.com

Received 3 April 2014; revised 1 May 2014; accepted 8 May 2014

Abstract

For years, rumors have circulated that the code for the original DOS operating system created by Microsoft for the IBM personal computer is actually copied from the CP/M operating system developed by Digital Research Incorporated. In this paper, scientifically tested and accepted forensic analysis mathematical techniques, step-by-step processes, and advanced software code comparison tools are used to compare early versions of the two code bases. The conclusion is reached that no copying of code takes place[1].

Keywords

DOS, CP/M, Copyright Infringement, Software Forensics, Software Correlation

1. Introduction

For purposes of better understanding, the introduction includes the historical background and the legal issues.

1.1. Historical Background

Gary Kildall is the man who, according to some, could have been and should have been the reigning king of software. Kildall created the CP/M (Control Program for Microcomputers) operating system that was used on many of the hobbyist personal computers before Apple and IBM introduced their machines. Kildall created

[1]Full disclosure: The process used is the process developed at my consulting company Zeidman Consulting. The tools used are the tools produced by my software company Software Analysis and Forensic Engineering. I have worked as an expert witness in intellectual property cases both for and against Microsoft, and until recently I was engaged as an expert for Microsoft in the case of *Motorola Mobility, Inc. v. Microsoft Corporation*, case 2:2011cv01408 in the Washington Western District Court. The initial results of this paper were summarized and published online in the July 2012 *IEEE Spectrum* magazine article "Did Bill Gates Steal the Heart of DOS?" (http://spectrum.ieee.org/computing/software/did-bill-gates-steal-the-heart-of-dos). This paper expands on those results by giving the underlying details and examining additional versions of MS-DOS.

How to cite this paper: Zeidman, R. (2014) A Code Correlation Comparison of the DOS and CP/M Operating Systems. *Journal of Software Engineering and Applications*, **7**, 513-529. http://dx.doi.org/10.4236/jsea.2014.76048

CP/M while working for Intel and in 1974 started Digital Research Inc. (DRI) to sell it.

IBM saw the potential for the microcomputer and in 1980 started a "skunk works" project in Boca Raton, Florida to create the IBM PC, released in 1981. This group was given the unique task of creating a machine not for global corporations and government agencies but for small businesses and individuals. They decided that rather than develop software in-house, as was typical at IBM, they would partner with one of the small companies already producing software for microcomputers. Their first stop, in 1980, was a small company in Bellevue, Washington called Microsoft that sold a successful version of the BASIC programming language for microcomputers. There, the young CEO Bill Gates told IBM that they should contact Gary Kildall at Digital Research Inc. (DRI) in Pacific Grove, California for the CP/M (Control Program for Microcomputers) operating system that was used on many hobbyist personal computers. Kildall had created CP/M while working for Intel and in 1974 started DRI to sell it. Kildall is the man who, according to some, could have been and should have been the reigning king of software [1]-[5].

But here is where the story varies, depending on who is telling it. In one version, the IBM executives flew down to meet Kildall who, as a member of the personal computer counterculture, did not trust "Big Brother" and so he took off in his plane for a joyride [1]. When the IBM execs showed up, they were met by Kildall's wife and business partner Dorothy who refused to sign IBM's standard non-disclosure agreement (NDA). After several hours of haggling over the NDA, the IBM executives got frustrated and left [1] [3]-[5].

In another version of the story, Kildall and DRI employee Tom Rolander went off in the plane to deliver software to a customer and left the license negotiations with Dorothy who normally handled those matters [3]. Dorothy felt the NDA was too restrictive and talked to their attorney Gerry Davis who advised her to wait for Kildall to return [1] [3]. Kildall returned later that day but accounts again differ as to whether he signed the NDA or even participated in discussions with IBM [1].

It is a fact that no deal was signed. Kildall later said that he met IBM negotiator Jack Sams on a flight to Florida that evening, negotiated a deal on the flight, and shook hands on it. Sams denied ever meeting Kildall [1] [3]. In fact, the IBM negotiators flew to Seattle that day, not Florida, and met again with Bill Gates. Gates knew of a similar microcomputer operating system, QDOS (later renamed 86-DOS) from nearby Seattle Computer Products (SCP) that sold microcomputer boards. Because DRI was late getting out its operating system for the new Intel 8086 processor, SCP had hired programmer Tim Paterson to write its own operating system called QDOS for "Quick and Dirty Operating System." Gates quickly acquired the rights to it for $75,000 [6]-[8] and hired Paterson to modify it into MS-DOS for licensing to IBM. SCP owner Rod Brock got what he wanted in the deal—the ability to bundle SCP's hardware with Microsoft's operating system and programming languages—resulting in more than $1 million in profits on record revenue of about $4 million in sales the next year [3].

The IBM PC became a huge success and DOS displaced CP/M as the leading microcomputer operating system; Kildall eventually negotiated a deal with IBM to offer CP/M on the PC. However, Kildall negotiated a very high license fee—much higher than MS-DOS—meaning IBM had to charge $240 per copy of CP/M rather than the $40 per copy it charged for MS-DOS [1]. Few people bought CP/M, and MS-DOS sales continued to grow.

Gary Kildall maintained that QDOS, and subsequently MS-DOS, had been directly copied from CP/M and thus infringed on his copyright [1] [9]. DRI attorney Gerry Davis claimed that forensic experts had proven that MS-DOS had been copied from CP/M but that in 1981 there was no way to go to court over copyright infringement and get a judgment [1]. This was not true, as explained in the next section.

1.2. Legal Background

A copyright is a form of intellectual property protection for the expression of an idea. The World Intellectual Property Organization (WIPO) defines a copyright as "a legal term describing rights given to creators for their literary and artistic works (including computer software)" [10]. According to the US Copyright Office, "copyright is a form of protection provided by the laws of the United States (title 17, US Code) to the authors of 'original works of authorship⋯' [and] is available to both published and unpublished works⋯" The copyright owner has the exclusive right to reproduce the work, prepare derivative works, distribute copies, perform the work publicly, display the work publicly, or to authorize others to do so [11].

From its beginnings, copyright has protected creative text, and software source code is inarguably creative text. This was tested when the first computer program was submitted for copyright registration, the SCOPAC-PROG.63 program from North American Aviation, on November 30, 1961 in the form of a magnetic tape. While

the Copyright Office was trying to determine how such a deposit could be registered, two short computer programs were submitted on April 20, 1964 by Columbia Law student John Francis Banzhaf III [12]. The copyrights for both student computer programs were registered in May 1964, and North American Aviation's computer program was registered in June 1964. The Computer Software Copyright Act of 1980 formalized the submission requirements for software, and the number of software source code copyright registrations exploded shortly thereafter [12] [13]. Note that a copyright exists, and the owner is entitled to all copyright protections, whether it is registered with the US Copyright office or not.

2. Code Comparisons

Was Bill Gates' fortune was built on infringement and deception[2]? The CodeSuite®software forensic tools were used to perform a code comparison. These are the only tools that have been accepted in US courts and that has been used in over 60 software copyright cases. CodeSuite uses scientifically accepted algorithms for detecting software copying [14]-[18], has been compared to other "software plagiarism detection[3]" algorithms [19]-[26]. A standard, accepted forensic analysis process was also used to filter the results [17] [27]. The CodeMatch® function of CodeSuite compares source code of different programs to find instances of copying. It narrows down areas in different source code files that are correlated. There are six reasons that code can be correlated:

- **Third-Party Source Code**. It is possible that widely available open source code or third-party libraries are used in both programs.
- **Code Generation Tools**. Automatic code generation tools generate software source code using similar lines of code.
- **Commonly Used Identifier Names**. Certain identifier names are commonly taught in schools or commonly used by programmers in certain industries.
- **Common Algorithms**. There may be an easy or well-understood way of writing a particular algorithm that most programmers use.
- **Common Author**. Two programs written by the same programmer will have style similarities.
- **Copying**. Code was copied from one program to another causing the programs to have similarity.

The website *The Unofficial CP/M Web site* has links to download CP/M source code files that include notices of copyright by Gary Kildall from 1975, shortly after he founded DRI [28]. They are written in the PL/M programming language that Kildall developed while he was employed at Intel. The source code for CP/M 2.0 from 1981 was also downloaded from the same site, but that code contains copyright notices from 1976, 1977, and 1978.These files are written in both PL/M and low-level assembly code. The executable binary files of CP/M 1.4 were also downloaded from the site and three source code files dated from March 22, 1979 through September 5, 1981.

The website *Howard's Seattle Computer Products SCP 86-DOS Resource Website* contains 86-DOS (QDOS) assembly language source code files and executable binary files with revision dates as late as April 28, 1981 that were also downloaded [29].

A functional MS-DOS 1.11 floppy disk for a Compaq computer was obtained, one of the earliest versions of DOS from Microsoft. The files on the disk are executable binary files.

2.1. Comparing CP/M Source to QDOS Source

The QDOS source code was compared to the CP/M source code to see if there was any evidence that QDOS was copied from or was a derivative of CP/M. This had to be done in two steps because the CP/M source code included files written in the PL/M programming language and files written in assembly language. Copying code from a high level language like PL/M to low-level assembly language is unlikely because the languages are so different, but the comparison was performed anyway for the sake of completeness.

There was some correlation of programming statements in the two programs, but these matching statements look like fairly common, simple statements. One statement that correlated between the two programs, for exam-

[2]The detailed results are too extensive to be included in their entirety in this paper. Instead, the code, the code comparison results, and string extractions can be downloaded in a zip file at http://www.ZeidmanConsulting.com/DOS_comparisons.

[3]I do not use the common term "software plagiarism detection" because plagiarism means copying without authorization. No software analysis algorithm can determine whether copying has been authorized, because that depends on issues that are not evident in the code itself, such as legal contracts and jurisdictional law.

ple, was

```
CALL CRLF
```

The identifier CRLF is a common abbreviation for the carriage-return/line-feed character pair at the end of every line in a text file in these operating systems. The statement `CALL` means that a procedure is being called, and in both cases these procedures simply terminate a line of text with a carriage return and a line feed. As of this writing, the term `CALL CRLF` occurs 22,300 times on the Internet according to Google, and most of these refer to a routine that writes a carriage return/linefeed. So the occurrence of this term in both programs can simply be attributed to common algorithms and common identifier names. A full list of statements found in both programs is given in **Table 1**.

Other correlation was due to identifiers with common names like `MAKE` or `BOOT` or numbers like `10`, `255`, or `5CH` (hex) that can be found in many programs. A full list of identifiers found in both programs is given in **Table 1**.

A little bit of correlation was due to matching comments and strings, but these comments and strings, such as `SECTORS PER TRACK`, are common operating system terms and messages that can be found in many programs. A full list of comments and strings found in both programs is given in **Table 1**.

There were no significant sequences of instructions that matched between the two programs, which would show similar, possibly copied functionality. The only matching sequences consisted of multiple `JMP` statements, which are commonly called “jump tables” and are a common programming technique. There were sequences of DB and DW statements, also common programming language techniques, that simply define data in the program, but the data values did not match.

Most of the correlation was due to partially matching identifiers, where only part of the identifier names are identical. This can be a clue to copying where a programmer changed the names enough to appear different but still retain some meaning. For example the variable name `FirstName` might be changed to `Fname`. Examination of these elements shows them to be commonly used identifier names or random characters. For example, the identifier `ENDMOD` in the CP/M source code partially matched the identifiers `MOD5` and `MOD6` in the QDOS source code.

The process for filtering out correlation due to reasons other than copying uses the SourceDetective® function of CodeSuite to search the Internet for other references to matching program elements. The entire filtering process is shown in **Figure 1**. If an element is found in two programs and is also found many times on the In

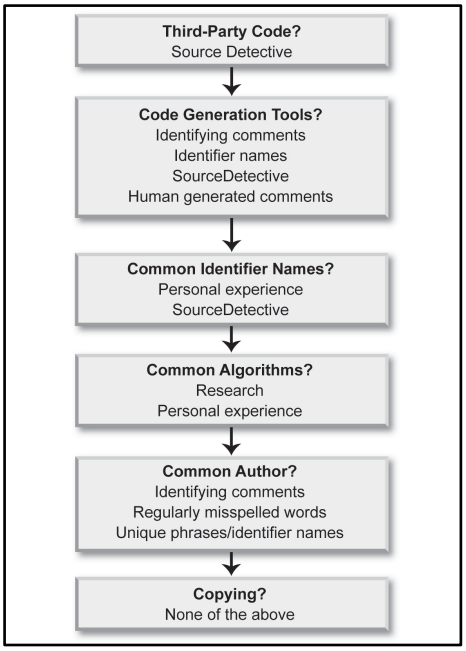

Figure 1. Filtering process to find copying.

Table 1. Matching program elements found in CP/M and QDOS source code.

Statements				
backsp:	backup:	BLANK:	boot:	call backup
CALL BLANK	CALL CRLF	CALL HEX	CALL HOME	CALL MAKE
CALL SEEK	CALL SETUP	CMP B	comerr:	CRLF:
DB 0	db 1	db 2	DW FILL	DW MOVE
DW TRACE	EI	FILL:	GETFLG:	HEX:
home:	IF OTHER	INIT:	JMP BOOT	JMP INIT
JMP PRINT	JMP READ	JMP WRITE	jnz comerr	JNZ STEP1
JZ BS	MAKE:	MOVE:	ORG 0	ORG 100H
OUT 0	PERR:	PRINT:	READ:	RETRY:
SCAN:	search:	SEC:	seek:	select:
SETUP:	write:			
Identifiers				
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24
25	26	27	30	31
32	33	35	39	40
50	51	63	64	80
128	255	256	1024	01H
02H	04H	08H	0A0H	0A8H
0BH	0C0H	0C4H	0CEH	0CH
0D0H	0DH	0E5H	0EH	0F0H
0F2H	0F3H	0F6H	0FBH	0fch
0FDH	0FEH	0FF80H	0FFH	0FH
100H	10H	14H	17H	18H
19H	200H	20H	21H	22H
32H	37H	38H	40H	4H
50H	5CH	5FH	72H	78h
7FH	800H	80H	84H	88H
8H	90H	9H	BACKSP	BACKUP
BADCOM	base	BLANK	BOOT	BS
COMERR	CRLF	DCOM	DIGIT	DIRECTION
DISK	DM	DONE	EI	FCB
FERR	FILL	FLAG	GETFLG	HEX
HOME	INIT	INP	INPUT	LOAD
MAKE	MOVE	NEXT	NOHEX	OUTPUT
OTHER	PERR	PRINT	RD	RDBYTE
READ	READCOM	RESTORE	RETRY	RLOOP

Continued

SCAN	SEARCH	SEC	SECSIZ	SECT
SECTOR	SEEK	SEL	SELECT	SERIAL
SETUP	STEP	STEP1	stat	status
TAB	TRACE	TRACK	UP	WRITE

Comments and Strings

BACKSPACE	BLOCK MASK	DECREMENT SECTOR COUNT.	error	EXTENT MASK
GET COMMAND LINE	INCREMENT SECTOR NUMBER.	LENGTH	MULTIPLY BY 16	PRINT IT
RETURN IF NOT.	RETURN.	RUBOUT	SAVE	SAVE COUNT
SAVE FOR LATER	SAVE LENGTH	SECTORS PER TRACK	SET DMA ADDRESS	

ternet, it is likely a commonly used term. If it is found in two programs but nowhere else on the Internet, then it is likely due to copying.

Normally all matching elements that had any hits on the Internet would be filtered out. In this case, in order to be a little more liberal, only filter out matching elements that were found more than 100 times on the Internet. In this way, even things that were found in other programs or documents on the Internet would not be filtered out.

After filtering, no identifiers remained, but one programming statement and two comments did remain[4].

2.1.1. Common Statement

The statement that remained after filtering was:

```
jnz comerr
```

This programming statement was found in only one place on the Internet. The instruction jnz is a standard program assembly language statement for “jump if not zero.” The comerr is a label in both programs that specifies the beginning of some routine. This looks like a combination of com, which could refer to a communications port or a command, and err, which typically means an error. One educated guess was that comerr is a routine that handles either communication errors or command errors, but it was necessary to look at the actual code routines. The QDOS comerr routine is shown in **Listing 1** while the CP/M comerr routine is shown in **Listing 2**. These are significantly different routines. The QDOS routine gets invoked when there is a problem reading a file. The CP/M routine is more complex and gets invoked when there is a problem with a command. These routines have no relationship to each other and do not signify copying.

The place on the Internet where comerr was found turned out to be a document that included snippets of MS-DOS source code [30].

2.1.2. Common Comments

Two comments remained after filtering. They were:

```
INCREMENT SECTOR NUMBER.
DECREMENT SECTOR COUNT.
```

By themselves, the comments are not uncommon, but two things struck me as particularly suspicious. First, both comments ended with a period. Some programmers have a programming style where they end their comments with period, so these matching comments could be a sign of a common programmer, but these two programs were supposedly not written by the same programmer.

Both files `IO4IOS32.ASM` and `IO4IOS64.ASM` of the CP/M 1.4 code had the same routine called `RDBLK1` where these comments were found, shown in **Listing 3**.

[4]Source Detective uses an API to search via Yahoo! The API does not report as many hits as entering the information via a web browser. Also the Google search engine covers more web pages. Thus some elements were given low hit numbers but when manually searching, the numbers were much greater.

```
COMERR:
    MOV DX,BADCOM
    MOV AH,9        ;Print string
    INT 21H
    EI
STALL:  JP  STALL
```

Listing 1. Comerr routine in file DOSIO.ASM from QDOS.

```
comerr:
    ;error in command string starting at position
    ;'staddr' and ending with first delimiter
    call crlf ;space to next line
    lhld staddr ;h,l address first to print
comerr0: ;print characters until blank or zero
    mov a,m! cpi ' '! jz comerr1; not blank
    ora a! jz comerr1; not zero, so print it
    push h! call printchar! pop h! inx h
    jmp comerr0; for another character
comerr1: ;print question mark,and delete sub file
    mvi a,'?'! call printchar
    call crlf! call del$sub
    jmp ccp ;restart with next command
```

Listing 2. Comerr routine in file os2ccp.asm in CP/M.

The file `DOSIO.ASM` of `QDOS` has a routine called `NEXTSECTOR`, shown in **Listing 4**.

Both of these routines handle disk access, thus the reference to disk sectors, but other than the two comments there are similarities but appear to be very different code. Furthermore, the CP/M file header comments refer to a company called Tarbell Electronics and imply that the code was developed by that company:

```
;( TARBELL ELECTRONICS) CVE MOD OF 9-5-81
```

Similarly the QDOS code header comment states that the code was developed for compatibility with several disk drive manufacturers including Tarbell Electronics:

```
; Assumes a CPU Support card at F0 hex for character I/O,
; with disk drivers for Tarbell, Cromemco, or North Star controllers.
```

Searching for Tarbell Electronics it was discovered that this company produced and sold floppy drives starting in the 1970s [31]. On the web page for Harte Technologies was found driver code that Tarbell originally supplied with its floppy drives [32]. In the code for CP/M, there are three files called `ABIOS24.ASM`, `2ABIOS24.ASM`, and `2ABIOS64.ASM` with a routine called `RBLK1` shown in **Listing 5**.

This Tarbell code also has a copyright notice at the top:

```
;-----------------------------------------------
; CP/M BASIC INPUT/OUTPUT OPERATING SYSTEM (BIOS)
; TARBELL ELECTRONICS
; 2.X VERSION OF 11-4-80
; Copyright (c) 1980 Tarbell Electronics
;-----------------------------------------------
```

While a copyright notice is not proof of copyright, hardware developers generally write drivers and then distribute them to enable use of their hardware. The Tarbell RDBLK1 routine is very similar to the one in the CP/M code and the implication is that CP/M and QDOS both relied on the Tarbell driver code.

```
RBLK1:  SHLD DMAADD ;SET STARTING ADDRESS.
    CALL SETSEC ;READ STARTING AT SECTOR IN C.
    CALL READ
    JNZ  RDERR  ;IF ERROR, PRINT MESSAGE.
    DCR  D      ;DECREMENT SECTOR COUNT.
    JZ   ALDON  ;ALL DONE WHEN D=0.
    INR  C      ;INCREMENT SECTOR NUMBER.
    MOV  A,C;IF SECTOR NUMBER
    CPI  27     ;IS NOT 27,
    JC   RBLK1  ;KEEP READING ON THIS TRACK.
    MVI  C,1;OTHERWISE, RESET SECTOR=1,
    INR  B      ;INCREMENT TRACK NUMBER,
    JMP  RDBLK  ;AND READ NEXT TRACK.
ALDON:  LDA  TEMP   ;RESTORE DISK NUMBER.
```

Listing 3. CP/M routine RDBLK1.

```
NEXTSECTOR:
    EI          ; Interrupts OK now.
    POP CX      ; Get sector count.
    DEC CL      ; Decrement sector count.
    JZ  OKRETURN    ; Return if done.
    INC CH      ; Increment sector number.
    CMP CH,10   ; Compare with number of sectors on track.
    JAE NEEDSTEP
    JMP SECTORLOOP  ; Read another sector from same track.
```

Listing 4. QDOS routine NEXTSECTOR.

```
RBLK1:  SHLD DMAADD     ;SET STARTING ADDRESS.
    CALL SETSEC  ;READ STARTING AT SECTOR IN C.
    PUSH B
    CALL READ   ;READ A SECTOR BACK.
    POP  B
    JNZ  RDERR  ;IF ERROR, PRINT MESSAGE.
    INR  C      ;INCREMENT SECTOR NUMBER.
    DCR  B      ;DECREMENT SECTOR COUNT.
    JNZ  RBLK1  ;NOT ZERO, KEEP READING
;
    IF INTRP;   IF INTERRUPTS ALLOWED,
    EI          ;ALLOW THEM AGAIN HERE.
    ENDIF
```

Listing 5. Tarbell routine RDBLK1.

Other than these examples, that could be explained by reasons other than copying, filtering out matching elements that were found more than 100 times on the Internet eliminated all matching elements, leaving no correlation at all.

2.2. Comparing MS-DOS Binary to CP/M Source

It was not possible to locate any source code for MS-DOS, which is understandable since it is a commercial product from an ongoing company. A floppy disk was located that contained MS-DOS 1.11 for the first Compaq computer. CodeSuite has a tool called BitMatch® that compares binary code to source code or to other binary code. The MS-DOS 1.11 binary code was compared to the CP/M source code files from all of the versions of CP/M that were obtained. Comparing binary files has a possibility of false negatives. If correlation is found after filtering, then the files were almost certainly copied., but if nothing is found, the results are inconclusive.

BitMatch does not compare programming statements because binary code instructions are very dependent on the tools used to compile the code. BitMatch does compare sequences of text characters in the binary, which it assumes are either identifiers or strings. Correlation was found due to 95 matching identifiers in both programs, which are listed in Table 2. With a few exceptions, these identifiers all are common words from operating systems and programming or just from the English language.

Correlation was found due to 11 matching comments and strings, which are listed in Table 2. These comments and strings are all common words or phrases from operating systems and programming. Filtering out matching elements that were found more than 100 times on the Internet eliminated all matching elements, leaving no correlation at all.

2.3. Comparing MS-DOS Binary to CP/M Binary

Next the MS-DOS 1.11 binary was compared to all of the binary files of the different versions of CP/M. There were 74 matching strings, shown in Table 3. These strings are also common words or phrases from operating systems and programming.

Table 2. Matching identifiers and strings found in MS-DOS binary code and CP/M source code.

Identifiers			
0	1100	1101	ALT
base	BEGIN	Bit	BLOCK
BOOT	BREAK	BUFFER	compare
CONT	copied	copy	Copying
COPYRIGHT	DEL	different	DIR
DIRECT	DISK	disks	DISPLAY
empty	EOF	ERASE	ERROR
ESC	FILE	find	first
FOUND	HEX	HIGH	INIT
INPUT	Insert	INT	JMP
key	LENGTH	LETTER	LOAD
MEMORY	mode	MODULE	NEXT
normally	NUMBER	NUMERIC	OBJ
one	OPEN	OUT	OVERFLOW
PAGE	per	read	reading
RENAME	RESERVED	RETRY	ROF
ROR	save	SCRATCH	SCREEN
SCRN	SECTOR	Seek	SELECT
set	SIN	source	STACK
START	status	table	Terminate
testing	THE	There	time
TIMEOUT	Track	transfer	TYPE
USER	VALUE	VERSION	VIDEO
Which	write	ZERO	
Strings			
com	DIR	ERASE	File not found
JMP	OUT	POP	PUSH
REN	RENAME	TYPE	

Table 3. Matching strings found in MS-DOS binary code and CP/M binary code.

Strings			
$File	(C)	(Y/N)	(Y/N)?
<2T	aborting	ALL	BAD
CANNOT	CHARACTER	COM	COMMAND
COMPLETE	COPY	COPYRIGHT	DATA
DESTINATION	DETECTED.	DIAGNOSTIC	DIR
DIRECTORY	DISK	drive	END
ERASE	ERROR	ERRORS	EXISTS
EXIT	FILE	FILE$	FILES
FOR	FOUND	found$Write	FULL
FUNCTION	HAS	INPUT	INVALID
MANY	MEMORY	MISSING	NAME
NEXT	NOT	OUT	OVERFLOW
PAUSE	POP	PRN	PROGRAM
read	ready	RETURN	SECTOR
SELECT	SOURCE	SPACE	START
SYMBOL	SYNTAX	TABLE	TEST
THE	THEN	TOO	Track
TYPE	VERSION	when	WITH
WRITE	Yq:		

Filtering out matching elements that were found more than 100 times on the Internet again results in no matching elements and no correlation at all.

3. Comparing MS-DOS to QDOS

As a baseline, the MS-DOS binary code was compared to both the QDOS source code and binary code. Since MS-DOS was derived from QDOS, there should be significant correlation. As expected, there is significant correlation between the two programs. Before filtering 252 strings were found in common between MS-DOS 1.11 and QDOS. Some of the matches are sequences of random characters that obviously match coincidentally, but after filtering out all things that can be found at least once on the Internet, some of the more interesting and conclusive similarities are:

```
AXBXCXDXSPBPSIDIDSESSSCSIPPC
NVUPDI
$No room in disk directory
QWASRDLIE
WVULLRQSP
$O.K.? $Line too long
```

The first sequence of alphabetic characters is particularly telling. Searching manually for this sequence on the Internet produces only 3 instances of that string, all of which seem to be snippets of QDOS code[5]. These identifiers that can be found in QDOS and MS-DOS and nowhere else on the Internet constitute conclusive confirmation that MS-DOS was derived from QDOS. This string actually combines the two-letter names of the registers inside the Intel 8086 processor [33], (except for the last name, PC, which is not an internal register):

[5]Search engine APIs that allow a program to automatically perform a search, like those used by Source Detective, often have slightly different results than a manual search via a web browser. Also, different search engines give slightly different results.

```
AX: Accumulator
BX: Base
CX: Count
DX: Data
SP: Stack Pointer
BP: Base Pointer
SI: Source Index
DI: Destination Index
DS: Data Segment
ES: Extra Segment
SS: Stack Segment
CS: Code Segment
IP: Instruction Pointer
```

4. Kildall's Hidden Message

According to science fiction writer and technology reporter Jerry Pournelle, there was a secret command in DOS that printed a copyright notice for DRI and Kildall's full name to the screen [34]. According to Pournelle, Kildall had told him about this command and typed it into DOS whereupon it produced the notice and allegedly proved that DOS source code was copied from CP/M source code. This story has several problems with it. First, no one knows the secret command. Second, Pournelle claims he wrote the command down, but will not show it to anyone. Third, such a message would be easily seen by opening the binary files in a simple text editor unless the message was encrypted. In the book *They Made America*, Kildall is quoted from his memoir as saying that he encrypted messages in CP/M to find copying [35], but remember that CP/M and DOS had to fit on a floppy disk that held only 160 Kbytes. Kildall's achievement was that he could squeeze an entire operating system into such a small footprint; it is difficult to imagine he could also squeeze an undetectable encryption routine.

If the message were unencrypted in the source code, the programmers who had the source code that they allegedly stole from DRI would see this extraneous routine and immediately remove it. A utility program was used to extract strings of text from binary files. Searching these strings, not only does Kildall's name not show up in QDOS or MS-DOS, it does not show up in CP/M either. The term "Digital Research" shows up in copyright notices in the CP/M binary files, but not in MS-DOS or QDOS binary files.

One could argue that Kildall used a simple masking algorithm rather than a full blown encryption algorithm to hide the message, but the masked data would most likely show up in both the CP/M and QDOS as seeming random text. Nothing like this was found. Also, CP/M defines intrinsic commands and extrinsic commands. Intrinsic commands are the basic commands that are buried in the code while the extrinsic commands are names of executable files. For example, the extrinsic CP/M command `DISKTEST` is in the file `DISKTEST.COM`. For a command to be hidden, it must be buried inside the files and therefore must be an intrinsic command. There are six documented intrinsic commands: `ERA`, `DIR`, `REN`, `SAVE`, and `TYPE`. All of these commands are parsed and executed in the CP/M source code file `os2ccp.asm` at lines 390 through 397:

```
intvec:
   ;intrinsic function names (all are four characters)
   db 'DIR '
   db 'ERA '
   db 'TYPE'
   db 'SAVE'
   db 'REN '
   db 'USER'
```

Whatever Jerry Pournelle saw, it was not MS-DOS or CP/M, and there is no secret command and hidden message.

5. Conclusions

The only conclusion is that QDOS and MS-DOS were not copied from CP/M.

Gary Kildall's fate was sad. He died in 1994 at the age of 52. Kildall had suffered from alcoholism in his later years [1] [36]. The circumstances of his death are as muddied and debated as the missed meeting with IBM.

Most agree that he suffered a head injury in a California biker bar [37]; some articles describe a brawl, some people claim he fell from a chair or down a staircase, and others report that he suffered a heart attack. Some claim he committed suicide and his family covered it up, but most agree that his alcoholism in one way or another led to his death [5] [36] [38].

Kildall deserves credit for creating the first personal computer operating system, but the syntax of CP/M [39] looked like a simpler version of many other operating systems in use at the time, including UNIX [40], developed in 1969, and VAX/VMS [41], introduced in 1978. While he is sometimes remembered as a pauper for "being cheated by Bill Gates," DRI was actually a successful company for many years.He eventually sold it to Novell in 1991 for $120 million [1]. Regardless of which stories about Kildall and DRI are correct, Kildall was undeniably very creative and innovative, and very successful. If he was not as successful as Bill Gates, it was not because CP/M source code was stolen to create MS-DOS.

6. Answers to Criticisms

When the initial article was published in the *IEEE Spectrum* online magazine [42], some criticisms were stated by readers that are addressed in this final section of the paper.

6.1. Was Code Claimed to Be Copied?

Some readers stated that Kildall never claimed MS-DOS source code was copied from the CP/M source code. However, the existing DRI website clearly states that MS-DOS is "an unauthorized clone of CP/M" [9]. The Software Engineering Lab (SGL) of the Institute of Computer Science Faculty of the University of Mons (UMONS) [43] defines cloning as follows:

> *Clones* are segments of code that are similar according to some definition of similarity. (Ira Baxter, 2002).
> A software clone is a special kind of *software duplicate*. It is a piece of software (e.g., a code fragment) that has been obtained by *cloning* (*i.e.*, duplicating via the copy-and-paste mechanism) another piece of software and perhaps making some additional changes to it. This primitive kind of software reuse is more harmful than it is beneficial. It actually makes the activities of debugging, maintenance and evolution considerably more difficult.

Clearly this definition means that a clone is a source code copy. Note that examining a product to understand how it works has always been perfectly legal as long as no code is directly copied. In the famous case of *Sega v. Accolade*, Judge Stephen Reinhardt of the US Court of Appeals made this clear in his decision [44]:

> We conclude that where disassembly is the only way to gain access to the ideas and functional elements embodied in a copyrighted computer program and where there is a legitimate reason for seeking such access, disassembly is a fair use of the copyrighted work, as a matter of law.

Granted, this decision came years after MS-DOS was created, but it was a long-standing rule of fair use. Yet Kildall claimed that QDOS, and subsequently MS-DOS, had been directly copied from CP/M and thus infringed on his copyright [1] [9]. More importantly, his attorney Gerry Davis, who would have understood copyright law, claimed that forensic experts had proven that MS-DOS had been copied from CP/M and infringed on the copyright [1].

Finally, if Kildall did not believe that MS-DOS was a direct copy of the CP/M source code, how could he have claimed there was a hidden command in MS-DOS that typed out a secret message [34] [35]? The only way such a routine could exist in both CP/M and MS-DOS is if source code was directly copied.

6.2. Were APIs Copied?

Some readers of the prior article claimed that application specific interfaces (APIs) were copied but not source code. First, this does not conform to the claims of infringement that were made by Kildall as explained above. Second, Tim Paterson admits that he modeled QDOS on CP/M [8], but copying functionality does not constitute copyright infringement. And in a recent ruling in the case of *Oracle v. Google*, Judge Alsup stated that APIs are not protected by copyright even when they are copied directly from the source code [45]:

> So long as the specific code used to implement a method is different, anyone is free under the Copyright

Act to write his or her own code to carry out exactly the same function or specification of any methods used in the Java API.

While it is not clear that this ruling will be upheld on appeal, the analysis showed no literal or non-literal copying of APIs that would constitute copyright infringement.

6.3. MS-DOS Disk Images

After publishing the *IEEE Spectrum* article, disk images of earlier versions of MS-DOS disks were obtained. Because it is not possible to verify the authenticity of these disk images, their analysis is not included in the main body of the paper, but are included here because the results are interesting and confirm the main analysis.

Gio Wiederhold, Professor Emeritus of Computer Science, Medicine, and Electrical Engineering at Stanford University, supplied a disk image of an MS-DOS 1.0 floppy disk in his possession. The image included binary represented in ASCII text and also as raw ASCII text. Searching the raw text for the words "Kildall," "digital," and "DRI" did not find any instances of these words. However, a reference to the name "Robert O'Rear" did show up that, after some research, was found to be the original project manager of MS-DOS at Microsoft [46].

Searching online a disk image was found posted on a website by vintage computer enthusiast Ray Arachelian (aka "Tech Knight") purporting to be an image of an MS-DOS 1.10 floppy disk [47]. Using an old IBM PC running Windows 98 the image was extracted to a disk using instructions found on another web page by author and programmer Daniel B. Sedory (aka "Starman") [48]. Searching the image for the words "Kildall," "digital," and "DRI" did not find any instances of these words.

These disk images were also compared against CP/M and QDOS as described below.

6.3.1. Comparing MS-DOS 1.10 Binary to CP/M Source

The MS-DOS 1.10 binary code comparison actually found fewer matches than the comparison with MS-DOS 1.11. Only 74 of the 95 matching CP/M identifiers that were previously found in MS-DOS 1.11 were found this time, but 18 other identifiers matched MS-DOS 1.10 that were not found in MS-DOS 1.11, shown in **Table 4**.

All of these matching identifiers are common words or programming terms found many times on the Internet as confirmed by Source Detective.

Of the 11 CP/M comments found in the MS-DOS 1.11 binary code, 9 of them were also found in the MS-DOS 1.10 code. There were no additional CP/M comments found in the MS-DOS 1.10 binary code.

6.3.2. Comparing MS-DOS 1.10 Binary to CP/M Binary

This comparison found that only 63 of the 74 CP/M strings found in MS-DOS 1.11 appeared in MS-DOS 1.10. Also 4 other matching identifiers were found in MS-DOS 1.10 that were not found in MS-DOS 1.11:

```
FAST
HEX
OUTPUT
USE
```

All of these matching strings are common words or programming terms found many times on the Internet as confirmed by SourceDetective.

6.3.3. Comparing MS-DOS 1.10 Binary to QDOS Source Code and Binary Code

The MS-DOS 1.10 binary was compared to QDOS source code. Before filtering 243 strings were found in

Table 4. Matching identifiers found in MS-DOS 1.10 binary code and CP/M source code.

Identifiers				
CLEAR	COLUMN	CTS	DEBUG	DEC
DELETE	DONE	FALSE	FOREVER	Items
MAKE	Note	OTHER	OUTPUT	Position
REQUIRES	SCROLL	TRUE		

common between MS-DOS 1.10 and QDOS. Surprisingly, this is less than the 252 strings found in common between MS-DOS 1.11 and QDOS, but some of the matching strings appear to be sequences of random characters that obviously match coincidentally. Also it is not certain that this is a complete, valid copy of MS-DOS 1.10. A large number of common strings were found including the particularly interesting ones that were found with MS-DOS1.11 plus this very long string—obviously a concatenation of error messages—that could not be the result of chance:

```
HEXCOMError in HEX file--conversion aborted$File not found$Address out of
range--conversion aborted$Disk directory full$
```

6.3.4. Comparing MS-DOS 1.0 Binary to CP/M Source

In the MS-DOS 1.0 binary code, again fewer matches were found than were found in MS-DOS 1.11. Only 54 of the 95 matching CP/M identifiers were found that were previously found in MS-DOS 1.11. The were 6 other matching identifiers in MS-DOS 1.0 that were not found in MS-DOS 1.11:

```
BIOS
CTS
DEBUG
DEC
ERROLOW
```

All of these matching identifiers are common words or programming terms found many times on the Internet, with the possible exception of the term ERRO, which appears to be a truncation of the word ERROR at the end of an error message in the MS-DOS 1.0 disk image, which leads me to believe that the image may be corrupted.

Of the 11 CP/M comments found in the MS-DOS 1.11 binary code, only 5 of them were also found in the MS-DOS 1.0 code. There were two CP/M comments found in the MS-DOS 1.0 binary code that were not found in MS-DOS 1.11:

```
CMP
MOVE
```

6.3.5. Comparing MS-DOS 1.0 Binary to CP/M Binary

This comparison found that only 55 of the 74 CP/M strings found in MS-DOS 1.11 appeared in MS-DOS 1.0. There were 4 other matching identifiers in MS-DOS 1.0 that were not found in MS-DOS 1.11:

```
CMP
ERROR$
HEX
LOAD
```

All of these matching strings are common words or programming terms found many times on the Internet.

6.3.6. Comparing MS-DOS 1.0 Binary to QDOS Source Code and Binary Code

The MS-DOS 1.0 binary was compared to QDOS source code. Before filtering, 234 strings were found in common between MS-DOS 1.10 and QDOS. This is surprisingly less than the 252 matching strings found in MS-DOS 1.11 and less than the 243 matching strings found in MS-DOS 1.10 Again though, some of the matching strings appear to be sequences of random characters that match coincidentally, and itis not known for certain that this is a complete, valid copy of MS-DOS 1.0. However, again a large number of common strings was found including the particularly interesting ones that were found with MS-DOS1.11 plus this very long string—obviously a concatenation of error messages—that could not be the result of chance:

```
HEXCOMError in HEX file--conversion aborted$File not found$Address out of
range--conversion aborted$Disk directory full$
```

6.4. Defamation Lawsuit

Some readers have pointed out that Tim Paterson sued Little, Brown and Co, the publisher of the book *They*

Made America and its authors Harold Evans, Gail Buckland, and David Lefer for defamation. The book contends that Paterson "[took] a ride on" Kildall's operating system, appropriated the "look and feel" of the CP/M operating system, and copied much of his operating system interface from CP/M. Paterson contended that statements in the book were "false and defamatory."

That case was dismissed on summary judgment (*i.e.*, without even holding a trial) by US District Judge Thomas S. Zilly. This is not proof that Paterson copied CP/M. There was no trial, no forensic examination, no expert testimony, and no offering of arguments or evidence regarding copying. This was a defamation case, not a copyright infringement case. In the US, speech, even incorrect speech, is protected by our valued First Amendment. According to the Free Online Dictionary, defamation is [49]:

> Any intentional false communication, either written or spoken, that harms a person's reputation; decreases the respect, regard, or confidence in which a person is held; or induces disparaging, hostile, or disagreeable opinions or feelings against a person.

If the authors of the book believed MS-DOS was copied, whether they were correct or not, they had the right to say so. Judge Zilly made this clear in his order [50]:

> Plaintiff Tim Paterson has failed to provide evidence that statements in Sir Harold Evans' chapter on Gary Kildall are provably false or defamatory. The statements in the Kildall chapter constitute non-actionable opinion protected by the First Amendment, or statements that are not provably false. In addition, as a limited purpose figure Mr. Paterson has failed to provide any evidence that Sir Harold Evans acted with actual malice.

So this summary judgment draws no conclusion about whether MS-DOS was copied from CP/M, but only that the authors believed that to be the case and thus had a first Amendment right to say so, just as the readers of my article have a First Amendment right to publicly disagree with my conclusion.

References

[1] Hamm, S. and Greene, J. (2004) The Man Who Could Have Been Bill Gates. Business Week. http://www.businessweek.com/magazine/content/04_43/b3905109_mz063.htm

[2] Computer History Museum (2013) What Was the First PC? http://www.computerhistory.org/revolution/personal-computers/17/297

[3] Wallace, J. and Erickson, J. (1992) Hard Drive. John Wiley & Sons, Hoboken.

[4] Bellins, M. (2011) Putting Microsoft on the Map: History of the MS-DOS Operating Systems, IBM & Microsoft. About.com Guide. http://inventors.about.com/od/computersoftware/a/Putting-Microsoft-On-The-Map.htm

[5] Akass, C. (2006) The Man Who Gave Bill Gates the World. Computeractive. http://www.computeractive.co.uk/pcw/news/1923088/the-bill-gates-world

[6] Smith, T. (2011) Microsoft's MS-DOS Is 30 Today. The Register. http://www.reghardware.com/2011/07/27/ms_dos_turns_30

[7] Honan, M. (2011) Bill Gates Spent the Best Money of His Life 30 Years Ago Today. Gizmodo. http://gizmodo.com/5825184/bill-gates-spent-the-best-money-of-his-life-30-years-ago-today, July 27, 2011

[8] Conner, D. (1998) Father of DOS Still Having Fun at Microsoft. MicroNews. http://www.patersontech.com/dos/microsoft-micronews.aspx

[9] CP/M (2011) The First PC Operating System. http://www.digitalresearch.biz/CPM.HTM

[10] World Intellectual Property Organization (WIPO) (2010) Intellectual Property—Some Basic Definitions. http://www.wipo.int/about-ip/en/studies/publications/ip_definitions.htm

[11] US Copyright Office, Library of Congress (2008) Circular 1, Copyright Basics.

[12] (1970) The Law Professor Behind: ASH, SOUP, PUMP and CRASH. New York Times. http://banzhaf.net/docs/NYTimesBehindASHSoup.pdf

[13] Hollaar, L. (2009) Legal Protection of Digital Information, BNA Books. http://digital-law-online.info

[14] Melling, L. and Zeidman, B. (2012) Comparing Android Applications to Find Copying. *Journal of Digital Forensics, Security and Law*, **7**, 55.

[15] Zeidman, R. (2009) DUPE: The Depository of Universal Plagiarism Examples. 5*th International Conference on IT Security Incident Management & IT Forensics*, 15-17 September 2009.

http://www1.gi-ev.de/fachbereiche/sicherheit/fg/sidar/imf/imf2009/slides/19-RumpSession1-Zeidman_DUPE_IMF2009.pdf

[16] Zeidman, R. (2008) Multidimensional Correlation of Software Source Code. *The 3rd International Workshop on Systematic Approaches to Digital Forensic Engineering*, Oakland, 22 May 2008, 144-156. http://dx.doi.org/10.1109/SADFE.2008.9

[17] Zeidman, R. (2007) Iterative Filtering of Retrieved Information to Increase Relevance. *Journal of Systemics, Cybernetics and Informatics*, **5**, 91-96.

[18] Zeidman, B. (2006) Software Source Code Correlation. 5*th IEEE/ACIS International Conference on Computer and Information Science*, 10-12 July 2006, Honolulu, 383-392.

[19] Clough, P. (2000) Plagiarism in Natural and Programming Languages: An Overview of Current Tools and Technologies. Research Memoranda, CS-00-05, Department of Computer Science, University of Sheffield, Sheffield.

[20] Parker, A. and Hamblen, J. (1989) Computer Algorithms for Plagiarism Detection. *IEEE Transactions on Education*, **32**, 94-99. http://dx.doi.org/10.1109/13.28038

[21] Whale, G. (1990) Identification of Program Similarity in Large Populations. *The Computer Journal*, **33**, 140-146. http://dx.doi.org/10.1093/comjnl/33.2.140

[22] Wise, M.J. (1996) YAP3: Improved Detection of Similarities in Computer Program and Other Texts. *SIGCSE* '96, Philadelphia, 15-17 February 1996, 130-134.

[23] Heckel, P. (1978) A Technique for Isolating Differences Between Files. *Communications of the ACM*, **21**, 264-268.

[24] Wise, M.J. (1993) String Similarity via Greedy String Tiling and Running Karp-Rabin Matching. Department of Computer Science Technical Report, Sydney University, Sydney.

[25] Prechelt, L., Malpohl, G. and Philippsen, M. (2002) Finding Plagiarisms among a Set of Programs with JPlag. *Journal of Universal Computer Science*, **8**, 1016-1038.

[26] Schleimer, S., Wilkerson, D. and Aiken, A. (2003) Winnowing: Local Algorithms for Document Fingerprinting. *SIGMOD* 2003, San Diego, 9-12 June 2003, 76-85.

[27] Zeidman, B. (2011) The Software IP Detective's Handbook. 1st Edition, Prentice Hall, Upper Saddle River.

[28] Chaudry, G. (2011) The Unofficial CP/M Web Site. http://www.cpm.z80.de

[29] Harte, H.M. (2008) Howard's Seattle Computer Products SCP 86-DOS Resource Website. http://www.86dos.org

[30] (2011) CUSTOMIZING MS-DOS Version 1.23 and Later. http://www.bitsavers.org/pdf/seattleComputer/Customizing_MS-DOS_1.23_and_Later.pdf

[31] Johnson, H. (2013) Tarbell S-100 Boards and Docs. http://www.retrotechnology.com/herbs_stuff/d_tarbell.html

[32] Harte, H. (2005) Tarbell Electronics, Tarbell Electronics Manuals. http://www.hartetechnologies.com/manuals/Tarbell

[33] Shvets, G. (2011) Intel 8086 Microprocessor Architecture. CPU World. http://www.cpu-world.com/Arch/8086.html

[34] Pournelle, J. (2011) Interview Discussing Kildall Secret Command in DOS. http://aolradio.podcast.aol.com/twit/TWiT0073H.mp3

[35] Evans, H., Buckland, G. and Lefer, D. (2004) They Made America. Little, Brown and Co., New York.

[36] Rivlin, G. (1999) The Plot to Get Bill Gates. 1st Edition, Crown Business, Random House, New York.

[37] (2011) Digital Research Family Members. http://www.digitalresearch.biz

[38] Young, J. and Kildall, G. (1997) The DOS that Wasn't. Forbes.com, 7 July 1997.

[39] Digital Research, Inc. (1976) An Introduction to CP/M Features and Facilities. http://www.cpm.z80.de/manuals/cpm13int.pdf

[40] Ritchie, D.M. (1979) The Evolution of the Unix Time-Sharing System. *Language Design and Programming Methodology*, Sydney, September 1979. http://www.read.seas.harvard.edu/~kohler/class/aosref/ritchie84evolution.pdf

[41] VargaÉkosEndre (2011) VAX: Virtual Address Extension. http://hampage.hu/vax/e_main.html

[42] Zeidman, B. (2012) Did Bill Gates Steal the Heart of DOS? IEEE Spectrum. http://spectrum.ieee.org/computing/software/did-bill-gates-steal-the-heart-of-dos

[43] (2012) Software Engineering Terminology, Software Engineering Lab (SGL) of the Institute of Computer Science Faculty of the University of Mons. http://informatique.umons.ac.be/genlog/SE/SE-contents.html

[44] Sega Enterprises Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1993)

[45] Oracle America, Inc. v. Google Inc., No. C 10-03561 WHA (N.D. Calif. April 10, 2012)

[46] Gobry, P.E. (2011) 10 Behind-the-Scenes Crankers Who Built The World's Greatest Startups. Business Insider, 6 April

2011.

[47] (2011) IBM DOS 1.10 Ripped and Packaged by Tech Knight for the Endangered Software Archive. http://www.mirrors.org/archived_software/www.techknight.com/esa/download

[48] Sedory, D.B. (2008) Tutorial on Extracting “Disk Images” from a *.DIM File Using HxD (a Disk/Hex Editor). http://thestarman.pcministry.com/tool/hxd/dimtut.htm

[49] (2007) Defamation Definition. Free Online Dictionary. http://legal-dictionary.thefreedictionary.com/defamation

[50] Paterson v. Little, Brown & Co., 502 F. Supp. 2d 1124, 1128 (W.D. Wash. 2007)